NXP Semiconductors
Reference Manual

Document identifier: IMXRT700RM
Rev. 5, 2025-10-31

i.MX RT700 Crossover Microcontrollers Reference Manual

Supports i.MX RT735S, i.MX RT758S and i.MX RT798S

Contents

Chapter 1 About This Manual..... 15
1.1 Audience.....15
1.2 Organization..... 15
1.3 Module descriptions.....15
1.4 Register descriptions..... 18
1.5 Conventions.....20
1.6 Editorial changes..... 21

Chapter 2 Introduction..... 23
2.1 Overview.....23
2.2 Feature summary..... 25
2.3 Architecture overview..... 32
2.4 Applications..... 40

Chapter 3 Memory Map Overview..... 42
3.1 Introduction.....42
3.2 System memory map.....42
3.3 Peripheral memory map..... 46

Chapter 4 Interrupt and DMA Events Assignments..... 53
4.1 Overview.....53
4.2 System interrupts.....53
4.3 DMA assignments..... 64

Chapter 5 Fusemap..... 70
5.1 Overview.....70
5.2 Boot fusemap..... 70
5.3 Fusemap descriptions table.....70
5.4 Shadow register reload.....71

Chapter 6 External Signals and Pin Multiplexing..... 72
6.1 Introduction.....72
6.2 Muxing options..... 72
6.3 Pin assignments..... 126

Chapter 7 Cortex-M33 Platform Overview..... 206
7.1 Overview.....206
7.2 Functional description.....208

Chapter 8 External Memory Controllers.....210
8.1 Overview.....210
8.2 Modules.....210

Chapter 9 Boot ROM..... 212
9.1 Overview.....212
9.2 High level boot flow..... 212
9.3 Pin allocation..... 214
9.4 Boot configuration and boot image.....222
9.5 Prime boot from XSPI NOR.....225
9.6 Prime boot from eMMC..... 247
9.7 Boot from serial NAND (XSPI).....251
9.8 Recovery boot..... 273

Chapter 10 ROM API.....276
10.1 Overview.....276
10.2 Functional description.....276

Chapter 11 In-System Programming (ISP)..... 291
11.1 Overview.....291
11.2 Functional description.....292
11.3 In-System programming protocol..... 294
11.4 Bootloader packet types.....295
11.5 Bootloader command set.....302
11.6 Bootloader Status Error Codes.....322
11.7 UART ISP.....327
11.8 I2C In-System Programming..... 330
11.9 SPI In-System programming..... 333
11.10 USB In-System Programming..... 338

Chapter 12 Power Overview.....341
12.1 Introduction.....341
12.2 Power architecture.....341
12.3 Power modes.....347
12.4 Power mode transitions.....357
12.5 SLEEPCON and PMC.....366
12.6 RAM Arbiter and shared memory power considerations.....368
12.7 Chip operation in low-power modes..... 369

Chapter 13 Clock Overview..... 372
13.1 Overview.....372
13.2 Clocking considerations.....373
13.3 Clock sources.....373
13.4 Generated clock sources.....374
13.5 Clock root generation..... 374
13.6 Clock Gating.....375

Chapter 14 Graphics Overview.....376
14.1 Overview.....376
14.2 Functional description.....379

Chapter 15 Timers Overview..... 381
15.1 Introduction.....381

Chapter 16 System Debug.....385
16.1 Overview.....385
16.2 Functional description.....385

Chapter 17 Audio Overview.....392
17.1 Audio overview.....392
17.2 Functional description.....394

Chapter 18 Analog Overview.....397
18.1 Analog Overview.....397

Chapter 19 Security Overview.....401
19.1 Overview.....401
19.2 Functional description.....403

Chapter 20 IO PAD Controller (IOPCTL).....405
20.1 Chip-specific IOPCTL information.....405
20.2 Overview.....405
20.3 Functional description.....407
20.4 Signals.....408
20.5 Initialization.....408
20.6 Memory map and register definition.....408

Chapter 21 Pin Interrupt and Pattern Match (PINT).....429
21.1 Chip-specific PINT information.....429
21.2 Overview.....430
21.3 Functional description.....431
21.4 External signals.....434
21.5 Initialization.....434
21.6 Application information.....434
21.7 Memory map and register definition.....437

Chapter 22 General-Purpose Input/Output (GPIO).....456
22.1 Chip-specific GPIO information.....456
22.2 Overview.....458
22.3 Functional description.....458
22.4 External signals.....461
22.5 Initialization.....462
22.6 Application information.....462
22.7 Memory map and register definition.....462

Chapter 23 Performance Monitor (CMX_PERFMON).....484
23.1 Chip-specific CMX_PERFMON information.....484
23.2 Overview.....485
23.3 Functional description.....486
23.4 External signals.....487
23.5 Initialization.....487
23.6 Memory map and register definition.....487

Chapter 24 Cache Memory Controller (XCACHE)....496
24.1 Chip-specific XCACHE information....496
24.2 Overview....496
24.3 Functional description....497
24.4 External signals....503
24.5 Initialization....503
24.6 Application information....504
24.7 XCACHE register descriptions....504

Chapter 25 Semaphores2 (SEMA42)....513
25.1 Chip-specific SEMA42 information....513
25.2 Overview....513
25.3 Functional description....515
25.4 External signals....517
25.5 Initialization....517
25.6 Memory map/register definition....517

Chapter 26 Messaging Unit (MU)....522
26.1 Chip-specific MU information....522
26.2 Overview....523
26.3 Functional description....524
26.4 External signals....527
26.5 Initialization....527
26.6 Application information....527
26.7 Register definition....532

Chapter 27 Power Management Controller (PMC)....579
27.1 Chip-specific PMC information....579
27.2 Overview....580
27.3 Functional description....582
27.4 External signals....601
27.5 Initialization....601
27.6 Application information....601
27.7 Memory map and register definitions....607

Chapter 28 Clock Controller (CLKCTL)....867
28.1 Chip-specific CLKCTL information....867
28.2 Overview....867
28.3 System clocks....869
28.4 Functional description....883
28.5 External signals....901
28.6 Initialization....902
28.7 CLKCTL register descriptions....902

Chapter 29 System Controller (SYSCON)....1240
29.1 Chip-specific SYSCON information....1240
29.2 Overview....1240
29.3 Functional description....1242
29.4 External signals....1247
29.5 Initialization....1247

29.6 Memory map and register definition..... 1247

Chapter 30 Sleep Controller (SLEEPCON)..... 1402
30.1 Chip-specific SLEEPCON information.....1402
30.2 Overview.....1402
30.3 Functional description.....1404
30.4 Memory map and register definition..... 1407

Chapter 31 Reset Controller (RSTCTL).....1617
31.1 Chip-specific RSTCTL information..... 1617
31.2 Overview.....1617
31.3 Functional description.....1619
31.4 External signals..... 1623
31.5 RSTCTL register descriptions..... 1624

Chapter 32 FRO Clock Generator (FRO)..... 1706
32.1 Chip-specific FRO information.....1706
32.2 Overview.....1707
32.3 Functional description.....1708
32.4 External signals..... 1710
32.5 Initialization.....1710
32.6 FRO register descriptions.....1711

Chapter 33 Frequency Measurement (FREQME)..... 1726
33.1 Chip-specific FREQME information.....1726
33.2 Overview.....1726
33.3 Functional description.....1727
33.4 External signals..... 1728
33.5 Initialization.....1728
33.6 Memory map and register definition..... 1729

Chapter 34 Nano Power 32 kHz Oscillator (OSC32KNP).....1738
34.1 Chip-specific OSC32KNP information..... 1738
34.2 Overview.....1738
34.3 Functional description.....1739
34.4 External signals..... 1740
34.5 Initialization.....1740
34.6 OSC32KNP register descriptions..... 1740

Chapter 35 Real Time Clock Subsystem (RTC_SS).....1745
35.1 Chip-specific RTC_SS information..... 1745
35.2 Overview.....1745
35.3 Functional description.....1746
35.4 External signals..... 1747
35.5 Initialization.....1747
35.6 Memory map and register description..... 1747

Chapter 36 Real Time Clock (RTC)..... 1754

36.1 Chip-specific RTC information....1754
36.2 Overview....1755
36.3 Functional description....1756
36.4 External Signals....1762
36.5 Memory map and registers....1762

Chapter 37 Process, Voltage and Temperature Sensor (PVTS)....1807
37.1 Chip-specific PVTS information....1807
37.2 Overview....1808
37.3 Functional description....1809
37.4 Application information....1811
37.5 Memory map and registers....1813

Chapter 38 Analog-to-Digital Converter (ADC)....1817
38.1 Chip-specific ADC information....1817
38.2 Overview....1818
38.3 Functional description....1820
38.4 External signals....1830
38.5 Initialization....1830
38.6 ADC register descriptions....1832

Chapter 39 Comparator (ACMP)....1881
39.1 Chip-specific ACMP information....1881
39.2 Overview....1882
39.3 CMP functional description....1885
39.4 DAC functional description....1887
39.5 CMP external signals....1888
39.6 Initialization....1888
39.7 ACMP register descriptions....1888

Chapter 40 Ultra Secured Digital Host Controller (uSDHC)....1899
40.1 Chip-specific uSDHC information....1899
40.2 Overview....1900
40.3 Functional description....1903
40.4 External signals....1933
40.5 Application information....1935
40.6 uSDHC memory map and register definition....1954
40.7 Glossary....2052

Chapter 41 External Serial Peripheral Interface (XSPI)....2054
41.1 Chip-specific XSPI information....2054
41.2 Overview....2057
41.3 Functional description....2058
41.4 External signals....2125
41.5 Initialization....2127
41.6 Application information....2131
41.7 Memory map and register definition....2148
41.8 Flash memory mapped AMBA bus....2297

Chapter 42 Cache Controller (CACHE64_CTRL)..... 2299
42.1 Chip-specific CACHE64_CTRL information..... 2299
42.2 Overview..... 2300
42.3 Functional description..... 2300
42.4 External signals..... 2307
42.5 Initialization..... 2307
42.6 Memory map and registers..... 2307

Chapter 43 CACHE64 Policy Select (CACHE64_POLSEL)..... 2316
43.1 Chip-specific CACHE64_POLSEL information..... 2316
43.2 Overview..... 2317
43.3 Functional description..... 2317
43.4 Application information..... 2318
43.5 Memory map and register definition..... 2319

Chapter 44 Memory Map Unit (MMU)..... 2323
44.1 Chip-specific MMU information..... 2323
44.2 Overview..... 2324
44.3 Functional description..... 2324
44.4 External signals..... 2328
44.5 Initialization..... 2328
44.6 Memory map and register definition..... 2328

Chapter 45 Neural Processing Unit (NPU)..... 2343
45.1 Chip-specific NPU information..... 2343
45.2 Overview..... 2343
45.3 Functional description..... 2347
45.4 External signals..... 2350
45.5 Initialization..... 2350
45.6 Memory map and register definitions..... 2350

Chapter 46 EZ Handler RISC-V (EZH-V)..... 2351
46.1 Chip-specific EZH-V information..... 2351
46.2 Introduction..... 2352
46.3 Features..... 2352
46.4 Block diagram..... 2353
46.5 Functional description..... 2353

Chapter 47 JPEG Decoder (JPEGDEC)..... 2355
47.1 Chip-specific JPEGDEC information..... 2355
47.2 Overview..... 2355
47.3 Functional description..... 2356
47.4 External signals..... 2356
47.5 Initialization..... 2356
47.6 Memory map and register definition..... 2356

Chapter 48 JPEG Decoder Wrapper (JPGDECWRP)..... 2357
48.1 Chip-specific JPGDECWRP information..... 2357
48.2 Overview..... 2357

48.3 Functional description....2359
48.4 External signals....2360
48.5 Initialization....2360
48.6 Memory map and register descriptions....2360

Chapter 49 PNG Decoder (PNGDEC)....2383
49.1 Chip-specific PNGDEC information....2383
49.2 Overview....2383
49.3 Functional description....2384
49.4 External signals....2386
49.5 Initialization....2386
49.6 PNGDEC register descriptions....2386

Chapter 50 Vector Graphics Processing Unit (VGPU)....2404
50.1 Chip-specific VGPU information....2404
50.2 Overview....2404
50.3 Functional description....2406
50.4 Software API....2412

Chapter 51 Universal Serial Bus Controller (USB)....2414
51.1 Chip-specific USB information....2414
51.2 Overview....2415
51.3 Functional description....2415
51.4 External signals....2450
51.5 Initialization....2451
51.6 Application information....2452
51.7 USB noncore memory map and register definitions....2454
51.8 USB core memory map and register definitions....2475

Chapter 52 Universal Serial Bus 2.0 PHY (USBPHY)....2575
52.1 Chip-specific USBPHY information....2575
52.2 Overview....2575
52.3 Functional description....2576
52.4 External signals....2582
52.5 Initialization....2582
52.6 Application information....2582
52.7 USBPHY register descriptions....2583

Chapter 53 USB Device Charger Detection Module (USBDCD)....2633
53.1 Chip-specific USBDCD information....2633
53.2 Overview....2633
53.3 Functional description....2634
53.4 External signals....2648
53.5 Initialization....2648
53.6 Application information....2649
53.7 USBDCD register descriptions....2649

Chapter 54 Mobile Industry Processor Interface and Display Serial Interface Controller (MIPI_DSI_HOST)....2661

54.1 Chip-specific MIPI_DSI_HOST information.....2661
54.2 Overview.....2662
54.3 DSI Host Controller Core.....2663
54.4 DSI D-PHY.....2675
54.5 DPHY Programming.....2684
54.6 Packet Data and Pixel Formats.....2685
54.7 MIPI_DSI_HOST Memory Map/Register Definition.....2686

Chapter 55 LCD Interface (LCDIF).....2746
55.1 Chip-specific LCDIF information.....2746
55.2 Overview.....2746
55.3 Functional description.....2749
55.4 External signals.....2777
55.5 Initialization.....2778
55.6 LCDIF register descriptions.....2783

Chapter 56 Sigma Delta Analog-to-Digital Converter (SDADC).....2869
56.1 Chip-specific SDADC information.....2869
56.2 Overview.....2869
56.3 Functional description.....2871
56.4 External signals.....2875
56.5 Initialization.....2875
56.6 Application information.....2875
56.7 Memory map and register definition.....2876

Chapter 57 PDM Microphone Interface (MICFIL).....2920
57.1 Chip-specific MICFIL information.....2920
57.2 Overview.....2920
57.3 Functional description.....2922
57.4 External signals.....2933
57.5 Initialization.....2933
57.6 Application information.....2936
57.7 MICFIL register descriptions.....2937

Chapter 58 Synchronous Audio Interface (SAI).....2968
58.1 Chip-specific SAI information.....2968
58.2 Overview.....2970
58.3 Functional description.....2971
58.4 External signals.....2977
58.5 Initialization.....2978
58.6 Memory map and register definition.....2978

Chapter 59 HiFi 1 DSP Platform (HiFi1).....3024
59.1 Chip-specific HiFi1 information.....3024
59.2 Overview.....3025
59.3 Functional description.....3026
59.4 Signals.....3028

Chapter 60 HiFi 4 DSP Platform (HiFi4).....3030

60.1 Chip-specific HiFi4 information....3030
60.2 Overview....3031
60.3 Functional description....3033
60.4 Signals....3035

Chapter 61 Micro-Tick Timer (UTICK)....3036
61.1 Chip-specific UTICK information....3036
61.2 Overview....3037
61.3 Functional description....3037
61.4 External signals....3038
61.5 Initialization....3038
61.6 UTICK register descriptions....3038

Chapter 62 Standard Counter/Timers (CTIMER)....3045
62.1 Chip-specific CTIMER information....3045
62.2 Overview....3047
62.3 Functional description....3049
62.4 External signals....3051
62.5 Initialization....3052
62.6 Application information....3052
62.7 CTIMER register descriptions....3052

Chapter 63 OS Event Timer (OSTIMER)....3072
63.1 Chip-specific OSTIMER information....3072
63.2 Overview....3072
63.3 Functional description....3074
63.4 External signals....3075
63.5 Initialization....3075
63.6 Memory map and register definition....3075

Chapter 64 SCTimer (SCT)....3104
64.1 Chip-specific SCT information....3104
64.2 Overview....3105
64.3 Functional description....3107
64.4 External signals....3113
64.5 Initialization....3113
64.6 Application information....3113
64.7 Memory map and register definition....3117

Chapter 65 Multi-Rate Timer (MRT)....3166
65.1 Chip-specific MRT information....3166
65.2 Overview....3166
65.3 Functional description....3167
65.4 External signals....3169
65.5 Initialization....3169
65.6 Memory map and register descriptions....3169

Chapter 66 Windowed Watchdog Timer (WWDT)....3179
66.1 Chip-specific WWDT information....3179

66.2 Overview....3180
66.3 Functional description....3182
66.4 External signals....3184
66.5 Initialization....3185
66.6 Memory map and register definition....3185

Chapter 67 Flexible I/O (FLEXIO)....3193
67.1 Chip-specific FLEXIO information....3193
67.2 Overview....3193
67.3 Functional description....3195
67.4 External signals....3205
67.5 Initialization....3205
67.6 Application information....3206
67.7 Memory map and registers....3225

Chapter 68 Low-Power Flexible Communications Interface (LP_FLEXCOMM)3274
68.1 Chip-specific LP_FLEXCOMM information....3274
68.2 LP_FLEXCOMM....3276
68.3 Low-Power Inter-Integrated Circuit (LPI2C)....3282
68.4 Low Power Serial Peripheral Interface (LPSPI)....3367
68.5 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)....3417

Chapter 69 Improved Inter-Integrated Circuit (I3C)....3473
69.1 Chip-specific I3C information....3473
69.2 Overview....3474
69.3 Functional description....3476
69.4 External signals....3489
69.5 Initialization....3489
69.6 Application information....3491
69.7 I3C register descriptions....3492

Chapter 70 Input Multiplexing (INPUTMUX)....3601
70.1 Chip-specific INPUTMUX information....3601
70.2 Overview....3601
70.3 Functional description....3602
70.4 External signals....3603
70.5 Initialization....3604
70.6 Memory map and register definition....3604

Chapter 71 Enhanced Direct Memory Access (eDMA)....3637
71.1 Chip-specific eDMA information....3637
71.2 Overview....3638
71.3 Functional description....3640
71.4 External signals....3645
71.5 Initialization....3645
71.6 Memory map and register definition....3658

Chapter 72 Network Interconnect Component (NIC)....3800
72.1 Chip-specific NIC information....3800

72.2 Overview....3802
72.3 Functional description....3803
72.4 External signals....3809
72.5 Initialization....3809
72.6 Memory map and register definition....3809
72.7 Glossary....3826

Chapter 73 On-Chip OTP Controller (OCOTP_CTRL)....3828
73.1 Chip-specific OCOTP_CTRL information....3828
73.2 Overview....3828
73.3 Functional Description....3829
73.4 External Signals....3833
73.5 Fuse Map....3833
73.6 Memory Map/Register Definition....3833

Chapter 74 Cyclic Redundancy Check (CRC)....3874
74.1 Chip-specific CRC information....3874
74.2 Overview....3874
74.3 Functional description....3875
74.4 External signals....3878
74.5 Initialization....3878
74.6 Use cases....3878
74.7 Memory map and register descriptions....3881

Appendix A Release notes....3886
A.1 About This Manual changes....3886
A.2 Introduction changes....3886
A.3 Memory Map Overview changes....3886
A.4 Interrupts and DMA Events Assignments changes....3886
A.5 Fusemap changes....3886
A.6 External Signals and Pin Multiplexing changes....3886
A.7 Cortex-M33 Platform Overview changes....3886
A.8 External Memory Controllers Overview changes....3886
A.9 Boot ROM changes....3886
A.10 ROM API chapter changes....3886
A.11 ISP changes....3887
A.12 Power Overview changes....3887
A.13 Clock Overview changes....3887
A.14 Graphics Overview changes....3887
A.15 Timers Overview changes....3887
A.16 System Debug changes....3887
A.17 Audio Overview changes....3887
A.18 Analog Overview changes....3887
A.19 Security Overview changes....3887
A.20 IO PAD Controller (IOPCTL)....3887
A.21 Pin Interrupt and Pattern Match (PINT)....3888
A.22 General-Purpose Input/Output (GPIO)....3888
A.23 Performance Monitor (CMX_PERFMON)....3888
A.24 Cache Memory Controller (XCACHE)....3888
A.25 Semaphores2 (SEMA42)....3889
A.26 Messaging Unit (MU)....3889

A.27 Power Management Controller (PMC)....3889
A.28 Clock Controller (CLKCTL)....3889
A.29 System Controller (SYSCON)....3889
A.30 Sleep Controller (SLEEPCON)....3890
A.31 Reset Controller (RSTCTL)....3890
A.32 FRO Clock Generator (FRO)....3890
A.33 Frequency Measurement (FREQME)....3890
A.34 Nano Power 32 kHz Oscillator (OSC32KNP)....3891
A.35 Real Time Clock Subsystem (RTC_SS)....3891
A.36 Real Time Clock (RTC)....3891
A.37 Process, Voltage and Temperature Sensor (PVTS)....3891
A.38 Analog-to-Digital Converter (ADC)....3891
A.39 Comparator (ACMP)....3892
A.40 Ultra Secured Digital Host Controller (uSDHC)....3892
A.41 External Serial Peripheral Interface (XSPI)....3892
A.42 Cache Controller (CACHE64_CTRL)....3892
A.43 CACHE64 Policy Select (CACHE64_POLSEL)....3893
A.44 Memory Map Unit (MMU)....3893
A.45 Neural Processing Unit (NPU)....3893
A.46 EZ Handler RISC-V (EZH-V)....3894
A.47 JPEG Decoder (JPEGDEC)....3894
A.48 JPEG Decoder Wrapper (JPGDECWRP)....3894
A.49 PNG Decoder (PNGDEC)....3894
A.50 Vector Graphics Processing Unit (VGPU)....3894
A.51 Universal Serial Bus Controller (USB)....3895
A.52 Universal Serial Bus 2.0 PHY (USBPHY)....3895
A.53 USB Device Charger Detection Module (USBDCD)....3895
A.54 Mobile Industry Processor Interface and Display Serial Interface Controller (MIPI_DSI_HOST)....3895
A.55 LCD Interface (LCDIF)....3896
A.56 Sigma Delta Analog-to-Digital Converter (SDADC)....3896
A.57 PDM Microphone Interface (MICFIL)....3896
A.58 Synchronous Audio Interface (SAI)....3896
A.59 HiFi 1 DSP Platform (HiFi1)....3896
A.60 HiFi 4 DSP Platform (HiFi4)....3897
A.61 Micro-Tick Timer (UTICK)....3897
A.62 Standard Counter/Timers (CTIMER)....3897
A.63 OS Event Timer (OSTIMER)....3897
A.64 SCTimer (SCT)....3898
A.65 Multi-Rate Timer (MRT)....3898
A.66 Windowed Watchdog Timer (WWDT)....3898
A.67 Flexible I/O (FLEXIO)....3898
A.68 Low-Power Flexible Communications Interface (LP_FLEXCOMM)....3898
A.69 Improved Inter-Integrated Circuit (I3C)....3899
A.70 Input Multiplexing (INPUTMUX)....3899
A.71 Enhanced Direct Memory Access (eDMA)....3899
A.72 Network Interconnect Component (NIC)....3900
A.73 On-Chip OTP Controller (OCOTP_CTRL)....3900
A.74 Cyclic Redundancy Check (CRC)....3900

Legal information....3901

Chapter 1
About This Manual

1.1 Audience

This reference manual (RM) is intended for system software, hardware developers, and applications programmers who need to develop products using this chip. It assumes that its users understand operating systems, microprocessor system design, and basic principles of software and hardware.

1.2 Organization

This manual has two main sets of chapters.

- Chapters in the first set contain information that applies to all components on the chip.
- Chapters in the second set are organized into functional groupings that detail particular areas of functionality.
  - Examples of these groupings are clocking, timers, and communication interfaces.
  - Each grouping includes chapters that provide a technical description of individual modules.

1.2.1 Attachments

This manual includes key information in the files attached to it. For example, memory map and I/O details. Use the content in these attachments in conjunction with this manual's content.

NOTE

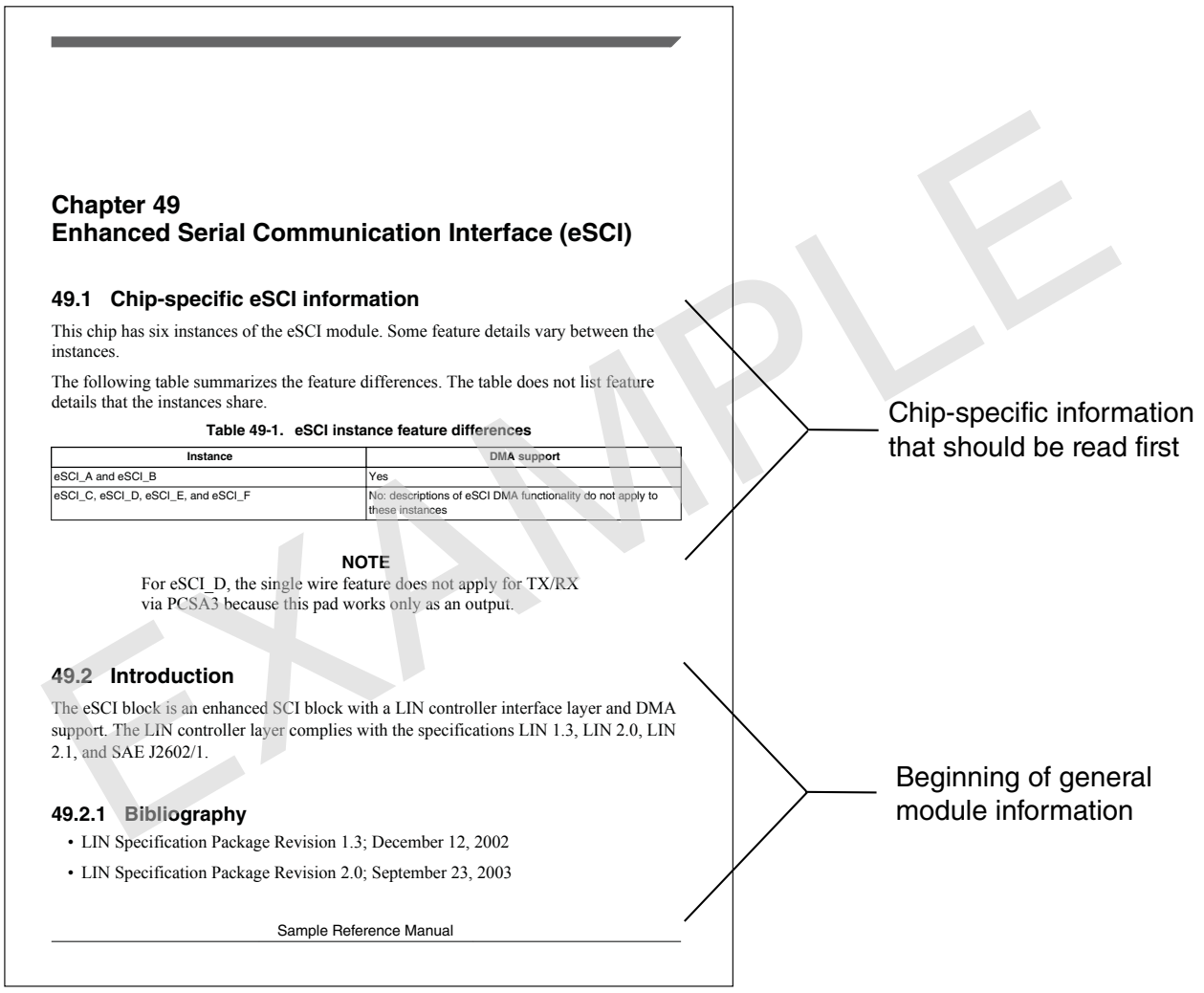
Select the paperclip icon on the left side of the PDF window to see the list of attachments.

1.3 Module descriptions

Each module chapter has two main parts:

- The first section, *chip-specific [module name]* information, provides details such as the number of module instances on the chip and connections between that module and the other ones. Read this section *first* because its content is crucial for understanding the information in the other sections of the chapter.
- The subsequent sections provide general information about the module, including its signals, registers, and functional description.

The following figure shows you an example of this demarcation.



Chapter 49
Enhanced Serial Communication Interface (eSCI)

49.1 Chip-specific eSCI information

This chip has six instances of the eSCI module. Some feature details vary between the instances.

The following table summarizes the feature differences. The table does not list feature details that the instances share.

Table 49-1. eSCI instance feature differences

Instance	DMA support
eSCI_A and eSCI_B	Yes
eSCI_C, eSCI_D, eSCI_E, and eSCI_F	No; descriptions of eSCI DMA functionality do not apply to these instances

NOTE
For eSCI_D, the single wire feature does not apply for TX/RX via PCSA3 because this pad works only as an output.

49.2 Introduction

The eSCI block is an enhanced SCI block with a LIN controller interface layer and DMA support. The LIN controller layer complies with the specifications LIN 1.3, LIN 2.0, LIN 2.1, and SAE J2602/1.

49.2.1 Bibliography

- LIN Specification Package Revision 1.3; December 12, 2002
- LIN Specification Package Revision 2.0; September 23, 2003

Sample Reference Manual

EXAMPLE

Figure 1. Example of chapter chip-specific information and general module information

1.3.1 Chip-specific information that clarifies content in the same chapter

The following figure shows an example of chip-specific information that clarifies general module information presented later in the chapter. In this case, the chip-specific register reset values supersede the reset values that appear in the register diagram.

System Integration Unit Lite2 (SIUL2)

Chapter 9
System Integration Unit Lite2 (SIUL2)

9.1 Chip-specific SIUL2 information

9.1.1 Feature configurations

In this device, the SIUL2_0 module instance does not support the following features described in the generic description:

- Interrupts
- DMA channels

9.1.2 Notes for IMCR

Out of reset, PA_00, PA_04, and PA_05 pads have JTAG input functionality selected by default. It should be disabled in the corresponding IMCR registers (IMCR61, IMCR60, and IMCR50 respectively) in order to use other functionality such as GPIO.

9.2 Introduction

9.2.1 Overview

The System Integration Unit Lite2 provides control over all the electrical pin controls and ports with 16 bits of bidirectional, general-purpose input and output signals. One of the most important functions of the SIUL2 is to enable the user to select the functions and electrical characteristics that appear on external device pins. It also controls the multiplexing of internal signals from one module to another and controls chip I/O. It supports as many as 32 external interrupts with trigger event configuration. The following figure is the block diagram of SIUL2 and its interfaces to other system components.

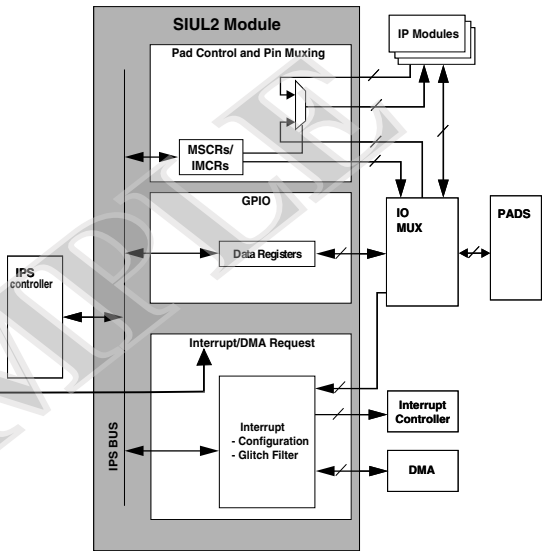


Figure 23. System Integration Unit Lite2 block diagram

This module provides dedicated pad control to general-purpose pads that can be configured as either inputs or outputs. The SIUL2 module provides registers that enable user software to read values from GPIO pads configured as inputs, and write values to GPIO pads configured as outputs:

- When configured as output, you can write to an internal register to control the state driven on the associated output pad.
- When configured as input, you can detect the state of the associated pad by reading the value from an internal register.
- When configured as input and output, the pad value can be read back, which can be used as a method of checking if the written value appeared on the pad.

To assist software development, GPIO data registers can be accessed using various mechanisms. These differing mechanisms allow support for port access or for bit manipulation without the need to use read-modify-write operations:

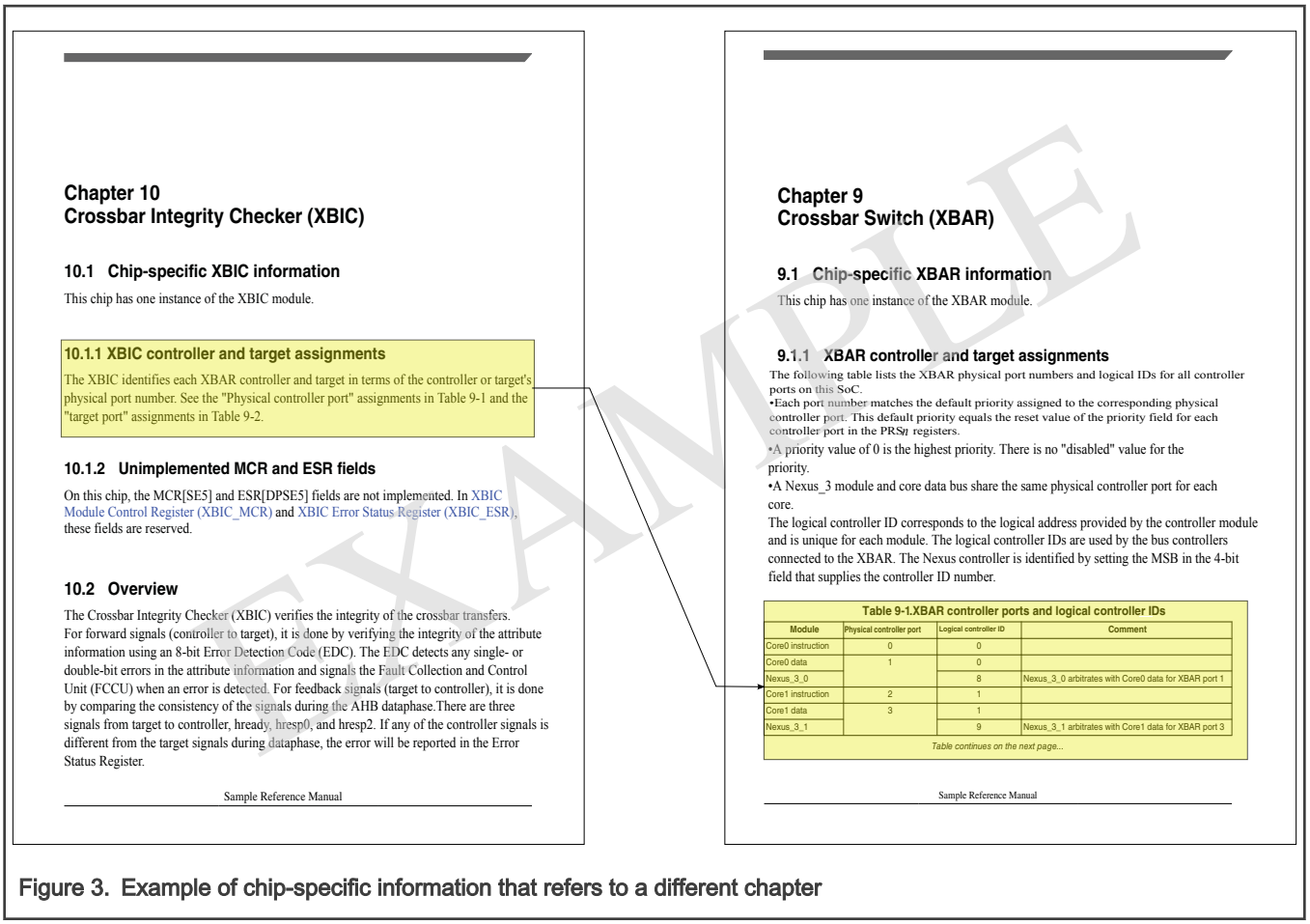
- Access to two 16-bit ports in one access
- Read/write access to a single bit
- A 16-bit port write with a bit mask, using single 32-bit access.

EXAMPLE

Figure 2. Example of chip-specific information that clarifies content in the same chapter

1.3.2 Chip-specific information that refers to a different chapter

Related chip-specific information may be provided in different chapters of the manual. The following figure shows an example of two such connected pieces of information. In this case, read both before you proceed.



Chapter 10
Crossbar Integrity Checker (XBIC)

10.1 Chip-specific XBIC information

This chip has one instance of the XBIC module.

10.1.1 XBIC controller and target assignments

The XBIC identifies each XBAR controller and target in terms of the controller or target's physical port number. See the "Physical controller port" assignments in Table 9-1 and the "target port" assignments in Table 9-2.

10.1.2 Unimplemented MCR and ESR fields

On this chip, the MCR[SE5] and ESR[DPSE5] fields are not implemented. In XBIC Module Control Register (XBIC_MCR) and XBIC Error Status Register (XBIC_ESR), these fields are reserved.

10.2 Overview

The Crossbar Integrity Checker (XBIC) verifies the integrity of the crossbar transfers. For forward signals (controller to target), it is done by verifying the integrity of the attribute information using an 8-bit Error Detection Code (EDC). The EDC detects any single- or double-bit errors in the attribute information and signals the Fault Collection and Control Unit (FCCU) when an error is detected. For feedback signals (target to controller), it is done by comparing the consistency of the signals during the AHB dataphase.There are three signals from target to controller, hready, hresp0, and hresp2. If any of the controller signals is different from the target signals during dataphase, the error will be reported in the Error Status Register.

Sample Reference Manual

Chapter 9
Crossbar Switch (XBAR)

9.1 Chip-specific XBAR information

This chip has one instance of the XBAR module.

9.1.1 XBAR controller and target assignments

The following table lists the XBAR physical port numbers and logical IDs for all controller ports on this SoC.

- Each port number matches the default priority assigned to the corresponding physical controller port. This default priority equals the reset value of the priority field for each controller port in the PRSn registers.
- A priority value of 0 is the highest priority. There is no "disabled" value for the priority.
- A Nexus_3 module and core data bus share the same physical controller port for each core.

The logical controller ID corresponds to the logical address provided by the controller module and is unique for each module. The logical controller IDs are used by the bus controllers connected to the XBAR. The Nexus controller is identified by setting the MSB in the 4-bit field that supplies the controller ID number.

Table 9-1.XBAR controller ports and logical controller IDs

Module	Physical controller port	Logical controller ID	Comment
Core0 instruction	0	0	
Core0 data	1	0	
Nexus_3_0		8	Nexus_3_0 arbitrates with Core0 data for XBAR port 1
Core1 instruction	2	1	
Core1 data	3	1	
Nexus_3_1		9	Nexus_3_1 arbitrates with Core1 data for XBAR port 3

Table continues on the next page...

Sample Reference Manual

EXAMPLE

Figure 3. Example of chip-specific information that refers to a different chapter

1.4 Register descriptions

Module chapters present register information in the following:

- Memory maps, which contain:
  - An offset from the module's base address
  - The mnemonic and name of each register
  - The width of each register (in bits)
  - The reset value of each register
- Register figures
- Field-description tables
- Associated text

The following figure shows register figure conventions used throughout the manual.

Access type	Access description	DITA output	Effect of Write on Value	Readback Value
RW	Read/write	R / W: Mnemonic	Changes to Written Value	Current Value
RO	Read-only	R: Mnemonic / W:	No Effect	Current Value
RU	Reserved, unimplemented	R / W: Reserved	No Effect	Value Undefined
ROZ	Reserved, Read-only zero	R: 0 / W:	No Effect	Returns All Zero
ROO	Reserved, Read-only one	R: 1 / W:	No Effect	Returns All Ones
WO	Write-only	R: / W: Mnemonic	Changed to Written Value	Value Undefined
WOZ	Reserved, Write-only zero	R: / W: 0	Write Value Must Be All 0s.	Value Undefined
WOO	Reserved, Write-only one	R: / W: 1	Write Value Must Be All 1s.	Value Undefined
W0C	Write zero to clear	R: Mnemonic / W: w0c	If a Bit in the Written Value is a 0, the Corresponding Bit in the Field is Set to 0. Otherwise, the Field Bit is Not Affected.	Current Value
W1C	Write one to clear	R: Mnemonic / W: w1c	If a Bit in the Written Value is a 1, the Corresponding Bit in the Field is set to 0. Otherwise, the Field Bit is Not Affected.	Current Value
R2C	Read to clear		Value is cleared following the read operation	Current Value
ROWZ	Read-only, writes zero	R: Mnemonic / W: 0	Write Value Must Be All 0s	Current Value
ROWO	Read-only writes one	R: Mnemonic / W: 1	Write Value Must Be All 1s	Current Value
ROWU	Ready-only writes undefined	R: Mnemonic / W: —	Writes Operation Undefine	Current Value
WORZ	Write-only reads zero	R: 0 / W: Mnemonic	Changes to Written Value	Returns All Zero
WORO	Write-only reads one	R: 1 / W: Mnemonic	Changes to Written Value	Returns All Ones
WORU	Write-only reads undefined	R: — / W: Mnemonic	Changes to Written Value	Value Undefined

Figure 4. Register figure conventions

NOTE

Reset values of reserved locations documented in this manual are subject to change and must not be used for diagnostic purposes.

1.5 Conventions

1.5.1 Notes and cautions

Specific information is provided as part of notes and cautions throughout this manual.

NOTE

Emphasizes information that deserves extra attention.

CAUTION

Informs you of situations that could lead to highly undesirable outcomes—such as damage to the chip or irreversible malfunction.

1.5.2 Numbering systems

The following suffixes identify different numbering systems:

Table 1. Numbering systems

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is mentioned as 101b. In some cases, *0b* is prefixed to binary numbers.
d	Decimal number. Decimal numbers are followed by this suffix only when there is a possibility of confusion. In general, decimal numbers are used without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is mentioned as 3Ch. In some cases, *0x* is prefixed to hexadecimal numbers.

1.5.3 Typographic notation

The following typographic notations are used throughout this document:

Table 2. Typographic notation

Example	Description
x and other italicized text	The italicized, lowercase *x* is used as a placeholder for replaceable numbers. In general, italicized text is used for titles of publications and for emphasis. Additionally, italics could be used for metasymbols in syntax descriptions. Plain lowercase letters are used as placeholders for single letters and numbers.
`code font`	Fixed-width font (such as Courier) used for code. It is used for a letter, word, or phrase that you want the user to type. For example, "Type `Read` and press Enter." This type of font is also used for instruction mnemonics, directives, symbols, subcommands, parameters, operators, computer-language elements, code listings, commands that appear in running text, and for sample code. Instruction mnemonics and directives in text and tables are mentioned in all caps; for example, BSR.
SR[SCM]	A mnemonic in square brackets represents the name of a register field. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets that are separated by a colon represent either: • A subset of a register's named field

Table continues on the next page...

Table 2. Typographic notation (continued)

Example	Description
	For example, REVNO[6:4] refers to bits 6-4 that are part of the COREREV field occupying bits 6-0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7-0 of the XAD bus.
MOD.REG	A period separates the elements of a hierarchy: subsystem.module.register. For example: • SWT.TO means that the TO register is located in the SWT module. • SMU.XRDC.CR means that the CR register is located in the XRDC module within the SMU subsystem.

1.5.4 Special terms

The following terms have special meanings.

Table 3. Special terms

Term	Meaning
Asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
Deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
Reserved	Refers to memory space, register, field, or programming setting. Writes to a reserved location can result in unpredictable functionality or behavior. You must: • Before writing to a location which contain reserved bits user must make sure the write operation will write the reserved bit with value specified as the reset value in NXP reference manual. • Consider undefined locations in memory to be reserved as reset value 0 shall be assumed. You might get a BERR(transfer error) response on access to undefined locations in memory. • If user reads data from memory area containing reserved bit, the value of reserved bits should be ignored and not used for any functional purposes. **NOTE** BootROM could modify the reserved bit values after reset. Please refer to the BootROM settings attachment.
Write 1 to clear (w1c)	Refers to the access type of a register field that is used to clear the field by writing the value 1 to it.
Undefined (u)	Refers to undefined reset values

1.6 Editorial changes

Each new release of this document includes editorial improvements such as:

- Spelling
- Grammar
- Punctuation
- Voice
- Tense
- Capitalization
- Formatting
- Presentation
- Navigation

Chapter 2
Introduction

2.1 Overview

The i.MX RT700 chip contains a rich set of peripherals within multiple functional chip domains. These domains include application core complexes, DSPs, on-chip SRAM, and several high-bandwidth interfaces to access off-chip flash. The chip is architected to be low-power by using advanced clock and power gating techniques, DVFS, and independent power control of specific chip domains. The chip features many highlights, including NXP proprietary machine-learning IP supported by the eIQ™ toolkit, a vector graphics processing unit (VGPU), and a 24-bit Sigma-Delta ADC.

2.1.1 System diagram

The system bus network, primary peripherals, and external interfaces for the chip are illustrated in the diagram below.

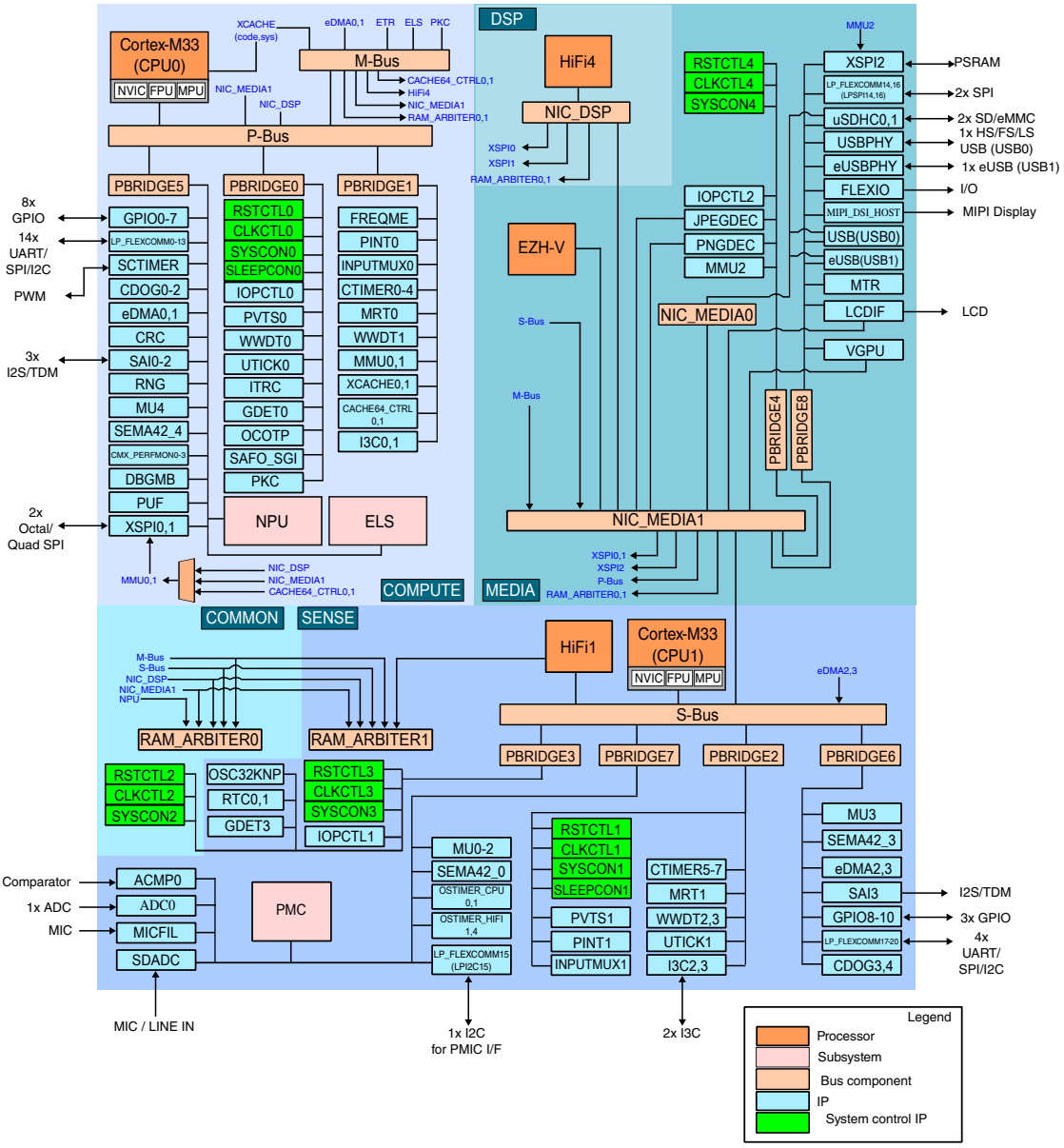


Figure 5. System bus diagram

The following map of the functional domains to the related power domains on the chip are noted in the table below:

Functional domain	Power domain	IP modules
Compute	VDD2_COMP	All modules in power domain
	VDD2_COM	RSTCTL0, CLKCTL0, SYSCON0, SLEEPCON0, IOPCTL0, OCOTP
Common	VDD2_COM	RAM_ARBITER
	VDDN_COM	RSTCTL2, CLKCTL2, SYSCON2, IOPCTL2

Table continues on the next page...

Table continued from the previous page...

Functional domain	Power domain	IP modules
DSP	VDD2_DSP	All modules in power domain
Media	VDD2_MEDIA	All modules in power domain
	VDDN_MEDIA	All modules in power domain
	VDDN_COM	XSPI2
	VDD1V8_PMC	All modules in power domain
	VDD1V8_AON	All modules in power domain
Sense	VDD1_SENSE	All modules in power domain

2.2 Feature summary

The following table shows the chip features organized by function.

Feature	Module	Description
Arm platform and debug	CPU0	• One Compute Domain CPU - Arm Cortex-M33 processor with Trustzone and built-in: — Nested Vectored Interrupt Controller (NVIC) — Non-maskable interrupt (NMI) input — Memory Protection Unit (MPU) — Hardware Floating Point Unit (FPU) — ETM Trace — System Tick Timer (SysTick).
	CPU1	• One Sense Domain CPU - Arm Cortex-M33 co-processor with Trustzone and built-in: — Nested Vectored Interrupt Controller (NVIC) — Non-maskable interrupt (NMI) input — Memory Protection Unit (MPU) — Hardware Floating Point Unit (FPU) — System Tick Timer (SysTick). **NOTE** The configuration of CPU1 does not include ETM.
	Floating Point Unit (FPU)	• Two FPUs (CPU0, CPU1) are single-precision floating point computation units compliant to the IEEE Standard for Floating-Point Arithmetic (IEEE 754).

Table continues on the next page...

Table continued from the previous page...

	Memory Protection Unit (MPU)	• Two MPUs (CPU0, CPU1) provide support for eight unified protection regions, overlapping protection regions with ascending region priority, and access permissions. MPU mismatches and permission violations invoke the Hard Fault handler.
	Nested Vectored Interrupt Controller (NVIC)	• Two NVICs(CPU0, CPU1) implement a relocatable vector table supporting external interrupts, a single non-maskable interrupt (NMI), and priority levels.
	System Tick Timer (SysTick)	See the ARMv8M Architecture Reference Manual for more information about this system timer.
	Cache Controller (CACHE64_CTRL)	• Two CACHE64_CTRLs - cache memory controllers and blocks of high-speed memory locations that contain address information (commonly known as a tag) and its associated data. CACHE64_CTRLs decrease the average time of a memory access.
	CACHE64 Policy Select (CACHE64_POLSEL)	• Two CACHE64_POLSELs monitor incoming bus addresses to define the policy that the cache controllers (CACHE64_CTRL) use. CACHE64_POLSELs and the cache controllers work in tandem with each other.
	Performance Monitor (CMX_PERFMON)	• Four CMX_PERFMONs (System Cache, Code Cache, XSPI0,1 Cache) - contain counters that you can configure to count events, which can be used to monitor the performance of a CPU, cache, or memory. You can configure CMX_PERFMON to select the events to be counted for each counter. An additional instruction counter is available to count CPU instruction fetches.
	Cache Memory Controller (XCACHE)	• Two XCACHEs (System Cache, Code Cache) - general-purpose AMBA AHB bus protocol caches that provide lower latency access to slower memories.
	Memory Map Unit (MMU)	• Three MMUs - remap AHB bus access address to another address indicated by MMU Look Up Table (LUT) SRAM. The remapped address in LUT SRAM can be configured via the internal register bus. • MMU modules are implemented separate from XSPI.
	Semaphores2 (SEMA42)	• Three SEMA42s - memory-mapped modules that provide robust hardware support needed in multi-core systems for implementing semaphores and provide a simple mechanism to achieve "lock and unlock" operations via a single write access. The hardware semaphore modules provide hardware-enforced gates as well as other useful system functions related to the gating mechanisms.
	Messaging Unit (MU)	• Five MUs - enable two processors on a chip to communicate and coordinate by passing messages (e.g. data, status, and control) through the MU interfaces. MUs also provide the ability for one processor to signal the other processor using interrupts.

Table continues on the next page...

Table continued from the previous page...

	Debug Mailbox (DBGMB)	• One DBGMB outlines the debug capabilities implemented in this chip.
Chip IO	IO PAD Controller (IOPCTL)	• Three IOPCTLs provide IO pad control functions. Each port pin PIOm_n has one IOPCTL register assigned to control the characteristics of the pin.
	Pin Interrupt and Pattern Match (PINT)	• Two PINTs provide interrupt functionality for a selected set of GPIO pins.
	General-Purpose Input/ Output (GPIO)	• Eleven GPIOs - communicate to the processor core via a zero wait-state interface for maximum pin. performance. • Up to 217 general purpose I/O (GPIO) pins with configurable pull-up/ pull-down resistors, and adjustable output driver slew rates. • Ports can be set, clear, toggle in bitwise • Mirrored, secure GPIO. • Individual GPIO pins can be used as edge and level-sensitive interrupt sources, each with its own interrupt vector. • All GPIO pins can contribute to one of two ganged (OR'd) interrupts from the GPIO module. • A group of up-to 8 GPIO pins (from Port0/1) can be selected for Boolean pattern matching which can generate interrupts and/or drive a "pattern-match" output.
Clock, power, and reset	Power Management Controller (PMC)	One PMC interfacing to CPU0 (PMC0) and CPU1 (PMC1), manages the hardware functions and connections that support the internal LDO supply and external PMIC supply interface for the VDD1, VDD2, VDDN, and VDD 1.8 V domains. Subsystem Features • Five VDDIO supplies (can be shared or independent) • RBB to provide additional control over power/performance trade-offs
	Clock Controller (CLKCTL)	• Five CLKCTLs (2 Sense, 2 Common, 1 Media) - contain and configure clock sources for peripherals and CPUs. • Crystal Oscillator (XTALOSC) 24 MHz source • Main System PLL: — Allows CPU max frequency operation without the need for a high-frequency crystal. Main System PLL may be run from an FRO, XTALOSC, or the CLKIN pin. — 4 PFD outputs provide flexible clock source to various peripherals.. • Audio PLL for the audio and other modules.
	System Controller (SYSCON)	• Five SYSCONs (2 Sense, 2 Common, 1 Media) provide controls and configures the system and peripherals for the multiple functions.

Table continues on the next page...

Table continued from the previous page...

	Sleep Controller (SLEEPCON)	• Two SLEEPCONs (1 Sense, 1 Common) - programmable low-power controllers to save power for applications.
	Reset Controller (RSTCTL)	• Five RSTCTLs (2 Sense, 2 Common, 1 Media) control module resets.
	FRO Clock Generator (FRO)	• Three FROs provide trimmable system clocks. — Two programmable internal 150-325 MHz FRO with 1% accuracy. — One Internal 192 MHz FRO oscillator with 1% accuracy.
	Frequency Measurement (FREQME)	• One FREQME accurately measures the frequency of an on- or off-chip target clock using a selectable on-chip reference clock.
	Nano Power 32 kHz Oscillator (OSC32KNP)	• One OSC32KNP - an analog interface to an external oscillator to provide a 32.768 kHz (32k) clock and includes a digital module to configure the analog interface, and monitor the 32k clock stability.
	Real Time Clock Subsystem (RTC_SS)	• Two RTC_SSs provide wakeup timers and miscellaneous functions for RTC modules
	Real Time Clock (RTC)	• Two RTCs, RTC0 (CPU0, controller) and RTC1 (CPU1, target) • low-power module that provides time keeping and calendaring functions. RTC can also be used to provide additional security measures and compensate against variations in the 32 kHz oscillator source. • Contains an independent power supply and dedicated oscillator. • Integrated wake-up timer can be used to wake the device up from low-power modes. • The RTC resides in the Always-on voltage domain. • 32 kHz real-time clock (RTC) oscillator that can optionally be used as a system clock.
Security	EdgeLock Secure (ELS)	• One ELS - a security subsystem supporting a wide range of cryptographic algorithms and provides strong key isolation from the rest of the chip.
On-chip memory	RAM Arbiter	• 7.5 MB of system SRAM accessible by the VDD2_COMP and VDD1_SENSE processing domain, DMA engines, and bus controllers
	Misc	• Additional SRAMs - uSDHC FIFOs, GPU, Display controller, MIPI PHY, and XSPI caches, PKC, M33 caches, NPU • Up to 96 KB of cache (32 KB instruction / 64 KB data) on the DSP HiFi4 interface to shared system RAM • 16 kbits OTP Fuses • Up to 256 KB ROM memory for factory-programmed drivers and APIs

Table continues on the next page...

Table continued from the previous page...

External memories and mass storage	Ultra Secured Digital Host Controller (uSDHC)	• Two SD/eMMC memory card interfaces with dedicated DMA controllers. One interface supports eMMC 5.0 with HS400/DDR operation.
	External Serial Peripheral Interface (XSPI)	• Three XSPI (octal/quad) Flash interfaces and x16 PSRAM memories: up to 400 MHz DDR/200 MHz SDR (at least 166 MHz at 1.0 V) with 32 KB cache for XSPI0 and XSPI1 interfaces. • XSPI0 and XSPI1 control 128 MB memories and XSPI2 controls 256 MB memories. See System memory map for more details. • XSPI0 and XSPI1 modules include on-the-fly encryption/decryption for execute-in-place, address-remapping to support dual-image boot, and garbage collection capabilities. DMA supported.
Images, graphics, video, and ML processing	Neural Processing Unit (NPU)	• One NPU- a Neural Network (NN) processing accelerator to accelerate NN matrix computations by deploying parallel Multiply Accumulate (MAC) blocks.
	JPEG Decoder (JPEGDEC)	• One JPEGDEC - a high-performance JPEG decoder for still image and video decompression applications.
	JPEG Decoder Wrapper (JPGDECWRP)	• One JPEGDECWRP - provides the bus interface and DMA engine for JPEGDEC.
	PNG Decoder (PNGDEC)	• One PNGDEC - decodes pictures in the PNG source format.
	Vector Graphics Processing Unit (VGPU)	• One VGPU - 2.5D VGPU provides a high-performance/area UI-based graphic core to enable hardware acceleration of the OpenVG API and display vector graphics on a variety of consumer devices, runnint at frequencies up to 325 MHz.
Display	Mobile Industry Processor Interface and Display Serial Interface Controller (MIPI_DSI_HOST)	• One MIPI_DSI provides an interface that allows communication with MIPI_DSI_HOST-compliant peripherals. • Supports transfer rates up to 895.1 Mbps with two lanes.
	LCD Interface (LCDIF)	• One LCDIF - a high-performance, low-power display processor unit (DPU), which outputs video and graphic streams from a single output panel with up to 720p display resolution.
Audio	HiFi 4 DSP Platform (HiFi4)	• Compute Domain DSP CPU - HiFi4 is an audio engine with a highly optimized audio digital signal processor (DSP), with an extended set of instructions for linear algebra, running at frequencies up to 325 MHz. • Up to 128 KB of DSP HiFi4 local TCM RAM (64 KB instruction / 64 KB data). These RAMs can be accessed by the M33 or the DMA engines via the inbound PIF port.

Table continues on the next page...

Table continued from the previous page...

	HiFi 1 DSP Platform (HiFi1)	• Sense Domain DSP CPU - HiFi1 is low-power audio DSP, running at frequencies up to 250 MHz.
	EZH-V with DSP extensions	• EZH-V Event / Algorithm Handler with 32 KB EZH code RAM and 64-bit I/D TCM.
	Sigma Delta Analog-to-Digital Converter (SDADC)	• One SDADC- a low-power high-performance audio ADC module that supports analog microphones, line inputs, and enables acoustic noise and echo cancellation.
	PDM Microphone Interface (MICFIL)	• One MICFIL - delivers audio from microphones to the processor in several applications • Supports up to 8 channels with associated decimators and Voice Activation Detect. One pair of channels can be streamed directly to SAI. • Supports DMA.
	Synchronous Audio Interface (SAI)	• Four SAIs - provide interfaces that support full-duplex serial digital audio interfaces with frame synchronization formats such as I2S, AC97, TDM, and Codec/DSP interfaces.
Connectivity	Universal Serial Bus Controller (USB)	• One high speed USB (USB0) — provides high-performance USB functionality that conforms to Universal Serial Bus Specification, Rev. 2.0. — USB high-speed host/device controller with on-chip PHY and dedicated DMA controller • One high speed eUSB (USB1) — provides high-performance USB functionality that conforms to Universal Serial Bus Specification, Rev. 2.0 and Embedded Universal Serial Bus 2.0 (eUSB) electrical specification. — eUSB high-speed host/device controller with dedicated DMA controller and on-chip PHY supports both native and repeater modes
	Universal Serial Bus 2.0 PHY (USBPHY)	• One USBPHY - interfaces with the USB host and device system at a low-speed (LS) rate of 1.5 Mbit/s, full-speed (FS) rate of 12 Mbit/s, or USB 2.0 highspeed (HS) rate of 480 Mbit/s.
	USB Device Charger Detection Module (USBDCD)	• One USBDCD - works with the USB transceiver to detect whether the USB device is attached to a charging port, either a Dedicated Charging Port (DCP) or a Charging Downstream Port (CDP).
Low-speed Communications	Flexible I/O (FLEXIO)	• One FLEXIO - emulates of a variety of serial/parallel communication protocols. • Flexible 16-bit timers with support for a variety of trigger, reset, enable, and disable conditions. • Can be configured to provide a parallel interface to an LCD

Table continues on the next page...

Table continued from the previous page...

	Low-Power Flexible Communications Interface (LP_FLEXCOMM)	• Eighteen configurable Universal Serial Interface LP_FLEXCOMM modules. LP_FLEXCOMM 0-13 are used for the VDD2_COMP domain, and LP_FLEXCOMM 17-20 are for the VDD1_SENSE domain. Each module contains an integrated FIFO and has DMA support. Each of the eighteen modules can be configured as: — Low-Power Universal Asynchronous Receive/Transmit (LPUART) with asynchronous transmit and receive. — Low-Power Serial Peripheral Interface (LPSPI) controller or responder, with up to four peripheral chip selects. — Low-Power Inter-Integrated Circuit (LPI2C) with separate controller and responder functions. • Two additional high-speed LPSPI interfaces supporting 50 MHz operation (LP_FLEXCOMM 14 and 16) • One additional LPI2C interface with pseudo open-drain pads (LP_FLEXCOMM15) for PMIC application and used as controller only.
	Improved Inter-Integrated Circuit (I3C)	• Four I3C bus interfaces, that act as an extension of the I2C bus protocol supporting higher speeds.
Timers	Micro-Tick Timer (UTICK)	• Two UTICKs (one for Compute domain and one for Sense domain) - 31-bit timers that each provides a fixed time interval between interrupts. • Ultra-low power micro-tick timer running from the watchdog oscillator, with capture capability for timestamping. • UTICK can be used to wake up the chip from Sleep mode and Deep Sleep mode.
	Standard Counter/ Timers (CTIMER)	• Eight (5 in the Compute domain and 3 in the Sense domain) standard 32-bit Counter/Timers modules with PWM capability
	OS Event Timer (OSTIMER)	• One 64-bit OSTIMER interfaces to CPU0, CPU1, HiFi1 and HiFi4 with individual match/capture and interrupt generation logic.
	SCTimer (SCT)	• One 32-bit SCT- multi-purpose timer with extensive event-generation, match/compare, and complex PWM and output control features with minimal or no CPU intervention. — 10 general-purpose/PWM outputs, 8 general-purpose inputs — Supports DMA and can trigger external DMA events — Supports fractional match values for high resolution
	Multi-Rate Timer (MRT)	• Two (one for Compute Domain and one for Sense Domain) 24-bit multi-rate timer modules with 4 channels each capable of generating repetitive interrupts at different, programmable frequencies.
	Windowed Watchdog Timer (WWDT)	• Four WWDTs, each with a dedicated watchdog oscillator (1 MHz LPOSC), which help reset or interrupt an erroneous microcontroller within a programmable time.

Table continues on the next page...

Table continued from the previous page...

Analog	Analog-to-Digital Converter (ADC)	• One ADC - a low-power 16-bit successive approximation ADC with sampling rates up to 35k samples per second with an enhanced ADC controller • Supports up to 14 single-ended channels or 7 differential channels • DMA supported.
	Comparator (ACMP)	• One ACMP - provides a circuit for comparing two analog input voltages.
Interconnects, crossbars and DMA	Input Multiplexing (INPUTMUX)	• Two INPUTMUXes - provide signal routing for internal peripherals.
	Enhanced Direct Memory Access (eDMA)	• Four eDMAs, each with a general purpose DMA engine, capable of performing complex data transfers with minimal intervention from a host controller. — eDMA can be configured such that one pair is secure and the other non-secure. — eDMA can also be configured as one-pair dedicated for the control CPUs and the other for the DSPs.
	Crossbar Switch Lite (AXBS_Lite)	• AXBS_Lite connects bus controllers and bus targets using a crossbar switch structure.
	Peripheral Bridge (AIPS_Lite)	• AIPS_Lite converts the crossbar switch interface to an interface that can access most of the target peripherals on this chip.
	Peripheral Bridge (PBRIDGE2)	• PBRIDGE2 converts an AMBA AHB interface to a peripheral interface.

2.3 Architecture overview

The RT700 is architected for sustained low power operation while supporting advanced peripherals and secure applications. The chip is a general purpose MCU, but is well-suited for IoT and wearable applications. It has five independent chip domains (Compute, Sense, Common, DSP, Media), two of which serve as the primary processing domains – the Compute and Sense Domains.

The Compute and Sense Domains each have a Cortex-M33 processor, co-located with a DSP, and an extensive set of peripherals, which include a Machine Learning module (NPU), a Graphics module (VGPU), and an advanced security subsystem (ELS). Each domain also has access to various global SoC resources, including shared system RAM, analog modules, and GPIO.

NOTE

For power architecture information, refer to the Power Overview chapter.

2.3.1 Processing Domains

2.3.1.1 Sense Domain

The Sense Domain implements a light-version of the Cortex M33 complex and a HiFi1 DSP. The primary feature of the Sense Domain is its ability to support primary MCU functions while maintaining low-energy consumption. This functional chip domain is essentially part of the Always-on (AON) power domain. With all other domains powered off, the Sense Domain can be configured to identify keywords, and upon a positive classification, wakeup other domains to carry out the remainder of the power heavy application processes.

2.3.1.2 Compute Domain

The Compute Domain implements a full-featured Cortex M33 complex and a HiFi4 DSP. The DSP can run at a different frequency from the M33 and the other modules in the domain, similar to the M33 and DSP in the Sense Domain. Each of the two domains can be independently powered-off allowing this chip to operate under minimum power consumption when the Compute Domain is not in use.

The Compute Domain features a split bus architecture to minimize power consumption. The bus is split into a memory bus (M-Bus) and a periphery bus (P-Bus) to keep toggling activity minimized to only required memory accesses.

Another architectural feature introduced in the Compute Domain is the integration of a cache controller associated to the M33. This provides a quasi single-cycle access to the shared SRAM, and helps eliminate power consumption dependence on the physical position of the SRAM. The SRAM is divided into individual partitions, which can independently be placed in low-power retention mode or powered off entirely. The chosen SRAM instances are also configured to minimize active power consumption, while maintaining low leakage in Deep Sleep Mode.

2.3.2 Processors

2.3.2.1 Cortex M33 CPU in the Compute Domain – CPU0

CPU0 serves as the main control CPU for this chip. CPU0 is responsible for conducting the booting process and interfacing with external memories including XSPI flash, USB, SPI, I2C, UART and eMMC. Boot source selection will be specified via OTP. Following boot-up and part initialization, CPU0 can be placed into sleep mode under software control. For more information on CPUs, refer to the Cortex-M33 platform overview.

2.3.2.2 Cortex M33 CPU in the Sense Domain – CPU1

CPU1 resides in the Sense Domain. This CPU has access to all peripherals in the Sense Domain and the shared region. System-level resources are configured and managed by the M33 in the Compute Domain (CPU0) . For more information on the CPUs, refer to the Cortex-M33 platform overview.

2.3.2.3 HiFi4 and HiFi1 DSP

The HiFi4 DSP is adjacent from the Compute Domain and serves as the DSP core for execution of audio voice codecs, pre- and post-processing modules, and offloading machine learning functions for accelerated processing. The HiFi1 DSP resides in the Sense Domain and is primarily intended for Always-on applications, Bluetooth LE audio, and Sensor HiFi1.

These DSP cores can access many of the on-chip peripherals including the audio modules, LP_FLEXCOMM, I3C, and an assortment of timers and other general system resources. The HiFi4 also has access to the 24-bit SDADC (mic input) in the Common Domain.

The audio modules include interfaces for 8 mic inputs (MICFIL - four channel-pairs) with decimators, filters, and hardware voice-activation detect (one channel). When voice is detected, an interrupt can wakeup the DSP CPU for voice/pattern matching.

2.3.2.4 EZH-V

The EZH-V is primarily implemented to provide a “smart DMA” engine for post-processing graphics data assembled by either or both the GPU and CPU, and then passing it to the FLEXIO for output. Processing operations may include data packing and endianness adjustment. The EZH-V is available for any other general-purpose handling that may be required and supports an extensive selection of trigger inputs from GPIOs and most on-chip peripherals.

2.3.3 Peripherals

This section covers several of the key peripherals on the chip, which complement the general peripherals and system control modules.

2.3.3.1 SDADC - Sigma Delta Analog-to-Digital Converter

This ADC is a high dynamic range audio ADC that supports both line input and microphone input. It provides low latency decimation to enable Acoustic Noise Cancellation and Acoustic Echo Cancellation. It has a high input impedance to reduce its loading to input sources and to improve its robustness to interference. The DAC inside ADC can be used as a standalone DAC to provide a low-impedance voltage output.

This intended use of the sigma delta converter is in systems using wide dynamic range analog mems microphones. These systems ranges from hearable devices, wearable devices, to battery operated devices for always-on listening. The battery powered devices have a broad application from remote controls to security devices. Since primary use for these converters is battery powered devices power consumption is an important spec.

The reason for implementing an SDADC on the chip include:

- Analog mic has a smaller footprint.
- Analog mic has better SNR than the digital mic alternative (dependent on sigma delta implementation)
- Analog mic is lower in power than the digital mic alternative (dependent on sigma delta implementation)

2.3.3.2 LP_FLEXCOMM – Low-power SPI, I2C, UART

Each LP_FLEXCOMM, except for the three dedicated LP_FLEXCOMMs, may be configured by the user either as a LPUART, LPSPI, or LPI2C module.

Two dedicated LPSPI LP_FLEXCOMM instances are provided. These standalone high-speed SPIs provide 50 MHz transfer rates. The pins associated with each of these modules are fixed on one set of specific GPIOs. (accessible by both CPUs and DSPs, and DMA controllers. DMA requests may be MUXed with other functions)

One dedicated LPI2C LP_FLEXCOMM instance is provided to communicate with the PMIC. This LPI2C uses pseudo open-drain pads and does not support DMA.

2.3.3.3 XSPI – serial memory controller

This chip includes three XSPI serial memory controllers. These controllers support serial flash and serial RAM memory devices. The interface is up to x16 depending on the XSPI instance being used. See XSPI instances features for more details about the three XSPI instances.

XSPI0 is primarily intended for code execution and boot from an off-chip SPI NOR flash memory. It supports execute-in-place (XIP) and on-the-fly (OTF) decryption using the Inline PRINCE encryption and decryption (IPED) sub-module integrated as part of the XSPIs. Because XSPI0 is intended for code execution it supports IPED CTR and GCM modes.

XSPI1 can be used for either as a boot source (connected to an off-chip SPI NOR flash memory) or as RAM expansion (connected to an off-chip pSRAM/HyperRAM device). XSPI1 supports execute-in-place (XIP) and on-the-fly (OTF) decryption using the Inline PRINCE encryption and decryption (IPED) sub-module integrated as part of the XSPIs. Because XSPI1 can be used for code or data, its IPED supports IPED CTR, GCM, and XEX modes.

XSPI2 is intended for RAM expansion (connected to an off-chip pSRAM/HyperRAM device). In particular, XSPI2 can be used for storing graphics and data used by the MEDIA domain.

Each of the XSPI instances is also integrated with an MMU. The MMU can be used for remapping memory ranges. For the XSPI boot instances (XSPI0 and XSPI1), the ROM uses the MMU module to support dual-image boot.

Figure 6 shows the initiators that access the external memory via each XSPI instance, see System memory map for more details about the initiators. As shown in the diagram, XSPI0 and XSPI1 are both interfaced to a CACHE64_CTRL; however, only the M-Bus initiator access path goes through the caches. All bus initiator accesses route through the MMUs. Only one initiator can access an XSPI instance at a time. For XSPI0 and XSPI1, you can configure the priority for arbitration between the three main ports to the XSPIs using SYSCON0.AXBS_CTRL.

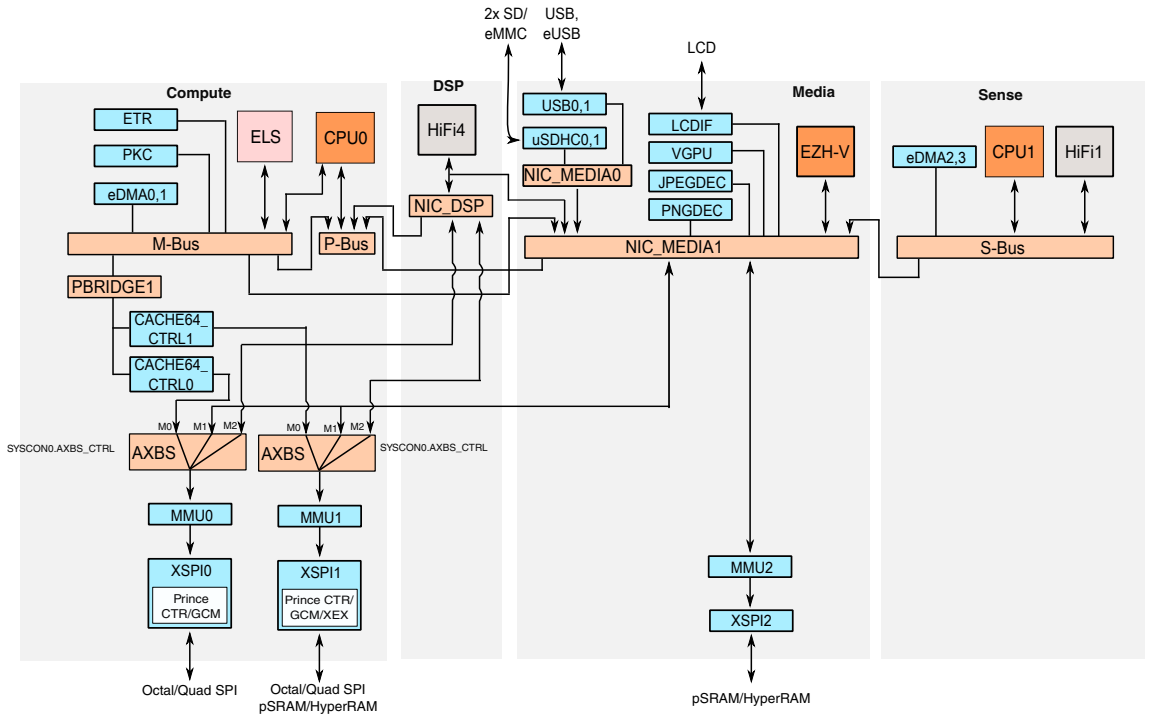


Figure 6. External memory initiators

2.3.3.4 eUSB Host/Device with on-chip PHY

eUSB (USB1) is introduced to address the following:

- IO power efficiency: Improve link power efficiency (active and idle)
- Process scalability: Provide a low voltage USB2.0 PHY solution
- Implementation simplicity: The PHY analog content is reduced. Digital mechanisms are employed for PHY functionality, for example, device disconnect detect.
- Support both USB2.0 inter-chip and out-of-the-box devices: Although eUSB and USB2.0 are not electrically compatible, a mechanism is defined for eUSB to support standard USB2.0 devices

2.3.4 Display interface

This chip has a pad segment devoted to driving a memory in pixel display. As a typical application, the display uses 2-bit color for a total of 6 data lines and has horizontal sync, vertical sync and clock signals. In addition, the chip can provide the panel back plane signals that are driven from the 32 kHz RTC clock.

2.3.5 System control modules

The chip implements local clock, reset, and power control across the main power domains. System control modules are instantiated in the VDD2_MEDIA, VDD2_COM, VDDN_COM, and twice in the VDD1_SENSE power domains. Refer to the related Clock overview, Power overview, PMC, RSTCTL, CLKCTL, SYSCON, SLEEPCON and IOPCTL chapters for more information.

2.3.5.1 RSTCTL – Reset Controller

The RSTCTL module is instantiated in the Common, Sense (main, private), Media, and VDDN domains. The module configures the local resets for associated peripherals in the domain. The details of supported domain resets, reset handling, and reset sources are detailed in the RSTCTL chapter.

2.3.5.2 CLKCTL – Clock Controller

The CLKCTL module is instantiated in the Common, Sense (main, private), Media, and VDDN domains. This module controls the source selection and clock generation of system and peripheral clocks. This module also captures the peripheral clock gate assignments across the chip. For more information about clock generation and peripheral clock gating, refer to the CLKCTL chapter.

2.3.5.3 SYSCON – System Controller

This module configures system resources to critical peripherals. The SYSCON controls bus accesses, NMI source selection, debug access to processors, clock gating to memory and peripherals, controls security features, temperature detect, GPIO config, and supports a gray-to-binary converter. Refer to SYSCON chapter for more information.

2.3.5.4 SLEEPCON – Sleep Controller

The SLEEPCON configures the low-power control for the chip. This module is instantiated with the PMC and CLKCTL instances in the VDD2_COM and VDD1_SENSE power domains. The module works in conjunction with the CLKCTL and PMC to coordinate power mode transitions, and enables wakeup interrupts. Refer to the SLEEPCON chapter for more information.

2.3.6 Bus architecture

The chip is categorized by voltage domains and functional domains. Some functional domains straddle multiple voltage domains, but are functionally grouped by the bus fabric within the domain. The bus fabric includes bus controllers, crossbars, and peripheral bridges, which connect sets of associated peripherals on the chip. Arbitration and priority for some of the bus components is managed by SYSCON or directly by the NIC.

NOTE

NIC_DSP and NIC_MEDIA0 have static configurations. Only NIC_MEDIA1 is configurable. See the NIC chapter for more information.

NOTE

SYSCON module controls several functions relating to the bus fabrics and crossbars, including priority. See SYSCON for more information.

The diagram below illustrates the initiators (cores), bus fabric (NICs and buses), and target (peripheral and internal SRAM) connections.

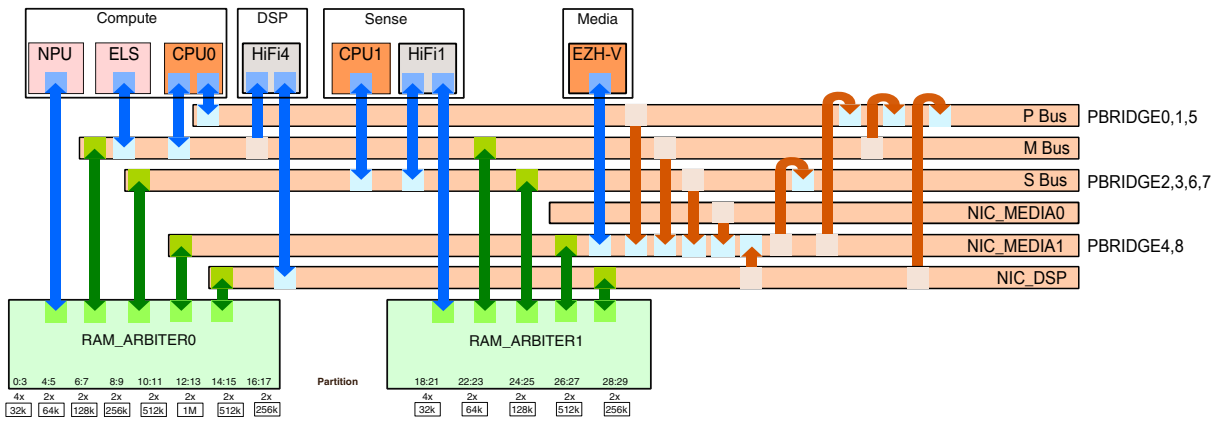


Figure 7. Chip bus topology diagram

NOTE

Initiator access to target peripherals are noted in the memory map. See the memory map for detailed peripheral access information.

2.3.6.1 RAM arbiters

The chip uses RAM arbiters to interface to the internal SRAM through the various buses on the chip. The RAM arbiters and SRAM reside in the Compute and Sense domains, which can operate in separate voltage domains. The Compute and Sense domains have access to each other's memory. RAM_ARBITERn consists of dedicated arbiters for each RAM partition. Table 4 and Table 5 show the ports on each bus and NIC via which initiators can access the RAM arbiters and SRAM partitions. A port on buses or NICs may connect to one or more memory partitions, but it can access one partition at a time.

NOTE

Ensure a shared memory partition is accessed by only one initiator at a time for better performance. The chip supports to access one memory partition by several initiators, but if several initiators access the a memory partition at the same time, it will reduce the performance because of latency.

The figure and tables below cover the initiators to the RAM arbiters.

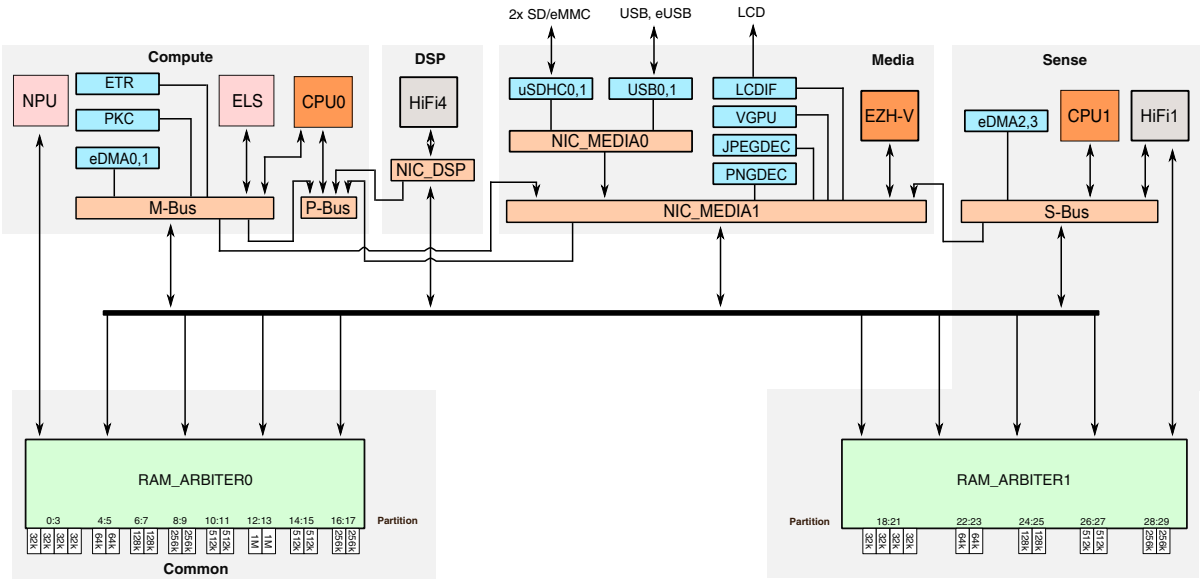


Figure 8. RAM arbiter initiators

Table 4. RAM_ARBITER0 shared SRAM partitions

Shared SRAM partition	Start address	End address	Initiator via ports on				
			NIC_MEDIA1 [1]	M-Bus	S-Bus	NIC_DSP	NPU
P0	2000_0000	2000_7FFF	AMIB3	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3	HiFi4	Two ports, one for NPU-D, the other for NPU-R
P1	2000_8000	2000_FFFF					
P2	2001_0000	2001_7FFF					
P3	2001_8000	2001_FFFF					
P4	2002_0000	2002_FFFF		CPU0, eDMA0, eMDA1, ELS, PKC, ETR			
P5	2003_0000	2003_FFFF					
P6	2004_0000	2005_FFFF		CPU0, eDMA0,			

Table continues on the next page...

Table 4. RAM_ARBITER0 shared SRAM partitions (continued)

Shared SRAM partition	Start address	End address	Initiator via ports on				
			NIC_MEDIA1 [1]	M-Bus	S-Bus	NIC_DSP	NPU
P7	2006_0000	2007_FFFF		eMDA1, ELS, PKC, ETR			
P8	2008_0000	200B_FFFF		CPU0, eDMA0, eMDA1, ELS, PKC, ETR			
P9	200C_0000	200F_FFFF					
P10	201_00000	2017_FFFF	AMIB4	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3		
P11	2018_0000	201F_FFFF					
P12	2020_0000	202F_FFFF	AMIB5	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3		
P13	2030_0000	203F_FFFF					
P14	2040_0000	2047_FFFF	AMIB6	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3		
P15	2048_0000	204F_FFFF					
P16	2050_0000	2053_FFFF	AMIB7	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3		
P17	2054_0000	2057_FFFF					

1. See Interconnection for initiators connect to each AMIB port.

Table 5. RAM_ARBITER1 shared SRAM partitions

Shared SRAM partition	Start address	End address	Initiator via ports in				
			NIC_MEDIA1 [1]	M-Bus	S-Bus	NIC_DSP	HiFi1
P18	2058_0000	2058_7FFF	AMIB8	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3	HiFi4	Two ports, one for HIFI-I, the other for HIFI-D
P19	2058_8000	2058_FFFF					
P20	2059_0000	2059_7FFF					
P21	2059_8000	2059_FFFF					
P22	205A_0000	205A_FFFF			CPU1 system bus, eDMA2, eDMA3		
P23	205B_0000	205B_FFFF					
P24	205C_0000	205D_FFFF			CPU1 system bus, eDMA2, eDMA3		
P25	205E_0000	205F_FFFF					

Table continues on the next page...

Table 5. RAM_ARBITER1 shared SRAM partitions (continued)

Shared SRAM partition	Start address	End address	Initiator via ports in				
			NIC_MEDIA1 [1]	M-Bus	S-Bus	NIC_DSP	HiFi1
P26	2060_0000	2067_FFFF	AMIB9	CPU0, eDMA0, eMDA1, ELS, PKC, ETR	CPU1 system bus, eDMA2, eDMA3		Two ports, one for HIFI-I, the other for HIFI-D
P27	2068_0000	206F_FFFF					
P28	2070_0000	2073_FFFF	AMIB10		CPU1 system bus, eDMA2, eDMA3		
P29	2074_0000	2077_FFFF					

1. See Interconnection for initiators connect to each AMIB port.

Table 6 provides baseline cycle counts for bus initiators to access RAM partitions based on the primary bus fabric that initiator uses. For example, CPU0 uses the M-Bus and can access RAM_ARBITER0 partitions in two clock cycles, but takes thirteen clock cycles to access RAM_ARBITER1 partitions. Memories are endpoints and take the clock of the bus trying to connect to it. Therefore, the cycle counts in Table 6 are based on the bus clock for the initiator. The cycle counts provided here are the minimum number of cycles to access the RAM. The actual cycle count will depend on RAM activity (how often the RAM is accessed and if it is accessed by multiple initiators), RAM clocking, and bus arbitration.

In general, there is no synchronization required to connect to memories using this arbitration scheme. However, if the partition was last accessed by an initiator using a different clock, then it takes two clock cycles to switch to the new initiator clock. The RAMs also support auto clock gating that turns off the clock to each partition if that partition is not accessed for a period. If the clock to a partition is off at the time an access starts, it takes 10 clock cycles to wake up the memory. This feature can be turned off, but memories will consume more power. See RAM Arbiter and shared memory power considerations for more details on low power features.

Table 6. Initiators access SRAM partition cycles via buses and NICs

Shared SRAM partition	M-Bus (cycles)	NIC_MEDIA 1 (cycles)	S-Bus (cycles)	NIC_DSP (cycles)	HiFi1 TCM (cycles)	NPU (cycles)
0-17	2	2	13	2	NA	2
18-29	13	13	1	13	1	NA

2.3.6.2 Compute domain

The Compute domain contains the main Cortex-M33 (CPU0) complex, HiFi4 DSP core, NPU core, and a diverse set of peripherals. One of the new architectural features is cache support for CPU0. In this scheme, shared SRAM is cached, but the peripheral set is directly accessed by CPU0. There are separate cache controllers for CPU0's code and system buses. The Compute domain has a split bus architecture, an M-Bus Matrix that primarily connects core Initiator to (shared) memory, and a P-Bus Matrix that interfaces to peripherals. The M-Bus and P-Bus are connected together through a Target port on the M-Bus to an Initiator port on the P-Bus.

This connection allows core Initiators other than CPU0 and HiFi4 to have memory and periphery access. This choice of architecture mitigates toggling activity as it separates memory access from peripheral access for the sake of low power consumption. The HiFi4 has a private cache controller, independent of CPU0's access the shared memory. The chip's security subsystem is also located in the Compute domain. This system is the ELS security subsystem. It is makes use of the hardware accelerator PKC.

2.3.6.3 Sense domain

The Sense domain has its own bus, S-Bus, which connects the other Cortex-M33 core (CPU1), a HiFi1 DSP, and two DMA controllers as core Initiators. The domain has shared SRAM (P18-P29). It also offers a simple set of peripherals, and a Messaging Unit (MU) with Semaphore (SEMA42) for local CPU1 – HiFi1 intercommunication.

The CPU1 of the Sense domain can access SRAM for code and data via S-Bus. The HiFi1 DSP does not have a private cache memory, instead its dual memory TCM bus interfaces are connected to shared memory.

The RTC0/1, LPI2C15, MICFIL, ACMP0 and ADC0 in Sense domain are shared modules which can be controlled by both CPU0 and CPU1. The rest of the modules in this Sense domain are private modules which are controlled by CPU1 only.

2.3.6.4 Common domain

The common domain holds modules related to system control, clock generation, reset control, pin characteristic control, and internal power regulators, as well as the shared SRAM (P0-P17, see System memory map) and peripherals such as the memory interface of XSPI0-2.

2.3.6.5 Media domain

The purpose of the Media domain is to hold peripherals that require high-performance. The Media domain includes a NIC interface (NIC_MEDIA0, NIC_MEDIA1 controllers), the uSDHC (NAND) controllers, the USB (USB0) and eUSB (USB1) IPs, and the graphics subsystem integrated by the JPEG decoder, PNG decoder, VGPU, LCDIF, DSI host interface, as well as a MIPI PHY.

The use cases with graphics involve access to external memory via XSPI2, with the corresponding memory map unit (MMU2) within the domain. The Media domain includes high-speed SPIs, and the smart DMA EZH-V. The Media domain has cross links to the Compute and Sense domains, and direct access to the shared memory.

2.3.6.6 Inter-domain communications

2.3.6.6.1 Messaging unit

The Messaging Unit (MU) is a shared peripheral with a separate IP bus interface and interrupt request for each host processor. The MU exposes a set of registers to each processor which facilitate inter-processor communication. The MU module enables two processors within the chip to communicate and coordinate by passing messages (e.g. data, status and control) through the MU interface. The MU also provides the ability for one processor to signal the other processor using interrupts. The MU ports are assigned by such rule as: MUA is always used for the Compute domain and MUB for Sense domain, and then within a domain MUA is assigned to the CPU core and the MUB is assigned to the DSP core.

2.3.6.6.2 Semaphore

The Hardware Semaphore module provides the hardware support needed in multi-core systems for implementing semaphores and provides a simple mechanism to achieve "lock/unlock" operations via a single write access.

2.3.7 Reference manual organization

The arrangement of the Reference Manual is based on the functional domains of the chip. See the specific functional overview chapters for more information covering the related peripherals and system related information.

2.4 Applications

This chip supports a broad-array of defined use cases and numerous general use applications. Below is a list of common applications and segments the chip supports, based on a full-featured part.

Table 7. Supported applications

Segment	Applications
Display and graphics	• Battery powered devices • Multiple display options (including OLED 24-bit color) • Augmented reality glasses
Audio	• Line in, analog and digital microphones • Battery powered Always-on listening
Machine learning (ML)	• Voice activation detection (VAD) • Keyword spotting gesture recognition
Wearable	• On-wrist devices • Fitness trackers
Hearable	• On-ear devices • Over-the-ear devices • Head mounted devices
Battery powered devices	• Remote control devices • Security devices toys

Chapter 3
Memory Map Overview

3.1 Introduction

This section introduces the memory architecture of the chip.

NOTE

Accessing the reserved memory regions can result in unpredictable behavior.

3.2 System memory map

The system memory map represents the system memory view for all processors. The memory slots in the table represent the non-secure address region.

NOTE

See i.MX RT700 Security Reference Manual for secure address region.

Table 8. System memory map

Start address	End address	Size	Allocation	Cached	Access[1]
0000_0000	0000_7FFF	32 KB	SRAM P0 alias	CPU0 code	CPU0
0000_8000	0000_FFFF	32 KB	SRAM P1 alias	CPU0 code	CPU0
0001_0000	0001_7FFF	32 KB	SRAM P2 alias	CPU0 code	CPU0
0001_8000	0001_FFFF	32 KB	SRAM P3 alias	CPU0 code	CPU0
0002_0000	0002_FFFF	64 KB	SRAM P4 alias	CPU0 code	CPU0
0003_0000	0003_FFFF	64 KB	SRAM P5 alias	CPU0 code	CPU0
0004_0000	0005_FFFF	128 KB	SRAM P6 alias	CPU0 code	CPU0
0006_0000	0007_FFFF	128 KB	SRAM P7 alias	CPU0 code	CPU0
0008_0000	000B_FFFF	256 KB	SRAM P8 alias	CPU0 code	CPU0
000C_0000	000F_FFFF	256 KB	SRAM P9 alias	CPU0 code	CPU0
0010_0000	0017_FFFF	512 KB	SRAM P10 alias	CPU0 code	CPU0
0018_0000	001F_FFFF	512 KB	SRAM P11 alias	CPU0 code	CPU0
0020_0000	002F_FFFF	1 MB	SRAM P12 alias	CPU0 code	CPU0
0030_0000	003F_FFFF	1 MB	SRAM P13 alias	CPU0 code	CPU0
0040_0000	0047_FFFF	512 KB	SRAM P14 alias	CPU0 code	CPU0
0048_0000	004F_FFFF	512 KB	SRAM P15 alias	CPU0 code	CPU0
0050_0000	0053_FFFF	256 KB	SRAM P16 alias	CPU0 code	CPU0
0054_0000	0057_FFFF	256 KB	SRAM P17 alias	CPU0 code	CPU0
0058_0000	0058_7FFF	32 KB	SRAM P18 alias	CPU0 code	CPU0, CPU1, HiFi1
0058_8000	0058_FFFF	32 KB	SRAM P19 alias	CPU0 code	CPU0, CPU1, HiFi1

Table continues on the next page...

Table 8. System memory map (continued)

Start address	End address	Size	Allocation	Cached	Access[1]
0059_0000	0059_7FFF	32 KB	SRAM P20 alias	CPU0 code	CPU0, CPU1, HiFi1
0059_8000	0059_FFFF	32 KB	SRAM P21 alias	CPU0 code	CPU0, CPU1, HiFi1
005A_0000	005A_FFFF	64 KB	SRAM P22 alias	CPU0 code	CPU0, CPU1, HiFi1
005B_0000	005B_FFFF	64 KB	SRAM P23 alias	CPU0 code	CPU0, CPU1, HiFi1
005C_0000	005D_FFFF	128 KB	SRAM P24 alias	CPU0 code	CPU0, CPU1, HiFi1
005E_0000	005F_FFFF	128 KB	SRAM P25 alias	CPU0 code	CPU0, CPU1, HiFi1
0060_0000	0067_FFFF	512 KB	SRAM P26 alias	CPU0 code	CPU0, CPU1, HiFi1
0068_0000	006F_FFFF	512 KB	SRAM P27 alias	CPU0 code	CPU0, CPU1, HiFi1
0070_0000	0073_FFFF	256 KB	SRAM P28 alias	CPU0 code	CPU0, CPU1, HiFi1
0074_0000	0077_FFFF	256 KB	SRAM P29 alias	CPU0 code	CPU0, CPU1, HiFi1
0078_0000	02FF_FFFF	40.5 MB	Reserved	—	—
0304_0000	07FF_FFFF	79.75 MB	Reserved	—	—
0800_0000	0FFF_FFFF	128 MB	XSPI1	CACHE64_CTRL1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1, Media domain controllers
2000_0000	2000_7FFF	32 KB	SRAM P0	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2000_8000	2000_FFFF	32 KB	SRAM P1	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2001_0000	2001_7FFF	32 KB	SRAM P2	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2001_8000	2001_FFFF	32 KB	SRAM P3	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2002_0000	2002_FFFF	64 KB	SRAM P4	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2003_0000	2003_FFFF	64 KB	SRAM P5	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]

Table continues on the next page...

Table 8. System memory map (continued)

Start address	End address	Size	Allocation	Cached	Access[1]
2004_0000	2005_FFFF	128 KB	SRAM P6	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2006_0000	2007_FFFF	128 KB	SRAM P7	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2008_0000	200B_FFFF	256 KB	SRAM P8	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
200C_0000	200F_FFFF	256 KB	SRAM P9	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2010_0000	2017_FFFF	512 KB	SRAM P10	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2018_0000	201F_FFFF	512 KB	SRAM P11	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2020_0000	202F_FFFF	1 MB	SRAM P12	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2030_0000	203F_FFFF	1 MB	SRAM P13	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2040_0000	2047_FFFF	512 KB	SRAM P14	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2048_0000	204F_FFFF	512 KB	SRAM P15	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2050_0000	2053_FFFF	256 KB	SRAM P16	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]

Table continues on the next page...

Table 8. System memory map (continued)

Start address	End address	Size	Allocation	Cached	Access[1]
2054_0000	2057_FFFF	256 KB	SRAM P17	CPU0 system	CPU0, CPU1, HiFi4, NPU, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2058_0000	2058_7FFF	32 KB	SRAM P18	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2058_8000	2058_FFFF	32 KB	SRAM P19	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2059_0000	2059_7FFF	32 KB	SRAM P20	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2059_8000	2059_FFFF	32 KB	SRAM P21	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
205A_0000	205A_FFFF	64 KB	SRAM P22	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
205B_0000	205B_FFFF	64 KB	SRAM P23	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
205C_0000	205D_FFFF	128 KB	SRAM P24	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
205E_0000	205F_FFFF	128 KB	SRAM P25	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2060_0000	2067_FFFF	512 KB	SRAM P26	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2068_0000	206F_FFFF	512 KB	SRAM P27	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]

Table continues on the next page...

Table 8. System memory map (continued)

Start address	End address	Size	Allocation	Cached	Access[1]
2070_0000	2073_FFFF	256 KB	SRAM P28	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2074_0000	2077_FFFF	256 KB	SRAM P29	CPU0 system	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers[2]
2078_0000	23FF_FFFF	56.5 MB	Reserved	—	—
2400_0000	2400_FFFF	64 KB	HiFi4_DTCM	No	CPU0, eDMA0, eDMA1
2401_0000	2401_FFFF	64 KB	Reserved	—	—
2402_0000	2402_FFFF	64 KB	HiFi4_ITCM	No	CPU0, eDMA0, eDMA1
2403_0000	2403_FFFF	64 KB	Reserved	—	—
2404_0000	240F_FFFF	768 KB	Reserved	—	—
2410_0000	2410_7FFF	32 KB	EZHV_ITCM	No	CPU0, CPU1, Media domain controllers
2410_8000	2410_FFFF	32 KB	EZHV_DTCM	No	CPU0, CPU1, Media domain controllers
2411_2000	27FF_FFFF	82.9297 MB	Reserved	—	—
2800_0000	2FFF_FFFF	128 MB	XSPI0	CACHE64_CTRL0	CPU0, HiFi4, eDMA0, eDMA1, Media domain controllers
4000_0000	4FFF_FFFF		PBRIDGE	No	See peripheral memory map
4052_0000	4052_1FFF	8 KB	NPU		CPU0, CPU1, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers
6000_0000	6FFF_FFFF	256 MB	XSPI2	No	CPU0, CPU1, HiFi4, eDMA0, eDMA1, eDMA2, eDMA3, Media domain controllers
E000_0000	E003_FFFF	256 KB	PPB_INT	No	CPU0, CPU1
E004_0000	E00F_FFFF	768 KB	PPB_EXT	No	CPU0, CPU1

1. Media domain controllers include uSDHC0, uSDHC1, USB0, USB1, LCDIF, MIPI_DSI_HOST, JPEGDEC, PNGDEC, VGPU.
2. Ensure only one initiator accesses the Shared RAM partition at a time for better performance.

3.3 Peripheral memory map

The peripheral memory map represents the peripheral slot assignments and the access rights for all processors. The memory slots in the table represent the non-secure address region.

NOTE

See i.MX RT700 Security Reference Manual for secure address region and secure peripheral slot assignments and the access rights for all processors.

Table 9. Peripheral memory map

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
4000_0000	4000_0FFF	4	RSTCTL0	Y	Y	—	—	Y
4000_1000	4000_1FFF	4	CLKCTL0	Y	Y	—	—	Y
4000_2000	4000_2FFF	4	SYSCON0	Y	Y	—	—	Y
4000_3000	4000_3FFF	4	SLEEPCON0	Y	Y	—	—	Y
4000_4000_	4000_4FFF	4	IOPCTL0	Y	Y	—	—	Y
4000_5000_	4000_5FFF	4	PVTS0	Y	Y	—	—	Y
4000_E000	4000_EFFF	4	WWDT0	Y	Y	—	—	Y
4000_F000	4000_FFFF	4	UTICK0	Y	Y	—	—	Y
4001_8000	4001_8FFF	4	OCOTP	Y	Y	—	—	Y
4002_3000	4002_3FFF	4	FREQME	Y	Y	—	—	Y
4002_5000	4002_5FFF	4	PINT0	Y	Y	—	—	Y
4002_6000	4002_6FFF	4	INPUTMUX0	Y	Y	—	—	Y
4002_8000	4002_8FFF	4	CTIMER0	Y	Y	—	—	Y
4002_9000	4002_9FFF	4	CTIMER1	Y	Y	—	—	Y
4002_A000	4002_AFFF	4	CTIMER2	Y	Y	—	—	Y
4002_B000	4002_BFFF	4	CTIMER3	Y	Y	—	—	Y
4002_C000	4002_CFFF	4	CTIMER4	Y	Y	—	—	Y
4002_D000	4002_DFFF	4	MRT0	Y	Y	—	—	Y
4002_E000	4002_EFFF	4	WWDT1	Y	Y	—	—	Y
4003_0000	4003_0FFF	4	MMU0	Y	Y	—	—	Y
4003_1000	4003_1FFF	4	MMU1	Y	Y	—	—	Y
4003_3000	4003_3FFF	4	XCACHE0 (CPU0 system bus)	Y	Y	—	—	Y
4003_4000	4003_4FFF	4	XCACHE1 (CPU0 code bus)	Y	Y	—	—	Y
4003_5000	4003_5FFF	4	CACHE64_CTRL0/ CACHE64_POLSEL0	Y	Y	—	—	Y
4003_6000	4003_6FFF	4	CACHE64_CTRL1/ CACHE64_POLSEL1	Y	Y	—	—	Y
4003_7000	4003_7FFF	4	I3C0	Y	Y	—	—	Y

Table continues on the next page...

Table 9. Peripheral memory map (continued)

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
4003_8000	4003_8FFF	4	I3C1	Y	Y	—	—	Y
4004_0000	4004_0FFF	4	RSTCTL1	—	—	Y	Y	Y
4004_1000	4004_1FFF	4	CLKCTL1	—	—	Y	Y	Y
4004_2000	4004_2FFF	4	SYSCON1	—	—	Y	Y	Y
4004_3000	4004_3FFF	4	PVTS1	—	—	Y	Y	Y
4004_4000	4004_4FFF	4	SLEEPCON1	—	—	Y	Y	Y
4004_5000	4004_5FFF	4	PINT1	—	—	Y	Y	Y
4004_6000	4004_6FFF	4	INPUTMUX1	—	—	Y	Y	Y
4004_8000	4004_8FFF	4	CTIMER5	—	—	Y	Y	Y
4004_9000	4004_9FFF	4	CTIMER6	—	—	Y	Y	Y
4004_A000	4004_AFFF	4	CTIMER7	—	—	Y	Y	Y
4004_D000	4004_DFFF	4	MRT1	—	—	Y	Y	Y
4005_0000	4005_0FFF	4	WWDT2	—	—	Y	Y	Y
4005_1000	4005_1FFF	4	WWDT3	—	—	Y	Y	Y
4005_2000	4005_2FFF	4	UTICK1	—	—	Y	Y	Y
4005_6000	4005_6FFF	4	I3C2	—	—	Y	Y	Y
4005_7000	4005_7FFF	4	I3C3	—	—	Y	Y	Y
4006_0000	4006_0FFF	4	RSTCTL3	Y	Y	Y	Y	Y
4006_1000	4006_1FFF	4	CLKCTL3	Y	Y	Y	Y	Y
4006_2000	4006_2FFF	4	SYSCON3	Y	Y	Y	Y	Y
4006_3000	4006_3FFF	4	OSC32KNP	Y	Y	Y	Y	Y
4006_4000	4006_4FFF	4	IOPCTL1	Y	Y	Y	Y	Y
4006_5000	4006_5FFF	4	CLKCTL2	Y	Y	Y	Y	Y
4006_6000	4006_6FFF	4	SYSCON2	Y	Y	Y	Y	Y
4006_7000	4006_7FFF	4	RSTCTL2	Y	Y	Y	Y	Y
4006_8000	4006_8FFF	4	RTC0 (LP)	Y	Y	—	—	—
4006_8000	4006_8FFF	4	RTC_SS0 (LP)	Y	Y	—	—	—
4006_9000	4006_9FFF	4	RTC1 (LP)	—	—	Y	Y	Y
4006_9000	4006_9FFF	4	RTC_SS1 (LP)	—	—	Y	Y	Y
400A_0000	400A_0FFF	4	RSTCTL4	Y	Y	Y	—	Y
400A_1000	400A_1FFF	4	CLKCTL4	Y	Y	Y	—	Y

Table continues on the next page...

Table 9. Peripheral memory map (continued)

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
400A_2000	400A_2FFF	4	SYSCON4	Y	Y	Y	—	Y
400A_5000	400A_5FFF	4	IOPCTL2	Y	Y	Y	—	Y
400A_6000	400A_6FFF	4	JPEGDECWRP/ JPEGDEC	Y	Y	Y	—	Y
400A_7000	400A_7FFF	4	PNGDEC	Y	Y	Y	—	Y
400A_8000	400A_8FFF	4	MMU2	Y	Y	Y	—	Y
4010_0000	4010_0FFF	4	GPIO0	Y	Y	—	—	Y
4010_1000	4010_1FFF	4	GPIO0_ALIAS	Y	Y	—	—	Y
4010_2000	4010_2FFF	4	GPIO1	Y	Y	—	—	Y
4010_3000	4010_3FFF	4	GPIO1_ALIAS	Y	Y	—	—	Y
4010_4000	4010_4FFF	4	GPIO2	Y	Y	—	—	Y
4010_5000	4010_5FFF	4	GPIO2_ALIAS	Y	Y	—	—	Y
4010_6000	4010_6FFF	4	GPIO3	Y	Y	—	—	Y
4010_7000	4010_7FFF	4	GPIO3_ALIAS	Y	Y	—	—	Y
4010_8000	4010_8FFF	4	GPIO4	Y	Y	—	—	Y
4010_9000	4010_9FFF	4	GPIO4_ALIAS	Y	Y	—	—	Y
4010_A000	4010_AFFF	4	GPIO5	Y	Y	—	—	Y
4010_B000	4010_BFFF	4	GPIO5_ALIAS	Y	Y	—	—	Y
4010_C000	4010_CFFF	4	GPIO6	Y	Y	—	—	Y
4010_D000	4010_DFFF	4	GPIO6_ALIAS	Y	Y	—	—	Y
4010_E000	4010_EFFF	4	GPIO7	Y	Y	—	—	Y
4010_F000	4010_FFFF	4	GPIO7_ALIAS	Y	Y	—	—	Y
4011_0000	4011_0FFF	4	LP_FLEXCOMM 0	Y	Y	—	—	Y
4011_1000	4011_1FFF	4	LP_FLEXCOMM 1	Y	Y	—	—	Y
4011_2000	4011_2FFF	4	LP_FLEXCOMM 2	Y	Y	—	—	Y
4011_3000	4011_3FFF	4	LP_FLEXCOMM 3	Y	Y	—	—	Y
4011_4000	4011_4FFF	4	SCT0	Y	Y	—	—	Y
4014_0000	4015_0FFF	68	eDMA0	Y	Y	—	—	Y
4015_1000	4015_1FFF	4	CRC	Y	Y	—	—	Y

Table continues on the next page...

Table 9. Peripheral memory map (continued)

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
4015_2000	4015_2FFF	4	SAI0	Y	Y	—	—	Y
4015_3000	4015_3FFF	4	SAI1	Y	Y	—	—	Y
4015_4000	4015_4FFF	4	SAI2	Y	Y	—	—	Y
4016_0000	4017_0FFF	68	eDMA1	Y	Y	—	—	Y
4017_1000	4017_1FFF	4	LP_FLEXCOMM4	Y	Y	—	—	Y
4017_2000	4017_2FFF	4	LP_FLEXCOMM5	Y	Y	—	—	Y
4017_3000	4017_3FFF	4	LP_FLEXCOMM6	Y	Y	—	—	Y
4017_4000	4017_4FFF	4	LP_FLEXCOMM7	Y	Y	—	—	Y
4017_5000	4017_5FFF	4	NPU	Y	Y	—	—	Y
4018_4000	4018_4FFF	4	XSPI0 (system domain)	Y	Y	—	—	Y
4018_5000	4018_5FFF	4	XSPI1 (system domain)	Y	Y	—	—	Y
4018_8000	4018_8FFF	4	MU4.MUB	—	Y	—	—	—
4018_9000	4018_9FFF	4	MU4.MUA	Y	—	—	—	—
4018_A000	4018_AFFF	4	SEMA42_4	Y	Y	—	—	Y
4018_C000	4018_CFFF	4	CMX_PERFMON0 for XCACHE1	Y	Y	—	—	Y
4018_D000	4018_DFFF	4	CMX_PERFMON1 for XCACHE0	Y	Y	—	—	Y
4018_E000	4018_EFFF	4	CMX_PERFMON2 for CACHE64_CTRL0	Y	Y	—	—	Y
4018_F000	4018_FFFF	4	CMX_PERFMON3 for CACHE64_CTRL1	Y	Y	—	—	Y
4019_9000	4019_9FFF	4	LP_FLEXCOMM8	Y	Y	—	—	Y
4019_A000	4019_AFFF	4	LP_FLEXCOMM9	Y	Y	—	—	Y

Table continues on the next page...

Table 9. Peripheral memory map (continued)

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
4019_B000	4019_BFFF	4	LP_FLEXCOMM 10	Y	Y	—	—	Y
4019_C000	4019_CFFF	4	LP_FLEXCOMM 11	Y	Y	—	—	Y
4019_D000	4019_DFFF	4	LP_FLEXCOMM 12	Y	Y	—	—	Y
4019_E000	4019_EFFF	4	LP_FLEXCOMM 13	Y	Y	—	—	Y
4020_0000	4020_0FFF	4	MU0.MUA	Y	—	—	—	—
4020_1000	4020_1FFF	4	MU0.MUB	—	—	—	Y	—
4020_2000	4020_2FFF	4	MU1.MUA	Y	—	—	—	—
4020_3000	4020_3FFF	4	MU1.MUB	—	—	Y	—	—
4020_4000	4020_4FFF	4	MU2.MUA	—	Y	—	—	—
4020_5000	4020_5FFF	4	MU2.MUB	—	—	Y	—	—
4020_6000	4020_6FFF	4	SEMA42_0	Y	Y	Y	Y	—
4020_7000	4020_7FFF	4	OSTIMER (CPU0)	Y	—	—	—	—
4020_8000	4020_8FFF	4	OSTIMER Alias1 (HiFi4)	—	Y	—	—	—
4020_9000	4020_9FFF	4	OSTIMER Alias2 (CPU1)	—	—	Y	—	—
4020_A000	4020_AFFF	4	OSTIMER Alias3 (HiFi1)	—	—	—	Y	—
4020_B000	4020_BFFF	4	ACMP0	Y	Y	Y	Y	—
4020_C000	4020_CFFF	4	ADC0	Y	Y	Y	Y	—
4020_D000	4020_DFFF	4	SDADC	Y	Y	Y	Y	—
4020_E000	4020_EFFF	4	MICFIL	Y	Y	Y	Y	—
4020_F000	4020_FFFF	4	PMC0	Y	—	—	—	—
4021_0000	4021_0FFF	4	PMC1	—	—	Y	—	—
4021_3000	4021_3FFF	4	LPI2C15	Y	Y	Y	Y	—
4030_0000	4030_8FFF	36	eDMA2	—	—	Y	Y	Y
4031_0000	4031_8FFF	36	eDMA3	—	—	Y	Y	Y
4031_9000	4031_9FFF	4	MU3.MUA	—	—	Y	—	—
4031_A000	4031_AFFF	4	MU3.MUB	—	—	—	Y	—

Table continues on the next page...

Table 9. Peripheral memory map (continued)

Start address	End address	Size (KB)	Allocation	Processor access to resource				
				CPU0[1]	HiFi4	CPU1[2]	HiFi1	EZH-V
4031_B000	4031_BFFF	4	SEMA42_3	—	—	Y	Y	—
4031_C000	4031_CFFF	4	SAI3	—	—	Y	Y	—
4032_0000	4032_0FFF	4	GPIO8	—	—	Y	Y	—
4032_1000	4032_1FFF	4	GPIO8_ALIAS	—	—	Y	Y	—
4032_2000	4032_2FFF	4	GPIO9	—	—	Y	Y	—
4032_3000	4032_3FFF	4	GPIO9_ALIAS	—	—	Y	Y	—
4032_4000	4032_4FFF	4	GPIO10	—	—	Y	Y	—
4032_5000	4032_5FFF	4	GPIO10_ALIAS	—	—	Y	Y	—
4032_6000	4032_6FFF	4	LP_FLEXCOMM 17	—	—	Y	Y	—
4032_7000	4032_7FFF	4	LP_FLEXCOMM 18	—	—	Y	Y	—
4032_8000	4032_8FFF	4	LP_FLEXCOMM 19	—	—	Y	Y	—
4032_9000	4032_9FFF	4	LP_FLEXCOMM 20	—	—	Y	Y	—
4040_5000	4040_5FFF	4	LPSPI16	Y	Y	Y	—	Y
4041_1000	4041_1FFF	4	XSPI2 (system domain)	Y	Y	Y	—	Y
4041_2000	4041_2FFF	4	uSDHC0	Y	Y	Y	—	Y
4041_3000	4041_3FFF	4	uSDHC1	Y	Y	Y	—	Y
4041_4000	4041_4FFF	4	USBPHY/ USBDCD	Y	Y	Y	—	Y
4041_6000	4041_6FFF	4	FLEXIO	Y	Y	Y	—	Y
4041_7000	4041_7FFF	4	MIPI_DSI_HOST	Y	Y	Y	—	Y
4041_8000	4041_8FFF	4	USB0	Y	Y	Y	—	Y
4041_9000	4041_9FFF	4	USB1	Y	Y	Y	—	Y
4048_0000	4048_3FFF	16	LCDIF	Y	Y	Y	—	Y
4048_4000	4048_4FFF	4	LPSPI14	Y	Y	Y	—	Y
4048_6000	4048_6FFF	4	VGPU	Y	Y	Y	—	Y
404C_0000	404F_FFFF	256	NIC_MEDIA1	Y	Y	Y	—	Y

1. Peripherals that can be accessed by CPU0 can also be accessed by eDMA0 and eDMA1.
2. Peripherals that can be accessed by CPU1 can also be accessed by eDMA2 and eDMA3.

Chapter 4
Interrupt and DMA Events Assignments

4.1 Overview

This chapter provides information on the assignments of the interrupts and DMA.

4.2 System interrupts

The Nested Vectored Interrupt Controller (NVIC) collects interrupt request sources and provides an interface to the Cortex-M cores (CPU0 and CPU1) and DSP cores (HiFi1 and HiFi4).

NOTE

The NVIC IRQ assignments on secure modules are not described here, see i.MX RT700 Security Reference Manual for details.

Table 10. CPU0 NVIC IRQ assignments

Vector offset address	IRQ	Module	Description
0x0000_0040	0	-	Reserved
0x0000_0044	1	UTICK0	UTICK0 interrupt
0x0000_0048	2	MRT0	MRT0 ored interrupt (repeat interrupt, shot interrupt or shot bus stall interrupt)
0x0000_004C	3	CTIMER0	CTIMER0 interrupt
0x0000_0050	4	CTIMER1	CTIMER1 interrupt
0x0000_0054	5	SCT0	SCT0 interrupt
0x0000_0058	6	CTIMER3	CTIMER3 interrupt
0x0000_005C	7	LP_FLEXCOMM0	LP_FLEXCOMM0 interrupt
0x0000_0060	8	LP_FLEXCOMM1	LP_FLEXCOMM1 interrupt
0x0000_0064	9	LP_FLEXCOMM2	LP_FLEXCOMM2 interrupt
0x0000_0068	10	LP_FLEXCOMM3	LP_FLEXCOMM3 interrupt
0x0000_006C	11	LP_FLEXCOMM4	LP_FLEXCOMM4 interrupt
0x0000_0070	12	LP_FLEXCOMM5	LP_FLEXCOMM5 interrupt
0x0000_0074	13	LPSPI14	LPSPI14 interrupt
0x0000_0078	14	LPI2C15	LPI2C15 interrupt
0x0000_007C	15	ADC0	ADC0 interrupt
0x0000_0080	16	SDADC	SDADC interrupt
0x0000_0084	17	ACMP0	ACMP0 interrupt
0x0000_0088	18	MICFIL	MICFIL interrupt for read data or error
0x0000_008C	19	MICFIL	MICFIL hardware voice activity detector interrupt or error interrupt

Table continues on the next page...

Table 10. CPU0 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0090	20	—	Reserved
0x0000_0094	21	—	Reserved
0x0000_0098	22	—	Reserved
0x0000_009C	23	—	Reserved
0x0000_00A0	24	—	Reserved
0x0000_00A4	25	—	Reserved
0x0000_00A8	26	RTC0	RTC0 alarm interrupt to computer domain
0x0000_00AC	27	RTC0	RTC0 wakeup interrupt to compute domain
0x0000_00B0	28	HiFi4	HiFi4 interrupt
0x0000_00B4	29	MU0	MU0 MUA, CPU0 to HiFi1
0x0000_00B8	30	MU1	MU1 MUA, CPU0 to CPU1
0x0000_00BC	31	MU4	MU4 MUA, CPU0 to HiFi4
0x0000_00C0	32	CTIMER2	CTIMER2 interrupt
0x0000_00C4	33	CTIMER4	CTIMER4 interrupt
0x0000_00C8	34	OSTIMER	OSTIMER CPU0 wakeup/interrupt
0x0000_00CC	35	LP_FLEXCOMM6	LP_FLEXCOMM6 interrupt
0x0000_00D0	36	LP_FLEXCOMM7	LP_FLEXCOMM7 interrupt
0x0000_00D4	37	uSDHC0	uSDHC0 interrupt
0x0000_00D8	38	uSDHC1	uSDHC1 interrupt
0x0000_00DC	39	I3C0	I3C0 interrupt
0x0000_00E0	40	USB0	USB0 interrupt
0x0000_00E4	41	USB1	USB1 interrupt
0x0000_00E8	42	WWDT0	WWDT0 interrupt
0x0000_00EC	43	WWDT1	WWDT1 interrupt
0x0000_00F0	44	USBPHY0	USBPHY UTM interrupt
0x0000_00F4	45	—	Reserved
0x0000_00F8	46	—	Reserved
0x0000_00FC	47	LP_FLEXCOMM8	LP_FLEXCOMM8 interrupt
0x0000_0100	48	LP_FLEXCOMM9	LP_FLEXCOMM9 interrupt
0x0000_0104	49	LP_FLEXCOMM10	LP_FLEXCOMM10 interrupt
0x0000_0108	50	LP_FLEXCOMM11	LP_FLEXCOMM11 interrupt
0x0000_010C	51	LP_FLEXCOMM12	LP_FLEXCOMM12 interrupt

Table continues on the next page...

Table 10. CPU0 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0110	52	LP_FLEXCOMM13	LP_FLEXCOMM13 interrupt
0x0000_0114	53	LPSPI16	LPSPI16 interrupt
0x0000_0118	54	I3C1	I3C1 interrupt
0x0000_011C	55	FLEXIO	FLEXIO interrupt
0x0000_0120	56	LCDIF	LCDIF interrupt
0x0000_0124	57	VGPU	VGPU interrupt from graphics core
0x0000_0128	58	MIPI_DSI_HOST	MIPI_DSI_HOST interrupt
0x0000_012C	59	eDMA0	eDMA0 channel 0 interrupt
0x0000_0130	60	eDMA0	eDMA0 channel 1 interrupt
0x0000_0134	61	eDMA0	eDMA0 channel 2 interrupt
0x0000_0138	62	eDMA0	eDMA0 channel 3 interrupt
0x0000_013C	63	eDMA0	eDMA0 channel 4 interrupt
0x0000_0140	64	eDMA0	eDMA0 channel 5 interrupt
0x0000_0144	65	eDMA0	eDMA0 channel 6 interrupt
0x0000_0148	66	eDMA0	eDMA0 channel 7 interrupt
0x0000_014C	67	eDMA0	eDMA0 channel 8 interrupt
0x0000_0150	68	eDMA0	eDMA0 channel 9 interrupt
0x0000_0154	69	eDMA0	eDMA0 channel 10 interrupt
0x0000_0158	70	eDMA0	eDMA0 channel 11 interrupt
0x0000_015C	71	eDMA0	eDMA0 channel 12 interrupt
0x0000_0160	72	eDMA0	eDMA0 channel 13 interrupt
0x0000_0164	73	eDMA0	eDMA0 channel 14 interrupt
0x0000_0168	74	eDMA0	eDMA0 channel 15 interrupt
0x0000_016C	75	eDMA1	eDMA1 channel 0 interrupt
0x0000_0170	76	eDMA1	eDMA1 channel 1 interrupt
0x0000_0174	77	eDMA1	eDMA1 channel 2 interrupt
0x0000_0178	78	eDMA1	eDMA1 channel 3 interrupt
0x0000_017C	79	eDMA1	eDMA1 channel 4 interrupt
0x0000_0180	80	eDMA1	eDMA1 channel 5 interrupt
0x0000_0184	81	eDMA1	eDMA1 channel 6 interrupt
0x0000_0188	82	eDMA1	eDMA1 channel 7 interrupt
0x0000_018C	83	eDMA1	eDMA1 channel 8 interrupt

Table continues on the next page...

Table 10. CPU0 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0190	84	eDMA1	eDMA1 channel 9 interrupt
0x0000_0194	85	eDMA1	eDMA1 channel 10 interrupt
0x0000_0198	86	eDMA1	eDMA1 channel 11 interrupt
0x0000_019C	87	eDMA1	eDMA1 channel 12 interrupt
0x0000_01A0	88	eDMA1	eDMA1 channel 13 interrupt
0x0000_01A4	89	eDMA1	eDMA1 channel 14 interrupt
0x0000_01A8	90	eDMA1	eDMA1 channel 15 interrupt
0x0000_01AC	91	GPIO0	GPIO0 channel 0 interrupt
0x0000_01B0	92	GPIO0	GPIO0 channel 1 interrupt
0x0000_01B4	93	GPIO1	GPIO1 channel 0 interrupt
0x0000_01B8	94	GPIO1	GPIO1 channel 1 interrupt
0x0000_01BC	95	GPIO2	GPIO2 channel 0 interrupt
0x0000_01C0	96	GPIO2	GPIO2 channel 1 interrupt
0x0000_01C4	97	GPIO3	GPIO3 channel 0 interrupt
0x0000_01C8	98	GPIO3	GPIO3 channel 1 interrupt
0x0000_01CC	99	GPIO4	GPIO4 channel 0 interrupt
0x0000_01D0	100	GPIO4	GPIO4 channel 1 interrupt
0x0000_01D4	101	GPIO5	GPIO5 channel 0 interrupt
0x0000_01D8	102	GPIO5	GPIO5 channel 1 interrupt
0x0000_01DC	103	GPIO6	GPIO6 channel 0 interrupt
0x0000_01E0	104	GPIO6	GPIO6 channel 1 interrupt
0x0000_01E4	105	GPIO7	GPIO7 channel 0 interrupt
0x0000_01E8	106	GPIO7	GPIO7 channel 1 interrupt
0x0000_01EC	107	PINT0	PINT0 interrupt 0
0x0000_01F0	108	PINT0	PINT0 interrupt 1
0x0000_01F4	109	PINT0	PINT0 interrupt 2
0x0000_01F8	110	PINT0	PINT0 interrupt 3
0x0000_01FC	111	PINT0	PINT0 interrupt 4
0x0000_0200	112	PINT0	PINT0 interrupt 5
0x0000_0204	113	PINT0	PINT0 interrupt 6
0x0000_0208	114	PINT0	PINT0 interrupt 7
0x0000_020C	115	SAI0	SAI0 TX/RX interrupt

Table continues on the next page...

Table 10. CPU0 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0210	116	SAI1	SAI1 TX/RX interrupt
0x0000_0214	117	SAI2	SAI2 TX/RX interrupt
0x0000_0218	118	XSPI0	XSPI0 Ored interrupt and flash error interrupt
0x0000_021C	119	XSPI1	XSPI1 Ored interrupt and flash error interrupt
0x0000_0220	120	XSPI2	XSPI2 Ored interrupt and flash error interrupt
0x0000_0224	121	MMU0	MMU0 interrupt
0x0000_0228	122	MMU1	MMU1 interrupt
0x0000_022C	123	MMU2	MMU2 interrupt
0x0000_0230	124	FREQME	FREQME interrupt
0x0000_0234	125	—	Reserved
0x0000_0238	126	—	Reserved
0x0000_023C	127	—	Reserved
0x0000_0240	128	—	Reserved
0x0000_0244	129	—	Reserved
0x0000_0248	130	—	Reserved
0x0000_024C	131	—	Reserved
0x0000_0250	132	—	Reserved
0x0000_0254	133	—	Reserved
0x0000_0258	134	—	Reserved
0x0000_025C	135	—	Reserved
0x0000_0260	136	SLEEPCON0	CPU0 wakeup event
0x0000_0264	137	CPU0	CTI interrupt (with BIST interrupt)
0x0000_0268	138	PMC0	PMC0 interrupt (including the source of PMIC_IRQN)
0x0000_026C	139	JPEGDEC	JPEGDEC Ored context 0-3 interrupt
0x0000_0270	140	PNGDEC	PNGDEC interrupt
0x0000_0274	141	NPU	NPU interrupt
0x0000_0278	142	—	Reserved
0x0000_027C	143	SYSCON3	TEMPDETECT0 temperature detect interrupt 0
0x0000_0280	144	SYSCON3	TEMPDETECT1 temperature detect interrupt 1
0x0000_0284	145	EZH-V	EZH-V interrupt
0x0000_0288	146	—	Reserved
0x0000_028C	147	—	Reserved

Table continues on the next page...

Table 10. CPU0 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0290	148	—	Reserved
0x0000_0294	149	—	Reserved
0x0000_0298	150	PVTS0	PVTS0 CPU0 interrupt
0x0000_029C	151	—	Reserved
0x0000_02A0	152	PVTS0	PVTS0 HiFi4 DSP core interrupt
0x0000_02A4	153	—	Reserved
0x0000_02A8	154	FRO0	FRO0 in VDD2_COM domain async interrupt
0x0000_02AC	155	FRO1	FRO1 in VDD2_COM domain async interrupt
0x0000_02B0	156	FRO2	FRO2 in VDD1_SENSE domain async interrupt
0x0000_02B4	157	OCOTP	OCOTP warning and integrity error interrupt

Table 11. CPU1 NVIC IRQ assignments

Vector offset address	IRQ	Module	Description
0x0000_0040	0	—	Reserved
0x0000_0044	1	—	Reserved
0x0000_0048	2	—	Reserved
0x0000_004C	3	FRO2	FRO2 in VDD1_SENSE domain async interrupt
0x0000_0050	4	—	Reserved
0x0000_0054	5	UTICK1	UTICK1 interrupt
0x0000_0058	6	MRT1	MRT1 Ored interrupt
0x0000_005C	7	CTIMER5	CTIMER5 interrupt
0x0000_0060	8	CTIMER6	CTIMER6 interrupt
0x0000_0064	9	CTIMER7	CTIMER7 interrupt
0x0000_0068	10	LPI2C15	LPI2C15 interrupt
0x0000_006C	11	LP_FLEXCOMM17	LP_FLEXCOMM17 interrupt
0x0000_0070	12	LP_FLEXCOMM18	LP_FLEXCOMM18 interrupt
0x0000_0074	13	LP_FLEXCOMM19	LP_FLEXCOMM19 interrupt
0x0000_0078	14	LP_FLEXCOMM20	LP_FLEXCOMM20 interrupt
0x0000_007C	15	ADC0	ADC0 interrupt
0x0000_0080	16	SDADC	SDADC interrupt
0x0000_0084	17	—	Reserved
0x0000_0088	18	ACMP0	ACMP0 interrupt
0x0000_008C	19	MICFIL	MICFIL interrupt for read data or Error

Table continues on the next page...

Table 11. CPU1 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0090	20	—	Reserved
0x0000_0094	21	—	Reserved
0x0000_0098	22	MICFIL	MICFIL hardware voice activity detector interrupt or error interrupt
0x0000_009C	23	RTC1	RTC1 alarm interrupt to sense domain
0x0000_00A0	24	RTC1	RTC1 wakeup interrupt to sense domain
0x0000_00A4	25	HiFi1	HiFi1 interrupt
0x0000_00A8	26	MU1	MU1 MUB, CPU1 to CPU0
0x0000_00AC	27	MU2	MU2 MUB, CPU1 to HiFi4
0x0000_00B0	28	MU3	MU3 MUA, CPU1 to HiFi1
0x0000_00B4	29	PMC1	PMC1 sense domain interrupt
0x0000_00B8	30	OSTIMER	OSTIMER CPU1 wakeup/interrupt
0x0000_00BC	31	uSDHC0	uSDHC0 interrupt
0x0000_00C0	32	uSDHC1	uSDHC1 interrupt
0x0000_00C4	33	I3C2	I3C2 interrupt
0x0000_00C8	34	USB0	USB0 interrupt
0x0000_00CC	35	USB1	USB1 interrupt
0x0000_00D0	36	WWDT2	WWDT2 interrupt
0x0000_00D4	37	WWDT3	WWDT3 interrupt
0x0000_00D8	38	USBPHY	USBPHY UTM interrupt
0x0000_00DC	39	—	Reserved
0x0000_00E0	40	I3C3	I3C3 interrupt
0x0000_00E4	41	FLEXIO	FLEXIO interrupt
0x0000_00E8	42	LCDIF	LCDIF interrupt
0x0000_00EC	43	VGPU	VGPU interrupt from graphics core
0x0000_00F0	44	MIPI_DSI_HOST	MIPI_DSI_HOST interrupt
0x0000_00F4	45	eDMA2	eDMA2 channel 0 interrupt
0x0000_00F8	46	eDMA2	eDMA2 channel 1 interrupt
0x0000_00FC	47	eDMA2	eDMA2 channel 2 interrupt
0x0000_0100	48	eDMA2	eDMA2 channel 3 interrupt
0x0000_0104	49	eDMA2	eDMA2 channel 4 interrupt
0x0000_0108	50	eDMA2	eDMA2 channel 5 interrupt
0x0000_010C	51	eDMA2	eDMA2 channel 6 interrupt

Table continues on the next page...

Table 11. CPU1 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0110	52	eDMA2	eDMA2 channel 7 interrupt
0x0000_0114	53	eDMA3	eDMA3 channel 0 interrupt
0x0000_0118	54	eDMA3	eDMA3 channel 1 interrupt
0x0000_011C	55	eDMA3	eDMA3 channel 2 interrupt
0x0000_0120	56	eDMA3	eDMA3 channel 3 interrupt
0x0000_0124	57	eDMA3	eDMA3 channel 4 interrupt
0x0000_0128	58	eDMA3	eDMA3 channel 5 interrupt
0x0000_012C	59	eDMA3	eDMA3 channel 6 interrupt
0x0000_0130	60	eDMA3	eDMA3 channel 7 interrupt
0x0000_0134	61	GPIO8	GPIO8 channel 0 interrupt
0x0000_0138	62	GPIO8	GPIO8 channel 1 interrupt
0x0000_013C	63	GPIO9	GPIO9 channel 0 interrupt
0x0000_0140	64	GPIO9	GPIO9 channel 1 interrupt
0x0000_0144	65	GPIO10	GPIO10 channel 0 interrupt
0x0000_0148	66	GPIO10	GPIO10 channel 1 interrupt
0x0000_014C	67	—	Reserved
0x0000_0150	68	—	Reserved
0x0000_0154	69	PINT1	PINT1 interrupt 0
0x0000_0158	70	PINT1	PINT1 interrupt 1
0x0000_015C	71	PINT1	PINT1 interrupt 2
0x0000_0160	72	PINT1	PINT1 interrupt 3
0x0000_0164	73	SAI3	SAI3 TX/RX interrupt
0x0000_0168	74	XSPI2	XSPI2 Ored interrupt and flash error interrupt
0x0000_016C	75	MMU2	MMU2 interrupt
0x0000_0170	76	—	Reserved
0x0000_0174	77	—	Reserved
0x0000_0178	78	—	Reserved
0x0000_017C	79	—	Reserved
0x0000_0180	80	—	Reserved
0x0000_0184	81	CPU1	CTI interrupt
0x0000_0188	82	JPEGDEC	JPEGDEC Ored context 0-3 interrupt
0x0000_018C	83	PNGDEC	PNGDEC interrupt

Table continues on the next page...

Table 11. CPU1 NVIC IRQ assignments (continued)

Vector offset address	IRQ	Module	Description
0x0000_0190	84	—	Reserved
0x0000_0194	85	SYSCON3	TEMPDETECT0 temperature detect interrupt 0
0x0000_0198	86	SYSCON3	TEMPDETECT1 temperature detect interrupt 1
0x0000_019C	87	EZH-V	EZH-V interrupt
0x0000_01A0	88	SLEEPCON1	CPU1 wakeup event
0x0000_01A4	89	PVTS1	PVTS1 CPU1 interrupt
0x0000_01A8	90	—	Reserved
0x0000_01AC	91	PVTS1	PVTS1 HiFi1 DSP core interrupt
0x0000_01B0	92	—	Reserved

Table 12. HiFi4 IRQ assignments

Vector offset address	IRQ	Description	Priority level
0x0000_0040	0	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT0 register	NMI
0x0000_0044	1	HiFi4 software IRQ	Level 2
0x0000_0048	2	HiFi4 internal timer 0 IRQ	Level 2
0x0000_004C	3	HiFi4 internal timer 1 IRQ	Level 3
0x0000_0050	4	HiFi4 IRQ for performance monitoring	Level 3
0x0000_0054	5	HiFi4 IRQ for write error	Level 3
0x0000_0058	6	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT1 register	Level 1
0x0000_005C	7	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT2 register	Level 1
0x0000_0060	8	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT3 register	Level 1
0x0000_0064	9	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT4 register	Level 1
0x0000_0068	10	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT5 register	Level 1
0x0000_006C	11	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT6 register	Level 1
0x0000_0070	12	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT7 register	Level 1
0x0000_0074	13	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT8 register	Level 1

Table continues on the next page...

Table 12. HiFi4 IRQ assignments (continued)

Vector offset address	IRQ	Description	Priority level
0x0000_0078	14	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT9 register	Level 1
0x0000_007C	15	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT10 register	Level 1
0x0000_0080	16	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT11 register	Level 2
0x0000_0084	17	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT12 register	Level 2
0x0000_0088	18	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT13 register	Level 2
0x0000_008C	19	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT14 register	Level 2
0x0000_0090	20	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT15 register	Level 2
0x0000_0094	21	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT16 register	Level 2
0x0000_0098	22	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT17 register	Level 2
0x0000_009C	23	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT18 register	Level 2
0x0000_00A0	24	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT19 register	Level 3
0x0000_00A4	25	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT20 register	Level 3
0x0000_00A8	26	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT21 register	Level 3
0x0000_00AC	27	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT22 register	Level 3
0x0000_00B0	28	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT23 register	Level 3
0x0000_00B4	29	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT24 register	Level 3
0x0000_00B8	30	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT25 register	Level 3
0x0000_00BC	31	Interrupt selected by INPUTMUX0_HIFI4_INTERRUPT26 register	Level 3

Table 13. HiFi1 IRQ assignments

Vector offset address	IRQ	Description	Priority level
0x0000_0040	0	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT0 register	NMI
0x0000_0044	1	HiFi1 software IRQ	Level 2
0x0000_0048	2	HiFi1 internal timer 0 IRQ	Level 2
0x0000_004C	3	HiFi1 internal timer 1 IRQ	Level 3
0x0000_0050	4	HiFi1 IRQ for performance monitoring	Level 3
0x0000_0054	5	HiFi1 IRQ for write error	Level 3
0x0000_0058	6	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT1 register	Level 1
0x0000_005C	7	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT2 register	Level 1
0x0000_0060	8	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT3 register	Level 1
0x0000_0064	9	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT4 register	Level 1
0x0000_0068	10	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT5 register	Level 1
0x0000_006C	11	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT6 register	Level 1
0x0000_0070	12	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT7 register	Level 1
0x0000_0074	13	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT8 register	Level 1
0x0000_0078	14	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT9 register	Level 1
0x0000_007C	15	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT10 register	Level 1
0x0000_0080	16	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT11 register	Level 2
0x0000_0084	17	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT12 register	Level 2
0x0000_0088	18	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT13 register	Level 2
0x0000_008C	19	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT14 register	Level 2
0x0000_0090	20	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT15 register	Level 2

Table continues on the next page...

Table 13. HiFi1 IRQ assignments (continued)

Vector offset address	IRQ	Description	Priority level
0x0000_0094	21	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT16 register	Level 2
0x0000_0098	22	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT17 register	Level 2
0x0000_009C	23	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT18 register	Level 2
0x0000_00A0	24	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT19 register	Level 3
0x0000_00A4	25	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT20 register	Level 3
0x0000_00A8	26	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT21 register	Level 3
0x0000_00AC	27	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT22 register	Level 3
0x0000_00B0	28	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT23 register	Level 3
0x0000_00B4	29	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT24 register	Level 3
0x0000_00B8	30	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT25 register	Level 3
0x0000_00BC	31	Interrupt selected by INPUTMUX1_HIFI1_INTERRUPT26 register	Level 3

4.3 DMA assignments

NOTE

The eDMA slot assignments on secure modules are not described here, see i.MX RT700 Security Reference Manual for details.

The tables below show the DMA request signals for the peripherals in the chip:

NOTE

All DMA requests are disabled by default. You need to enable the DMA requests of eDMA0 and eDMA1 to see them by SYSCON0.EDMA0_EN0-SYSCON0.EDMA0_EN3 and SYSCON0.EDMA1_EN0-SYSCON0.EDMA1_EN3 registers.

Table 14. eDMA0 and eDMA1 slot assignments

eDMA0/eDMA1 slots	Module	Descriptions
0	-	Reserved
1	MICFIL	MICFIL DMA request for out data read
2	XSPI0	XSPI0 RX FIFO drain DMA request

Table continues on the next page...

Table 14. eDMA0 and eDMA1 slot assignments (continued)

eDMA0/eDMA1 slots	Module	Descriptions
3	XSPI0	XSPI0 TX FIFO fill DMA request
4	XSPI1	XSPI1 RX FIFO drain DMA request
5	XSPI1	XSPI1 TX FIFO fill DMA request
6	XSPI2	XSPI2 RX FIFO drain DMA request
7	XSPI2	XSPI2 TX FIFO fill DMA request
8	PINT0	PINT0 interrupt 0
9	PINT0	PINT0 interrupt 1
10	PINT0	PINT0 interrupt 2
11	PINT0	PINT0 interrupt 3
12	PINT0	PINT0 interrupt 4
13	PINT0	PINT0 interrupt 5
14	PINT0	PINT0 interrupt 6
15	PINT0	PINT0 interrupt 7
16	CTIMER0	CTIMER0 DMA request M0
17	CTIMER0	CTIMER0 DMA request M1
18	CTIMER1	CTIMER1 DMA request M0
19	CTIMER1	CTIMER1 DMA request M1
20	CTIMER2	CTIMER2 DMA request M0
21	CTIMER2	CTIMER2 DMA request M1
22	CTIMER3	CTIMER3 DMA request M0
23	CTIMER3	CTIMER3 DMA request M1
24	CTIMER4	CTIMER4 DMA request M0
25	CTIMER4	CTIMER4 DMA request M1
26	SCT0	SCT0 DMA request A
27	SCT0	SCT0 DMA request B
28	ADC0	ADC0 FIFO request 0
29	ADC0	ADC0 FIFO request 1
30	SDADC	SDADC FIFO request 0
31	SDADC	SDADC FIFO request 1
32	SDADC	SDADC FIFO request 2
33	SDADC	SDADC FIFO request 3
34	ACMP0	ACMP0 DMA request

Table continues on the next page...

Table 14. eDMA0 and eDMA1 slot assignments (continued)

eDMA0/eDMA1 slots	Module	Descriptions
35	—	Reserved
36	—	Reserved
37	FLEXIO	FLEXIO shift register 0 or timer 0
38	FLEXIO	FLEXIO shift register 1 or timer 1
39	FLEXIO	FLEXIO shift register 2 or timer 2
40	FLEXIO	FLEXIO shift register 3 or timer 3
41	FLEXIO	FLEXIO shift register 4 or timer 4
42	FLEXIO	FLEXIO shift register 5 or timer 5
43	FLEXIO	FLEXIO shift register 6 or timer 6
44	FLEXIO	FLEXIO shift register 7 or timer 7
45	LP_FLEXCOMM0	LP_FLEXCOMM0 receive request
46	LP_FLEXCOMM0	LP_FLEXCOMM0 transmit request
47	LP_FLEXCOMM1	LP_FLEXCOMM1 receive request
48	LP_FLEXCOMM1	LP_FLEXCOMM1 transmit request
49	LP_FLEXCOMM2	LP_FLEXCOMM2 receive request
50	LP_FLEXCOMM2	LP_FLEXCOMM2 transmit request
51	LP_FLEXCOMM3	LP_FLEXCOMM3 receive request
52	LP_FLEXCOMM3	LP_FLEXCOMM3 transmit request
53	LP_FLEXCOMM4	LP_FLEXCOMM4 receive request
54	LP_FLEXCOMM4	LP_FLEXCOMM4 transmit request
55	LP_FLEXCOMM5	LP_FLEXCOMM5 receive request
56	LP_FLEXCOMM5	LP_FLEXCOMM5 transmit request
57	LP_FLEXCOMM6	LP_FLEXCOMM6 receive request
58	LP_FLEXCOMM6	LP_FLEXCOMM6 transmit request
59	LP_FLEXCOMM7	LP_FLEXCOMM7 receive request
60	LP_FLEXCOMM7	LP_FLEXCOMM7 transmit request
61	LP_FLEXCOMM8	LP_FLEXCOMM8 receive request
62	LP_FLEXCOMM8	LP_FLEXCOMM8 transmit request
63	LP_FLEXCOMM9	LP_FLEXCOMM9 receive request
64	LP_FLEXCOMM9	LP_FLEXCOMM9 transmit request
65	LP_FLEXCOMM10	LP_FLEXCOMM10 receive request
66	LP_FLEXCOMM10	LP_FLEXCOMM10 transmit request

Table continues on the next page...

Table 14. eDMA0 and eDMA1 slot assignments (continued)

eDMA0/eDMA1 slots	Module	Descriptions
67	LP_FLEXCOMM11	LP_FLEXCOMM11 receive request
68	LP_FLEXCOMM11	LP_FLEXCOMM11 transmit request
69	LP_FLEXCOMM12	LP_FLEXCOMM12 receive request
70	LP_FLEXCOMM12	LP_FLEXCOMM12 transmit request
71	LP_FLEXCOMM13	LP_FLEXCOMM13 receive request
72	LP_FLEXCOMM13	LP_FLEXCOMM13 transmit request
73	LPSPI14	LPSPI14 receive request
74	LPSPI14	LPSPI14 transmit request
75	LPSPI16	LPSPI16 receive request
76	LPSPI16	LPSPI16 transmit request
77	I3C0	I3C0 transmit request (to-bus request)
78	I3C0	I3C0 receive request (from-bus request)
79	I3C1	I3C1 transmit request (to-bus request)
80	I3C1	I3C1 receive request (from-bus request)
81	SAI0	SAI0 receive request
82	SAI0	SAI0 transmit request
83	SAI1	SAI1 receive request
84	SAI1	SAI1 transmit request
85	SAI2	SAI2 receive request
86	SAI2	SAI2 transmit request
87	GPIO0	GPIO0 DMA request 0
88	GPIO0	GPIO0 DMA request 1
89	GPIO1	GPIO1 DMA request 0
90	GPIO1	GPIO1 DMA request 1
91	GPIO2	GPIO2 DMA request 0
92	GPIO2	GPIO2 DMA request 1
93	GPIO3	GPIO3 DMA request 0
94	GPIO3	GPIO3 DMA request 1
95	GPIO4	GPIO4 DMA request 0
96	GPIO4	GPIO4 DMA request 1
97	GPIO5	GPIO5 DMA request 0
98	GPIO5	GPIO5 DMA request 1

Table continues on the next page...

Table 14. eDMA0 and eDMA1 slot assignments (continued)

eDMA0/eDMA1 slots	Module	Descriptions
99	GPIO6	GPIO6 DMA request 0
100	GPIO6	GPIO6 DMA request 1
101	GPIO7	GPIO7 DMA request 0
102	GPIO7	GPIO7 DMA request 1
103	LPI2C15	LPI2C15 receive request
104	LPI2C15	LPI2C15 transmit request

NOTE

All DMA requests are disabled by default. You need to enable the DMA requests of eDMA2 and eDMA3 to see them by SYSCON1.EDMA2_EN0-SYSCON1.EDMA2_EN1 and SYSCON1.EDMA3_EN0-SYSCON1.EDMA3_EN1 registers.

Table 15. eDMA2 and eDMA3 slot assignments

eDMA2/eDMA3 slots	Module	Descriptions
0	-	Reserved
1	MICFIL	MICFIL DMA request for out data read
2	XSPI2	XSPI2 RX FIFO drain DMA request
3	XSPI2	XSPI2 TX FIFO fill DMA request
4	PINT1	PINT1 interrupt 0
5	PINT1	PINT1 interrupt 1
6	PINT1	PINT1 interrupt 2
7	PINT1	PINT1 interrupt 3
8	CTIMER5	CTIMER5 DMA request M0
9	CTIMER5	CTIMER5 DMA request M1
10	CTIMER6	CTIMER6 DMA request M0
11	CTIMER6	CTIMER6 DMA request M1
12	CTIMER7	CTIMER7 DMA request M0
13	CTIMER7	CTIMER7 DMA request M1
14	ADC0	ADC0 FIFO request 0
15	ADC0	ADC0 FIFO request 1
16	SDADC	SDADC FIFO request 0
17	SDADC	SDADC FIFO request 1
18	SDADC	SDADC FIFO request 2
19	SDADC	SDADC FIFO request 3
20	ACMP0	ACMP0 DMA request

Table continues on the next page...

Table 15. eDMA2 and eDMA3 slot assignments (continued)

eDMA2/eDMA3 slots	Module	Descriptions
21	-	Reserved
22	-	Reserved
23	FLEXIO	FLEXIO shift register 0 or timer 0
24	FLEXIO	FLEXIO shift register 1 or timer 1
25	FLEXIO	FLEXIO shift register 2 or timer 2
26	FLEXIO	FLEXIO shift register 3 or timer 3
27	FLEXIO	FLEXIO shift register 4 or timer 4
28	FLEXIO	FLEXIO shift register 5 or timer 5
29	FLEXIO	FLEXIO shift register 6 or timer 6
30	FLEXIO	FLEXIO shift register 7 or timer 7
31	LP_FLEXCOMM17	LP_FLEXCOMM17 receive request
32	LP_FLEXCOMM17	LP_FLEXCOMM17 transmit request
33	LP_FLEXCOMM18	LP_FLEXCOMM18 receive request
34	LP_FLEXCOMM18	LP_FLEXCOMM18 transmit request
35	LP_FLEXCOMM19	LP_FLEXCOMM19 receive request
36	LP_FLEXCOMM19	LP_FLEXCOMM19 transmit request
37	LP_FLEXCOMM20	LP_FLEXCOMM20 receive request
38	LP_FLEXCOMM20	LP_FLEXCOMM20 transmit request
39	I3C2	I3C2 transmit request (to-bus request)
40	I3C2	I3C2 receive request (from-bus request)
41	I3C3	I3C3 transmit request (to-bus request)
42	I3C3	I3C3 receive request (from-bus request)
43	SAI3	SAI3 receive request
44	SAI3	SAI3 transmit request
45	GPIO8	GPIO8 DMA request 0
46	GPIO8	GPIO8 DMA request 1
47	GPIO9	GPIO9 DMA request 0
48	GPIO9	GPIO9 DMA request 1
49	GPIO10	GPIO10 DMA request 0
50	GPIO10	GPIO10 DMA request 1
51	LPI2C15	LPI2C15 receive request
52	LPI2C15	LPI2C15 transmit request

Chapter 5
Fusemap

5.1 Overview

This chapter details the various fuses and their associated configuration settings. This chapter also describes various modes and the selection of the required boot devices.

NOTE

The fuses marked as “Reserved” are reserved for NXP internal (and future) use only, or are security related. Customers must not attempt to burn these, because the IC behavior may be unpredictable. The reserved fuses can be read as either '0' or '1'. Security-related fuses are described in the Security Reference Manual (SRM) for this device.

5.1.1 Lock bits

There are two kinds of lock bits, write lock and read lock.

The write lock bits are used by the controller to control writing/programming of the fuse bits and the shadow registers (which hold the value read from the fuses). Each word can be write locked independently, which means that it can no longer be programmed. If the word is write locked, only read is permitted and the overwriting of the shadow registers is not permitted. If the word is not locked, then it can be read sensed and programmed and the shadow registers can be overwritten (to allow emulation of programming without actually blowing fuses). The supplemental array of fuses are always override locked.

Read lock bits are a specific set of fuses that give you the ability to block bus access to specific words. Read locks are on parts of the array consisting of multiple words. The bits for these are stored in the supplementary array section in the last word.

See Lock Fuses tab in attached i.MXRT700_Fusemap.xlsx spreadsheet for details about lock fuses.

5.2 Boot fusemap

This section describes the various modes and the selection of the required boot devices. A separate map is given for each and every boot device. The device select is specified by the BOOT_CFG0[1:0] fuses.

Table 16. Boot device select

Boot Device	BOOT_CFG0[1]	BOOT_CFG0[0]
Boot pins determine the source	0	0
XSPI flash	0	1
uSDHC flash	1	0
Serial boot (ISP)	1	1

NOTE

The fuses marked as “Reserved” are reserved for NXP internal (and future) use only. Customers **must not** attempt to burn these, because the IC behavior may be unpredictable. The reserved fuses can be read as either '0' or '1'. Security-related fuses are described in the Security Reference Manual (SRM) for this device.

The Fuse Word Index and Fuse Bit Index mapping for the BOOT_CFGn fuses is provided in the following table.

See "Boot Fusemap" tab in i.MXRT700_Fusemap.xlsx for details about boot fusemap.

5.3 Fusemap descriptions table

This section provides the chip fusemap descriptions.

The following two mechanism are used by the fuse block to ensure that its values are correct:

- Redundancy (RED) mode: In this case only 16 of the 32 bits are available (bits 15:0 are automatically copied to bits 31:16) and checked on read. Each pair of bits can be programmed at different times. This allows for multiple programming operations on a word, so bits can be configured at different times.
- ECC mode: The ECC is generated for the whole word. The whole word must be setup at one time and individual bits cannot be modified independently.

See "Fusemap descriptions" in i.MXRT700_Fusemap.xlsx spreadsheet for details.

5.4 Shadow register reload

All fuse words in fusemap are shadowed. Once an OTP fuse word is programmed, the value will be loaded into corresponding OTP shadow register of OCOTP after either:

- power-on reset in all life cycle states
- or warm reset when the life cycle is in-field state or after.

See Shadow Register Reload for more details.

Chapter 6
External Signals and Pin Multiplexing

6.1 Introduction

Pins have several functions available via signal multiplexing to optimize functionality in small packages. This chapter illustrates which chip's signals are multiplexed on which external pin.

NOTE

Not all the pins are available for each package. You need to configure the module functions according to the package pin availability.

6.2 Muxing options

6.2.1 Muxing options (324FOWLP)

The pinmux selection is done through the FSEL field of the IOPCTL instance associated to the corresponding PIO pad. Alternate pin functions are selected through the AMENA field for analog inputs or through the corresponding boot mode selection detailed in the Boot chapter.

Table 17. Muxing options (324FOWLP)

Module	External signal	Pad	FSEL assignment
ACMP0	CMP0_INA	PIO10_8	Alternate pin function
	CMP0_INB	PIO10_9	Alternate pin function
	CMP0_INC, SDADC_INP3	PIO10_16	Alternate pin function
	CMP0_IND, SDADC_INN3	PIO10_17	Alternate pin function
	CMP0_OUT	PIO10_5	Function 3
	CMP0_OUT	PIO8_6	Function 4
	CMP0_OUT	PIO8_19	Function 4
ADC0	ADC0_A0	PIO10_0	Alternate pin function
	ADC0_A1	PIO10_2	Alternate pin function
	ADC0_A2	PIO10_4	Alternate pin function
	ADC0_A4, SDADC_INP1	PIO10_12	Alternate pin function
	ADC0_A5, SDADC_INP2	PIO10_14	Alternate pin function
	ADC0_B0	PIO10_1	Alternate pin function
	ADC0_B1	PIO10_3	Alternate pin function
	ADC0_B2	PIO10_5	Alternate pin function
	ADC0_B4, SDADC_INN1	PIO10_13	Alternate pin function
	ADC0_B5, SDADC_INN2	PIO10_15	Alternate pin function
BOOT	ISP boot select 0	PIO1_7	Alternate pin function

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	00b - SDHC0 (eMMC or SD set in OTP) 01b - XSPI0 10b – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) 11b - XSPI1		
	ISP boot select 1 00b - SDHC0 (eMMC or SD set in OTP) 01b - XSPI0 10b – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) 11b - XSPI1	PIO1_8	Alternate pin function
CLKCTL	XTALIN	XTALIN32KHZ	Alternate pin function
	XTALOUT	XTALOUT32KHZ	Alternate pin function
	MCLK	PIO1_19	Function 2
	MCLK	PIO8_4	Function 3
	CLKIN	PIO1_9	Function 5
	CLKIN	PIO1_15	Function 5
	CLKOUT_VDD1	PIO1_2	Function 6
	CLKOUT_VDD1	PIO1_9	Function 6
	CLKOUT_VDD1	PIO1_10	Function 6
	CLKOUT_VDD2	PIO0_14	Function 6
	CLKOUT_VDD2	PIO0_22	Function 6
	LOW_FREQ_CLKOUT	PIO2_14	Function 6
	LOW_FREQ_CLKOUT_N	PIO2_15	Function 6
	MCLK	PIO0_5	Function 6
	MCLK	PIO0_21	Function 6
	XTALIN	XTALINSYS	Function 0
	XTALOUT	XTALOUTSYS	Function 0
CTIMER_C	CTIMER_C_INP10	PIO0_6	Function 4
	CTIMER_C_INP11	PIO0_7	Function 4
	CTIMER_C_INP12	PIO1_6	Function 4
	CTIMER_C_INP13	PIO1_7	Function 4
	CTIMER_C_INP14	PIO1_8	Function 4

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	CTIMER_C_INP15	PIO1_9	Function 4
	CTIMER_C_INP2	PIO0_14	Function 4
	CTIMER_C_INP3	PIO0_15	Function 4
	CTIMER_C_INP4	PIO0_21	Function 4
	CTIMER_C_INP5	PIO0_22	Function 4
	CTIMER_C_INP8	PIO1_3	Function 4
	CTIMER_C_INP9	PIO0_0	Function 4
	CTIMER_C_INP0	PIO0_10	Function 5
CTIMER_S	CTIMER_S_INP0	PIO8_0	Function 2
	CTIMER_S_INP1	PIO8_3	Function 2
	CTIMER_S_INP2	PIO8_8	Function 2
	CTIMER_S_INP3	PIO8_10	Function 2
	CTIMER_S_INP4	PIO8_16	Function 2
	CTIMER_S_INP5	PIO8_17	Function 2
	CTIMER_S_INP7	PIO8_20	Function 2
	CTIMER_S_INP9	PIO8_30	Function 2
	CTIMER_S_INP0	PIO8_5	Function 3
	CTIMER_S_INP1	PIO8_12	Function 3
	CTIMER_S_INP3	PIO8_13	Function 3
	CTIMER_S_INP4	PIO8_18	Function 3
	CTIMER_S_INP6	PIO8_19	Function 3
CTIMER0	CTIMER0_MAT0	PIO0_1	Function 4
	CTIMER0_MAT0	PIO0_9	Function 4
	CTIMER0_MAT1	PIO0_2	Function 4
	CTIMER0_MAT1	PIO0_16	Function 4
	CTIMER0_MAT2	PIO0_3	Function 4
	CTIMER0_MAT3	PIO0_4	Function 4
CTIMER1	CTIMER1_MAT0	PIO0_10	Function 4
	CTIMER1_MAT0	PIO1_4	Function 4
	CTIMER1_MAT1	PIO0_11	Function 4
	CTIMER1_MAT1	PIO1_5	Function 4
	CTIMER1_MAT1	PIO1_12	Function 4

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	CTIMER1_MAT2	PIO0_12	Function 4
	CTIMER1_MAT3	PIO0_13	Function 4
CTIMER2	CTIMER2_MAT0	PIO0_5	Function 4
	CTIMER2_MAT0	PIO0_17	Function 4
	CTIMER2_MAT1	PIO0_18	Function 4
	CTIMER2_MAT1	PIO1_10	Function 4
	CTIMER2_MAT2	PIO0_19	Function 4
	CTIMER2_MAT2	PIO1_11	Function 4
	CTIMER2_MAT3	PIO0_20	Function 4
	CTIMER2_MAT3	PIO1_13	Function 4
CTIMER3	CTIMER3_MAT0	PIO1_14	Function 4
	CTIMER3_MAT1	PIO1_15	Function 4
	CTIMER3_MAT2	PIO1_16	Function 4
	CTIMER3_MAT3	PIO1_17	Function 4
CTIMER4	CTIMER4_MAT0	PIO0_31	Function 4
	CTIMER4_MAT0	PIO1_18	Function 4
	CTIMER4_MAT1	PIO1_0	Function 4
	CTIMER4_MAT1	PIO1_19	Function 4
	CTIMER4_MAT3	PIO1_2	Function 4
CTIMER5	CTIMER5_MAT0	PIO8_1	Function 2
	CTIMER5_MAT0	PIO8_18	Function 2
	CTIMER5_MAT0	PIO8_21	Function 2
	CTIMER5_MAT1	PIO8_2	Function 2
	CTIMER5_MAT1	PIO8_19	Function 2
	CTIMER5_MAT1	PIO8_22	Function 2
	CTIMER5_MAT2	PIO8_4	Function 2
	CTIMER5_MAT2	PIO8_23	Function 2
	CTIMER5_MAT3	PIO8_5	Function 2
	CTIMER5_MAT3	PIO8_24	Function 2
CTIMER6	CTIMER6_MAT0	PIO8_6	Function 2
	CTIMER6_MAT0	PIO8_25	Function 2
	CTIMER6_MAT1	PIO8_7	Function 2

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	CTIMER6_MAT1	PIO8_26	Function 2
	CTIMER6_MAT2	PIO8_9	Function 2
	CTIMER6_MAT2	PIO8_27	Function 2
	CTIMER6_MAT3	PIO8_11	Function 2
	CTIMER6_MAT3	PIO8_28	Function 2
CTIMER7	CTIMER7_MAT0	PIO8_12	Function 2
	CTIMER7_MAT0	PIO8_31	Function 2
	CTIMER7_MAT1	PIO8_13	Function 2
	CTIMER7_MAT2	PIO8_14	Function 2
	CTIMER7_MAT3	PIO8_15	Function 2
Debug	JTAG_TCK/SWCLK	PIO8_0	Function 5
	JTAG_TD0/SWO	PIO8_3	Function 5
	JTAG_TDI	PIO8_2	Function 5
	JTAG_TMS/SWDIO	PIO8_1	Function 5
	JTAG_TRSTN	PIO8_4	Function 5
	TRACECLK	PIO1_10	Function 5
	TRACEDATA0	PIO1_11	Function 5
	TRACEDATA1	PIO1_12	Function 5
	TRACEDATA2	PIO1_13	Function 5
	TRACEDATA3	PIO1_14	Function 5
EZH	EZH_PIO0	PIO2_0	Function 4
	EZH_PIO1	PIO2_1	Function 4
	EZH_PIO10	PIO2_10	Function 4
	EZH_PIO11	PIO2_11	Function 4
	EZH_PIO12	PIO2_12	Function 4
	EZH_PIO13	PIO2_13	Function 4
	EZH_PIO14	PIO2_14	Function 4
	EZH_PIO15	PIO2_15	Function 4
	EZH_PIO16	PIO3_0	Function 4
	EZH_PIO17	PIO3_1	Function 4
	EZH_PIO18	PIO3_2	Function 4
	EZH_PIO19	PIO3_3	Function 4

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	EZH_PIO2	PIO2_2	Function 4
	EZH_PIO20	PIO3_4	Function 4
	EZH_PIO21	PIO3_5	Function 4
	EZH_PIO22	PIO3_6	Function 4
	EZH_PIO23	PIO3_7	Function 4
	EZH_PIO24	PIO3_8	Function 4
	EZH_PIO25	PIO3_9	Function 4
	EZH_PIO26	PIO3_10	Function 4
	EZH_PIO27	PIO3_11	Function 4
	EZH_PIO3	PIO2_3	Function 4
	EZH_PIO4	PIO2_4	Function 4
	EZH_PIO5	PIO2_5	Function 4
	EZH_PIO6	PIO2_6	Function 4
	EZH_PIO7	PIO2_7	Function 4
	EZH_PIO8	PIO2_8	Function 4
	EZH_PIO9	PIO2_9	Function 4
FLEXIO	FLEXIO_D0	PIO2_0	Function 5
	FLEXIO_D1	PIO2_1	Function 5
	FLEXIO_D10	PIO2_10	Function 5
	FLEXIO_D11	PIO2_11	Function 5
	FLEXIO_D12	PIO2_12	Function 5
	FLEXIO_D13	PIO2_13	Function 5
	FLEXIO_D14	PIO2_14	Function 5
	FLEXIO_D15	PIO2_15	Function 5
	FLEXIO_D2	PIO2_2	Function 5
	FLEXIO_D3	PIO2_3	Function 5
	FLEXIO_D4	PIO2_4	Function 5
	FLEXIO_D5	PIO2_5	Function 5
	FLEXIO_D6	PIO2_6	Function 5
	FLEXIO_D7	PIO2_7	Function 5
	FLEXIO_D8	PIO2_8	Function 5
	FLEXIO_D9	PIO2_9	Function 5

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
FREQME	FREQME_GPIO_B_CLK	PIO1_19	Function 3
GPIO	PIO0_0	PIO0_0	Function 0
	PIO0_1	PIO0_1	Function 0
	PIO0_10	PIO0_10	Function 0
	PIO0_11	PIO0_11	Function 0
	PIO0_12	PIO0_12	Function 0
	PIO0_13	PIO0_13	Function 0
	PIO0_14	PIO0_14	Function 0
	PIO0_15	PIO0_15	Function 0
	PIO0_16	PIO0_16	Function 0
	PIO0_17	PIO0_17	Function 0
	PIO0_18	PIO0_18	Function 0
	PIO0_19	PIO0_19	Function 0
	PIO0_2	PIO0_2	Function 0
	PIO0_20	PIO0_20	Function 0
	PIO0_21	PIO0_21	Function 0
	PIO0_22	PIO0_22	Function 0
	PIO0_3	PIO0_3	Function 0
	PIO0_31	PIO0_31	Function 0
	PIO0_4	PIO0_4	Function 0
	PIO0_5	PIO0_5	Function 0
	PIO0_6	PIO0_6	Function 0
	PIO0_7	PIO0_7	Function 0
	PIO0_9	PIO0_9	Function 0
	PIO1_0	PIO1_0	Function 0
	PIO1_10	PIO1_10	Function 0
	PIO1_11	PIO1_11	Function 0
	PIO1_12	PIO1_12	Function 0
	PIO1_13	PIO1_13	Function 0
	PIO1_14	PIO1_14	Function 0
	PIO1_15	PIO1_15	Function 0
	PIO1_16	PIO1_16	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO1_17	PIO1_17	Function 0
	PIO1_18	PIO1_18	Function 0
	PIO1_19	PIO1_19	Function 0
	PIO1_2	PIO1_2	Function 0
	PIO1_3	PIO1_3	Function 0
	PIO1_4	PIO1_4	Function 0
	PIO1_5	PIO1_5	Function 0
	PIO1_6	PIO1_6	Function 0
	PIO1_7	PIO1_7	Function 0
	PIO1_8	PIO1_8	Function 0
	PIO1_9	PIO1_9	Function 0
	PIO10_0	PIO10_0	Function 0
	PIO10_1	PIO10_1	Function 0
	PIO10_12	PIO10_12	Function 0
	PIO10_13	PIO10_13	Function 0
	PIO10_14	PIO10_14	Function 0
	PIO10_15	PIO10_15	Function 0
	PIO10_16	PIO10_16	Function 0
	PIO10_17	PIO10_17	Function 0
	PIO10_2	PIO10_2	Function 0
	PIO10_3	PIO10_3	Function 0
	PIO10_4	PIO10_4	Function 0
	PIO10_5	PIO10_5	Function 0
	PIO10_6	PIO10_6	Function 0
	PIO10_7	PIO10_7	Function 0
	PIO10_8	PIO10_8	Function 0
	PIO10_9	PIO10_9	Function 0
	PIO2_0	PIO2_0	Function 0
	PIO2_1	PIO2_1	Function 0
	PIO2_10	PIO2_10	Function 0
	PIO2_11	PIO2_11	Function 0
	PIO2_12	PIO2_12	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO2_13	PIO2_13	Function 0
	PIO2_14	PIO2_14	Function 0
	PIO2_15	PIO2_15	Function 0
	PIO2_2	PIO2_2	Function 0
	PIO2_3	PIO2_3	Function 0
	PIO2_4	PIO2_4	Function 0
	PIO2_5	PIO2_5	Function 0
	PIO2_6	PIO2_6	Function 0
	PIO2_7	PIO2_7	Function 0
	PIO2_8	PIO2_8	Function 0
	PIO2_9	PIO2_9	Function 0
	PIO3_0	PIO3_0	Function 0
	PIO3_1	PIO3_1	Function 0
	PIO3_10	PIO3_10	Function 0
	PIO3_11	PIO3_11	Function 0
	PIO3_2	PIO3_2	Function 0
	PIO3_3	PIO3_3	Function 0
	PIO3_4	PIO3_4	Function 0
	PIO3_5	PIO3_5	Function 0
	PIO3_6	PIO3_6	Function 0
	PIO3_7	PIO3_7	Function 0
	PIO3_8	PIO3_8	Function 0
	PIO3_9	PIO3_9	Function 0
	PIO4_0	PIO4_0	Function 0
	PIO4_1	PIO4_1	Function 0
	PIO4_10	PIO4_10	Function 0
	PIO4_11	PIO4_11	Function 0
	PIO4_12	PIO4_12	Function 0
	PIO4_13	PIO4_13	Function 0
	PIO4_14	PIO4_14	Function 0
	PIO4_15	PIO4_15	Function 0
	PIO4_16	PIO4_16	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO4_17	PIO4_17	Function 0
	PIO4_18	PIO4_18	Function 0
	PIO4_19	PIO4_19	Function 0
	PIO4_2	PIO4_2	Function 0
	PIO4_20	PIO4_20	Function 0
	PIO4_3	PIO4_3	Function 0
	PIO4_4	PIO4_4	Function 0
	PIO4_5	PIO4_5	Function 0
	PIO4_6	PIO4_6	Function 0
	PIO4_7	PIO4_7	Function 0
	PIO4_8	PIO4_8	Function 0
	PIO4_9	PIO4_9	Function 0
	PIO5_0	PIO5_0	Function 0
	PIO5_1	PIO5_1	Function 0
	PIO5_10	PIO5_10	Function 0
	PIO5_11	PIO5_11	Function 0
	PIO5_12	PIO5_12	Function 0
	PIO5_13	PIO5_13	Function 0
	PIO5_14	PIO5_14	Function 0
	PIO5_15	PIO5_15	Function 0
	PIO5_16	PIO5_16	Function 0
	PIO5_17	PIO5_17	Function 0
	PIO5_18	PIO5_18	Function 0
	PIO5_19	PIO5_19	Function 0
	PIO5_2	PIO5_2	Function 0
	PIO5_20	PIO5_20	Function 0
	PIO5_3	PIO5_3	Function 0
	PIO5_4	PIO5_4	Function 0
	PIO5_5	PIO5_5	Function 0
	PIO5_6	PIO5_6	Function 0
	PIO5_7	PIO5_7	Function 0
	PIO5_8	PIO5_8	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO5_9	PIO5_9	Function 0
	PIO6_0	PIO6_0	Function 0
	PIO6_1	PIO6_1	Function 0
	PIO6_10	PIO6_10	Function 0
	PIO6_11	PIO6_11	Function 0
	PIO6_12	PIO6_12	Function 0
	PIO6_2	PIO6_2	Function 0
	PIO6_3	PIO6_3	Function 0
	PIO6_4	PIO6_4	Function 0
	PIO6_5	PIO6_5	Function 0
	PIO6_6	PIO6_6	Function 0
	PIO6_7	PIO6_7	Function 0
	PIO6_8	PIO6_8	Function 0
	PIO6_9	PIO6_9	Function 0
	PIO7_0	PIO7_0	Function 0
	PIO7_1	PIO7_1	Function 0
	PIO7_10	PIO7_10	Function 0
	PIO7_11	PIO7_11	Function 0
	PIO7_12	PIO7_12	Function 0
	PIO7_13	PIO7_13	Function 0
	PIO7_14	PIO7_14	Function 0
	PIO7_15	PIO7_15	Function 0
	PIO7_16	PIO7_16	Function 0
	PIO7_17	PIO7_17	Function 0
	PIO7_18	PIO7_18	Function 0
	PIO7_19	PIO7_19	Function 0
	PIO7_2	PIO7_2	Function 0
	PIO7_20	PIO7_20	Function 0
	PIO7_21	PIO7_21	Function 0
	PIO7_22	PIO7_22	Function 0
	PIO7_23	PIO7_23	Function 0
	PIO7_24	PIO7_24	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO7_25	PIO7_25	Function 0
	PIO7_3	PIO7_3	Function 0
	PIO7_4	PIO7_4	Function 0
	PIO7_5	PIO7_5	Function 0
	PIO7_6	PIO7_6	Function 0
	PIO7_7	PIO7_7	Function 0
	PIO7_8	PIO7_8	Function 0
	PIO7_9	PIO7_9	Function 0
	PIO8_0	PIO8_0	Function 0
	PIO8_1	PIO8_1	Function 0
	PIO8_10	PIO8_10	Function 0
	PIO8_11	PIO8_11	Function 0
	PIO8_12	PIO8_12	Function 0
	PIO8_13	PIO8_13	Function 0
	PIO8_14	PIO8_14	Function 0
	PIO8_15	PIO8_15	Function 0
	PIO8_16	PIO8_16	Function 0
	PIO8_17	PIO8_17	Function 0
	PIO8_18	PIO8_18	Function 0
	PIO8_19	PIO8_19	Function 0
	PIO8_2	PIO8_2	Function 0
	PIO8_20	PIO8_20	Function 0
	PIO8_21	PIO8_21	Function 0
	PIO8_22	PIO8_22	Function 0
	PIO8_23	PIO8_23	Function 0
	PIO8_24	PIO8_24	Function 0
	PIO8_25	PIO8_25	Function 0
	PIO8_26	PIO8_26	Function 0
	PIO8_27	PIO8_27	Function 0
	PIO8_28	PIO8_28	Function 0
	PIO8_3	PIO8_3	Function 0
	PIO8_30	PIO8_30	Function 0

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO8_31	PIO8_31	Function 0
	PIO8_4	PIO8_4	Function 0
	PIO8_5	PIO8_5	Function 0
	PIO8_6	PIO8_6	Function 0
	PIO8_7	PIO8_7	Function 0
	PIO8_8	PIO8_8	Function 0
	PIO8_9	PIO8_9	Function 0
I3C0	I3C0_PUR	PIO1_16	Function 5
	I3C0_PUR	PIO3_6	Function 5
	I3C0_SCL	PIO1_17	Function 5
	I3C0_SDA	PIO1_18	Function 5
I3C1	I3C1_PUR	PIO1_6	Function 5
	I3C1_PUR	PIO3_11	Function 5
	I3C1_SCL	PIO1_8	Function 5
	I3C1_SDA	PIO1_7	Function 5
I3C2	I3C2_PUR	PIO8_16	Function 4
	I3C2_SCL	PIO8_15	Function 4
	I3C2_SDA	PIO8_14	Function 4
I3C3	I3C3_PUR	PIO8_20	Function 4
	I3C3_SCL	PIO8_22	Function 4
	I3C3_SDA	PIO8_21	Function 4
LCDIF	LCD_D0	PIO2_6	Function 6
	LCD_D1	PIO2_7	Function 6
	LCD_D10	PIO4_14	Function 6
	LCD_D11	PIO4_15	Function 6
	LCD_D12	PIO4_17	Function 6
	LCD_D13	PIO4_18	Function 6
	LCD_D14	PIO4_19	Function 6
	LCD_D15	PIO4_20	Function 6
	LCD_D16	PIO4_6	Function 6
	LCD_D17	PIO4_7	Function 6
	LCD_D18	PIO4_8	Function 6

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LCD_D19	PIO4_9	Function 6
	LCD_D2	PIO2_8	Function 6
	LCD_D20	PIO4_1	Function 6
	LCD_D21	PIO4_2	Function 6
	LCD_D22	PIO4_3	Function 6
	LCD_D23	PIO4_4	Function 6
	LCD_D3	PIO2_9	Function 6
	LCD_D4	PIO2_10	Function 6
	LCD_D5	PIO2_11	Function 6
	LCD_D6	PIO2_12	Function 6
	LCD_D7	PIO2_13	Function 6
	LCD_D8	PIO4_12	Function 6
	LCD_D9	PIO4_13	Function 6
	LCD_DOTCLK	PIO2_3	Function 6
	LCD_ENABLE	PIO2_2	Function 6
	LCD_HSYNC	PIO2_4	Function 6
	LCD_VSYNC	PIO2_5	Function 6
	LCD_DBI_CSX_AB	PIO2_0	Function 7
	LCD_DBI-D0	PIO2_6	Function 7
	LCD_DBI-D1	PIO2_7	Function 7
	LCD_DBI-D10	PIO4_14	Function 7
	LCD_DBI-D11	PIO4_15	Function 7
	LCD_DBI-D12	PIO4_17	Function 7
	LCD_DBI-D13	PIO4_18	Function 7
	LCD_DBI-D14	PIO4_19	Function 7
	LCD_DBI-D15	PIO4_20	Function 7
	LCD_DBI-D2	PIO2_8	Function 7
	LCD_DBI-D3	PIO2_9	Function 7
	LCD_DBI-D4	PIO2_10	Function 7
	LCD_DBI-D5	PIO2_11	Function 7
	LCD_DBI-D6	PIO2_12	Function 7
	LCD_DBI-D7	PIO2_13	Function 7

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LCD_DBI-D8	PIO4_12	Function 7
	LCD_DBI-D9	PIO4_13	Function 7
	LCD_DBI-DATA_OEN	PIO2_2	Function 7
	LCD_DBI-DCX-AB	PIO2_1	Function 7
	LCD_DBI-E	PIO2_5	Function 7
	LCD_DBI-RWDX	PIO2_3	Function 7
	LCD_DBI-WRX	PIO2_4	Function 7
LP_FLEXCOMM0	LP_FLEXCOMM0_P0	PIO0_31	Function 1
	LP_FLEXCOMM0_P1	PIO1_0	Function 1
	LP_FLEXCOMM0_P2	PIO3_7	Function 1
	LP_FLEXCOMM0_P3	PIO1_2	Function 1
	LP_FLEXCOMM0_P3	PIO3_8	Function 1
	LP_FLEXCOMM0_P4	PIO1_3	Function 1
	LP_FLEXCOMM0_P5	PIO1_4	Function 1
	LP_FLEXCOMM0_P6	PIO1_5	Function 1
	LP_FLEXCOMM0_P4	PIO3_4	Function 2
	LP_FLEXCOMM0_P5	PIO3_5	Function 2
	LP_FLEXCOMM0_P6	PIO3_6	Function 2
LP_FLEXCOMM0 (BOOT)	LP_FLEXCOMM0_P0 (UART boot)	PIO0_31	Alternate pin function
	LP_FLEXCOMM0_P1 (UART boot)	PIO1_0	Alternate pin function
	LPSPI0_PCS0 (SPI boot)	PIO3_3	Alternate pin function
	LPSPI0_SCK (SPI boot)	PIO3_1	Alternate pin function
	LPSPI0_SIN (SPI boot)	PIO3_2	Alternate pin function
	LPSPI0_SOUT (SPI boot)	PIO3_0	Alternate pin function
LP_FLEXCOMM1	LP_FLEXCOMM1_P0	PIO1_7	Function 1
	LP_FLEXCOMM1_P1	PIO1_8	Function 1
	LP_FLEXCOMM1_P2	PIO1_9	Function 1
	LP_FLEXCOMM1_P3	PIO1_10	Function 1
	LP_FLEXCOMM1_P4	PIO1_6	Function 1
	LP_FLEXCOMM1_P0	PIO7_10	Function 2
	LP_FLEXCOMM1_P1	PIO7_11	Function 2
	LP_FLEXCOMM1_P2	PIO7_12	Function 2

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM1_P3	PIO7_13	Function 2
	LP_FLEXCOMM1_P4	PIO7_19	Function 2
	LP_FLEXCOMM1_P5	PIO7_20	Function 2
	LP_FLEXCOMM1_P6	PIO7_21	Function 2
	LP_FLEXCOMM1_P5	PIO0_9	Function 5
LP_FLEXCOMM2	LP_FLEXCOMM2_P0	PIO1_11	Function 1
	LP_FLEXCOMM2_P1	PIO1_12	Function 1
	LP_FLEXCOMM2_P2	PIO1_14	Function 1
	LP_FLEXCOMM2_P3	PIO1_15	Function 1
	LP_FLEXCOMM2_P4	PIO1_13	Function 1
	LP_FLEXCOMM2_P0	PIO7_17	Function 2
	LP_FLEXCOMM2_P1	PIO7_18	Function 2
	LP_FLEXCOMM2_P2	PIO7_22	Function 2
	LP_FLEXCOMM2_P3	PIO7_23	Function 2
	LP_FLEXCOMM2_P4	PIO7_20	Function 3
	LP_FLEXCOMM2_P5	PIO7_21	Function 3
	LP_FLEXCOMM2_P6	PIO7_25	Function 3
	LP_FLEXCOMM2_P5	PIO0_15	Function 6
	LP_FLEXCOMM2_P6	PIO0_16	Function 6
LP_FLEXCOMM3	LP_FLEXCOMM3_P0	PIO1_18	Function 1
	LP_FLEXCOMM3_P1	PIO1_17	Function 1
	LP_FLEXCOMM3_P2	PIO1_16	Function 1
	LP_FLEXCOMM3_P3	PIO1_19	Function 1
	LP_FLEXCOMM3_P4	PIO7_0	Function 2
	LP_FLEXCOMM3_P5	PIO7_1	Function 2
	LP_FLEXCOMM3_P6	PIO7_9	Function 2
	LP_FLEXCOMM3_P0	PIO8_21	Function 3
	LP_FLEXCOMM3_P1	PIO8_22	Function 3
	LP_FLEXCOMM3_P2	PIO3_4	Function 3
	LP_FLEXCOMM3_P3	PIO3_5	Function 3
	LP_FLEXCOMM3_P4	PIO3_9	Function 3
	LP_FLEXCOMM3_P5	PIO3_10	Function 3

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
LP_FLEXCOMM3 (BOOT)	LP_FLEXCOMM3_P0 (I2C boot)	PIO1_18	Alternate pin function
	LP_FLEXCOMM3_P1 (I2C boot)	PIO1_17	Alternate pin function
LP_FLEXCOMM4	LP_FLEXCOMM4_P0	PIO0_3	Function 1
	LP_FLEXCOMM4_P1	PIO0_4	Function 1
	LP_FLEXCOMM4_P2	PIO0_6	Function 1
	LP_FLEXCOMM4_P3	PIO0_7	Function 1
	LP_FLEXCOMM4_P4	PIO0_5	Function 1
	LP_FLEXCOMM4_P6	PIO0_9	Function 1
LP_FLEXCOMM5	LP_FLEXCOMM5_P0	PIO0_10	Function 1
	LP_FLEXCOMM5_P1	PIO0_11	Function 1
	LP_FLEXCOMM5_P2	PIO0_12	Function 1
	LP_FLEXCOMM5_P3	PIO0_13	Function 1
	LP_FLEXCOMM5_P4	PIO0_14	Function 1
	LP_FLEXCOMM5_P5	PIO0_15	Function 1
	LP_FLEXCOMM5_P6	PIO0_16	Function 1
	LP_FLEXCOMM5_P5	PIO3_7	Function 3
	LP_FLEXCOMM5_P6	PIO3_8	Function 3
LP_FLEXCOMM6	LP_FLEXCOMM6_P0	PIO0_21	Function 1
	LP_FLEXCOMM6_P1	PIO0_22	Function 1
	LP_FLEXCOMM6_P2	PIO0_19	Function 1
	LP_FLEXCOMM6_P2	PIO3_9	Function 1
	LP_FLEXCOMM6_P3	PIO0_20	Function 1
	LP_FLEXCOMM6_P3	PIO3_10	Function 1
	LP_FLEXCOMM6_P4	PIO3_11	Function 1
	LP_FLEXCOMM6_P5	PIO0_17	Function 1
	LP_FLEXCOMM6_P6	PIO0_18	Function 1
LP_FLEXCOMM7	LP_FLEXCOMM7_P0	PIO7_14	Function 2
	LP_FLEXCOMM7_P1	PIO7_15	Function 2
	LP_FLEXCOMM7_P2	PIO6_0	Function 2
	LP_FLEXCOMM7_P2	PIO7_16	Function 2
	LP_FLEXCOMM7_P3	PIO7_8	Function 3
	LP_FLEXCOMM7_P3	PIO7_19	Function 3

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM7_P4	PIO7_9	Function 3
	LP_FLEXCOMM7_P5	PIO7_1	Function 3
	LP_FLEXCOMM7_P6	PIO7_0	Function 3
LP_FLEXCOMM8	LP_FLEXCOMM8_P2	PIO0_0	Function 1
	LP_FLEXCOMM8_P3	PIO0_1	Function 1
	LP_FLEXCOMM8_P4	PIO0_2	Function 1
	LP_FLEXCOMM8_P0	PIO4_0	Function 2
	LP_FLEXCOMM8_P1	PIO4_1	Function 2
	LP_FLEXCOMM8_P2	PIO4_2	Function 2
	LP_FLEXCOMM8_P3	PIO4_3	Function 2
	LP_FLEXCOMM8_P4	PIO4_4	Function 2
	LP_FLEXCOMM8_P5	PIO4_5	Function 2
	LP_FLEXCOMM8_P6	PIO4_6	Function 2
	LP_FLEXCOMM8_P0	PIO0_6	Function 5
	LP_FLEXCOMM8_P1	PIO0_7	Function 5
LP_FLEXCOMM9	LP_FLEXCOMM9_P0	PIO4_7	Function 2
	LP_FLEXCOMM9_P1	PIO4_8	Function 2
	LP_FLEXCOMM9_P2	PIO4_9	Function 2
	LP_FLEXCOMM9_P3	PIO4_10	Function 2
	LP_FLEXCOMM9_P4	PIO4_11	Function 2
	LP_FLEXCOMM9_P5	PIO4_12	Function 2
	LP_FLEXCOMM9_P6	PIO4_13	Function 2
	LP_FLEXCOMM9_P2	PIO3_6	Function 3
	LP_FLEXCOMM9_P3	PIO3_11	Function 3
	LP_FLEXCOMM9_P3	PIO1_4	Function 5
	LP_FLEXCOMM9_P4	PIO1_5	Function 5
	LP_FLEXCOMM9_P5	PIO1_2	Function 5
	LP_FLEXCOMM9_P6	PIO1_3	Function 5
	LP_FLEXCOMM9_P0	PIO0_2	Function 7
	LP_FLEXCOMM9_P1	PIO0_5	Function 7
LP_FLEXCOMM10	LP_FLEXCOMM10_P0	PIO4_14	Function 2
	LP_FLEXCOMM10_P0	PIO6_1	Function 2

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM10_P1	PIO4_15	Function 2
	LP_FLEXCOMM10_P1	PIO6_2	Function 2
	LP_FLEXCOMM10_P2	PIO4_16	Function 2
	LP_FLEXCOMM10_P2	PIO6_3	Function 2
	LP_FLEXCOMM10_P3	PIO4_17	Function 2
	LP_FLEXCOMM10_P3	PIO6_4	Function 2
	LP_FLEXCOMM10_P4	PIO4_18	Function 2
	LP_FLEXCOMM10_P4	PIO6_5	Function 2
	LP_FLEXCOMM10_P5	PIO4_19	Function 2
	LP_FLEXCOMM10_P5	PIO6_6	Function 2
	LP_FLEXCOMM10_P6	PIO4_20	Function 2
	LP_FLEXCOMM10_P6	PIO6_7	Function 2
LP_FLEXCOMM11	LP_FLEXCOMM11_P0	PIO5_14	Function 2
	LP_FLEXCOMM11_P1	PIO5_15	Function 2
	LP_FLEXCOMM11_P2	PIO5_16	Function 2
	LP_FLEXCOMM11_P3	PIO5_17	Function 2
	LP_FLEXCOMM11_P4	PIO5_18	Function 2
	LP_FLEXCOMM11_P5	PIO5_19	Function 2
	LP_FLEXCOMM11_P6	PIO5_20	Function 2
	LP_FLEXCOMM11_P0	PIO6_10	Function 3
	LP_FLEXCOMM11_P1	PIO6_11	Function 3
	LP_FLEXCOMM11_P2	PIO6_8	Function 3
	LP_FLEXCOMM11_P3	PIO6_9	Function 3
LP_FLEXCOMM12	LP_FLEXCOMM12_P0	PIO5_7	Function 2
	LP_FLEXCOMM12_P0	PIO7_2	Function 2
	LP_FLEXCOMM12_P1	PIO5_8	Function 2
	LP_FLEXCOMM12_P1	PIO7_3	Function 2
	LP_FLEXCOMM12_P2	PIO5_9	Function 2
	LP_FLEXCOMM12_P2	PIO7_4	Function 2
	LP_FLEXCOMM12_P3	PIO5_6	Function 2
	LP_FLEXCOMM12_P3	PIO7_5	Function 2
	LP_FLEXCOMM12_P4	PIO5_11	Function 2

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM12_P4	PIO7_6	Function 2
	LP_FLEXCOMM12_P5	PIO5_12	Function 2
	LP_FLEXCOMM12_P5	PIO7_7	Function 2
	LP_FLEXCOMM12_P6	PIO5_13	Function 2
	LP_FLEXCOMM12_P6	PIO7_8	Function 2
LP_FLEXCOMM13	LP_FLEXCOMM13_P0	PIO5_3	Function 2
	LP_FLEXCOMM13_P0	PIO6_8	Function 2
	LP_FLEXCOMM13_P1	PIO5_4	Function 2
	LP_FLEXCOMM13_P1	PIO6_9	Function 2
	LP_FLEXCOMM13_P2	PIO5_5	Function 2
	LP_FLEXCOMM13_P2	PIO6_10	Function 2
	LP_FLEXCOMM13_P3	PIO5_10	Function 2
	LP_FLEXCOMM13_P3	PIO6_11	Function 2
	LP_FLEXCOMM13_P4	PIO5_0	Function 2
	LP_FLEXCOMM13_P4	PIO7_24	Function 2
	LP_FLEXCOMM13_P5	PIO5_1	Function 2
	LP_FLEXCOMM13_P5	PIO7_25	Function 2
	LP_FLEXCOMM13_P6	PIO5_2	Function 2
LP_FLEXCOMM14 (LPSPI14)	LPSPI14_PCS0	PIO3_3	Function 1
	LPSPI14_PCS1	PIO3_6	Function 1
	LPSPI14_PCS2	PIO3_5	Function 1
	LPSPI14_PCS3	PIO3_4	Function 1
	LPSPI14_SCK	PIO3_1	Function 1
	LPSPI14_SIN	PIO3_2	Function 1
	LPSPI14_SOUT	PIO3_0	Function 1
LP_FLEXCOMM15 (LPI2C15)	PMIC_I2C_SCL	PMIC_I2C_SCL	Alternate pin function
	PMIC_I2C_SDA	PMIC_I2C_SDA	Alternate pin function
	PMIC_I2C_SCL	PMIC_I2C_SCL	Function 0
	PMIC_I2C_SDA	PMIC_I2C_SDA	Function 0
LP_FLEXCOMM16 (LPSPI16)	LPSPI16_PCS0	PIO2_5	Function 1
	LPSPI16_PCS1	PIO2_8	Function 1
	LPSPI16_PCS2	PIO2_7	Function 1

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LPSPI16_PCS3	PIO2_6	Function 1
	LPSPI16_SCK	PIO2_3	Function 1
	LPSPI16_SIN	PIO2_4	Function 1
	LPSPI16_SOUT	PIO2_2	Function 1
LP_FLEXCOMM17	LP_FLEXCOMM17_P0	PIO8_5	Function 1
	LP_FLEXCOMM17_P1	PIO8_6	Function 1
	LP_FLEXCOMM17_P2	PIO8_2	Function 1
	LP_FLEXCOMM17_P3	PIO8_3	Function 1
	LP_FLEXCOMM17_P4	PIO8_4	Function 1
	LP_FLEXCOMM17_P5	PIO8_0	Function 1
	LP_FLEXCOMM17_P6	PIO8_1	Function 1
	LP_FLEXCOMM17_P2	PIO8_12	Function 4
	LP_FLEXCOMM17_P3	PIO8_13	Function 4
	LP_FLEXCOMM17_P4	PIO8_11	Function 4
LP_FLEXCOMM18	LP_FLEXCOMM18_P0	PIO8_7	Function 1
	LP_FLEXCOMM18_P1	PIO8_8	Function 1
	LP_FLEXCOMM18_P2	PIO8_9	Function 1
	LP_FLEXCOMM18_P3	PIO8_10	Function 1
	LP_FLEXCOMM18_P4	PIO8_11	Function 1
	LP_FLEXCOMM18_P5	PIO8_12	Function 1
	LP_FLEXCOMM18_P6	PIO8_13	Function 1
LP_FLEXCOMM19	LP_FLEXCOMM19_P0	PIO8_14	Function 1
	LP_FLEXCOMM19_P1	PIO8_15	Function 1
	LP_FLEXCOMM19_P2	PIO8_16	Function 1
	LP_FLEXCOMM19_P3	PIO8_17	Function 1
	LP_FLEXCOMM19_P4	PIO8_18	Function 1
	LP_FLEXCOMM19_P5	PIO8_19	Function 1
	LP_FLEXCOMM19_P6	PIO8_20	Function 1
	LP_FLEXCOMM19_P0	PIO8_23	Function 3
	LP_FLEXCOMM19_P1	PIO8_24	Function 3
	LP_FLEXCOMM19_P2	PIO8_25	Function 3
	LP_FLEXCOMM19_P3	PIO8_26	Function 3

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM19_P4	PIO8_27	Function 3
	LP_FLEXCOMM19_P5	PIO8_28	Function 3
LP_FLEXCOMM20	LP_FLEXCOMM20_P0	PIO8_30	Function 1
	LP_FLEXCOMM20_P1	PIO8_31	Function 1
	LP_FLEXCOMM20_P3	PIO8_21	Function 1
	LP_FLEXCOMM20_P4	PIO8_22	Function 1
MICFIL	PDM_CLK	PIO10_0	Function 5
	PDM_CLK	PIO10_8	Function 5
	PDM_CLK	PIO10_13	Function 5
	PDM_DATA01	PIO10_1	Function 5
	PDM_DATA01	PIO10_9	Function 5
	PDM_DATA01	PIO10_14	Function 5
	PDM_DATA23	PIO10_2	Function 5
	PDM_DATA23	PIO10_15	Function 5
	PDM_DATA45	PIO10_3	Function 5
	PDM_DATA45	PIO10_16	Function 5
	PDM_DATA67	PIO10_4	Function 5
	PDM_DATA67	PIO10_12	Function 5
	PDM_DATA67	PIO10_17	Function 5
MIPI_DSI_HOST	MIPI_DSI_CLKN	MIPI_DSI_CLKN	Alternate pin function
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	Alternate pin function
	MIPI_DSI_D0N	MIPI_DSI_D0N	Alternate pin function
	MIPI_DSI_D0P	MIPI_DSI_D0P	Alternate pin function
	MIPI_DSI_D1N	MIPI_DSI_D1N	Alternate pin function
	MIPI_DSI_D1P	MIPI_DSI_D1P	Alternate pin function
	MIPI_DSI_CLKN	MIPI_DSI_CLKN	Function 0
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	Function 0
	MIPI_DSI_D0N	MIPI_DSI_D0N	Function 0
	MIPI_DSI_D0P	MIPI_DSI_D0P	Function 0
	MIPI_DSI_D1N	MIPI_DSI_D1N	Function 0
	MIPI_DSI_D1P	MIPI_DSI_D1P	Function 0
OSC32KNP	32KHZ_CLKOUT	PIO1_3	Function 6

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	32KHZ_CLKOUT	PIO1_6	Function 6
	32KHZ_CLKOUT	PIO1_13	Function 6
	EXTAL	XTALIN32KHZ	Function 0
	XTAL	XTALOUT32KHZ	Function 0
PMC	PMIC_IRQN	PMIC_IRQN	Alternate pin function
	PMIC_MODE0	PMIC_MODE0	Alternate pin function
	PMIC_MODE1	PMIC_MODE1	Alternate pin function
	RESETN	RESETN	Alternate pin function
	PMIC_IRQN	PMIC_IRQN	Function 0
	PMIC_MODE0	PMIC_MODE0	Function 0
	PMIC_MODE1	PMIC_MODE1	Function 0
	RESETN	RESETN	Function 0
RTC	RTCXIN	XTALINSYS	Alternate pin function
	RTCXOUT	XTALOUTSYS	Alternate pin function
SAI0[1]	SAI0_RX_BCLK	PIO0_0	Function 5
	SAI0_RX_DATA	PIO0_2	Function 5
	SAI0_RX_SYNC	PIO0_1	Function 5
	SAI0_TX_BCLK	PIO0_3	Function 5
	SAI0_TX_DATA	PIO0_4	Function 5
	SAI0_TX_SYNC	PIO0_5	Function 5
SAI1[1]	SAI1_RX_BCLK	PIO0_17	Function 5
	SAI1_RX_DATA	PIO0_18	Function 5
	SAI1_RX_SYNC	PIO0_19	Function 5
	SAI1_TX_BCLK	PIO0_20	Function 5
	SAI1_TX_DATA	PIO0_21	Function 5
	SAI1_TX_SYNC	PIO0_22	Function 5
	SAI1_RX_DATA	PIO0_1	Function 6
	SAI1_TX_BCLK	PIO0_0	Function 6
SAI2[1]	SAI2_RX_BCLK	PIO0_15	Function 5
	SAI2_RX_DATA	PIO0_14	Function 5
	SAI2_RX_SYNC	PIO0_16	Function 5
	SAI2_TX_BCLK	PIO0_11	Function 5

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	SAI2_TX_DATA	PIO0_12	Function 5
	SAI2_TX_SYNC	PIO0_13	Function 5
	SAI2_RX_DATA	PIO0_10	Function 6
SAI3[1]	SAI3_RX_BCLK	PIO8_23	Function 1
	SAI3_RX_DATA	PIO8_24	Function 1
	SAI3_RX_SYNC	PIO8_25	Function 1
	SAI3_TX_BCLK	PIO8_26	Function 1
	SAI3_TX_DATA	PIO8_27	Function 1
	SAI3_TX_SYNC	PIO8_28	Function 1
	SAI3_RX_SYNC	PIO8_17	Function 3
	SAI3_TX_BCLK	PIO8_15	Function 3
	SAI3_TX_DATA	PIO8_16	Function 3
	SAI3_TX_SYNC	PIO8_14	Function 3
SCT0	SCT0_GPIN0	PIO0_0	Function 2
	SCT0_GPIN0	PIO0_14	Function 2
	SCT0_GPIN0	PIO0_17	Function 2
	SCT0_GPIN0	PIO1_2	Function 2
	SCT0_GPIN1	PIO0_1	Function 2
	SCT0_GPIN1	PIO0_9	Function 2
	SCT0_GPIN1	PIO0_18	Function 2
	SCT0_GPIN1	PIO1_3	Function 2
	SCT0_GPIN2	PIO0_2	Function 2
	SCT0_GPIN2	PIO0_15	Function 2
	SCT0_GPIN2	PIO0_19	Function 2
	SCT0_GPIN2	PIO1_4	Function 2
	SCT0_GPIN3	PIO0_3	Function 2
	SCT0_GPIN3	PIO0_16	Function 2
	SCT0_GPIN3	PIO0_20	Function 2
	SCT0_GPIN3	PIO1_5	Function 2
	SCT0_GPIN3	PIO1_12	Function 2
	SCT0_GPIN4	PIO0_4	Function 2
	SCT0_GPIN4	PIO0_10	Function 2

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	SCT0_GPIN4	PIO0_22	Function 2
	SCT0_GPIN4	PIO1_6	Function 2
	SCT0_GPIN5	PIO0_5	Function 2
	SCT0_GPIN5	PIO0_11	Function 2
	SCT0_GPIN5	PIO1_7	Function 2
	SCT0_GPIN6	PIO0_6	Function 2
	SCT0_GPIN6	PIO0_12	Function 2
	SCT0_GPIN6	PIO0_21	Function 2
	SCT0_GPIN6	PIO1_8	Function 2
	SCT0_GPIN7	PIO0_7	Function 2
	SCT0_GPIN7	PIO0_13	Function 2
	SCT0_GPIN7	PIO1_9	Function 2
	SCT0_OUT0	PIO0_0	Function 3
	SCT0_OUT0	PIO0_17	Function 3
	SCT0_OUT0	PIO1_6	Function 3
	SCT0_OUT1	PIO0_1	Function 3
	SCT0_OUT1	PIO0_9	Function 3
	SCT0_OUT1	PIO0_18	Function 3
	SCT0_OUT1	PIO1_7	Function 3
	SCT0_OUT2	PIO0_2	Function 3
	SCT0_OUT2	PIO0_15	Function 3
	SCT0_OUT2	PIO0_19	Function 3
	SCT0_OUT2	PIO1_8	Function 3
	SCT0_OUT3	PIO0_3	Function 3
	SCT0_OUT3	PIO0_16	Function 3
	SCT0_OUT3	PIO0_20	Function 3
	SCT0_OUT3	PIO1_9	Function 3
	SCT0_OUT4	PIO0_4	Function 3
	SCT0_OUT4	PIO0_10	Function 3
	SCT0_OUT4	PIO0_22	Function 3
	SCT0_OUT5	PIO0_5	Function 3
	SCT0_OUT5	PIO0_11	Function 3

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	SCT0_OUT6	PIO0_6	Function 3
	SCT0_OUT6	PIO0_12	Function 3
	SCT0_OUT6	PIO0_21	Function 3
	SCT0_OUT6	PIO1_2	Function 3
	SCT0_OUT7	PIO0_7	Function 3
	SCT0_OUT7	PIO0_13	Function 3
	SCT0_OUT7	PIO1_3	Function 3
	SCT0_OUT8	PIO0_14	Function 3
	SCT0_OUT8	PIO0_31	Function 3
	SCT0_OUT8	PIO1_4	Function 3
	SCT0_OUT9	PIO1_0	Function 3
	SCT0_OUT9	PIO1_5	Function 3
	SCT0_OUT9	PIO1_12	Function 3
SDADC	SDADC_INN0	PIO10_7	Alternate pin function
	SDADC_INP0	PIO10_6	Alternate pin function
USB0	USB0_DM	USB0_DM	Function 0
	USB0_DP	USB0_DP	Function 0
	USB0_VBUS	USB0_VBUS	Function 0
	USB0_OVERCURRENTN	PIO1_17	Function 2
	USB0_OVERCURRENTN	PIO2_0	Function 2
	USB0_OVERCURRENTN	PIO3_10	Function 2
	USB0_PORTPWRN	PIO1_18	Function 2
	USB0_PORTPWRN	PIO2_1	Function 2
	USB0_PORTPWRN	PIO3_11	Function 2
USB1	USB1_DM	USB0_DM	Alternate pin function
	USB1_DP	USB0_DP	Alternate pin function
	USB1_VBUS	USB0_VBUS	Alternate pin function
	EUSB_DM	EUSB_DM	Alternate pin function
	EUSB_DP	EUSB_DP	Alternate pin function
	EUSB_DM	EUSB_DM	Function 0
	EUSB_DP	EUSB_DP	Function 0
uSDHC0 (BOOT)	SDHC0_CLK (SDHC0 boot)	PIO7_0	Alternate pin function

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	SDHC0_CMD (SDHC0 boot)	PIO7_1	Alternate pin function
	SDHC0_DATA0 (SDHC0 boot)	PIO7_6	Alternate pin function
	SDHC0_DATA1 (SDHC0 boot)	PIO7_7	Alternate pin function
	SDHC0_DATA2 (SDHC0 boot)	PIO7_8	Alternate pin function
	SDHC0_DATA3 (SDHC0 boot)	PIO7_9	Alternate pin function
uSDHC0	SDHC0_CARD_DET_N	PIO7_11	Function 1
	SDHC0_CLK	PIO7_0	Function 1
	SDHC0_CMD	PIO7_1	Function 1
	SDHC0_DATA0	PIO7_6	Function 1
	SDHC0_DATA1	PIO7_7	Function 1
	SDHC0_DATA2	PIO7_8	Function 1
	SDHC0_DATA3	PIO7_9	Function 1
	SDHC0_DATA4	PIO7_2	Function 1
	SDHC0_DATA5	PIO7_3	Function 1
	SDHC0_DATA6	PIO7_4	Function 1
	SDHC0_DATA7	PIO7_5	Function 1
	SDHC0_RESET_N	PIO7_12	Function 1
	SDHC0_WR_PRT	PIO7_10	Function 1
	SDHC0_DS	PIO7_10	Function 5
uSDHC1 (BOOT)	SDHC1_CLK (SDHC1 boot)	PIO7_17	Alternate pin function
	SDHC1_CMD (SDHC1 boot)	PIO7_18	Alternate pin function
	SDHC1_DATA0 (SDHC1 boot)	PIO7_19	Alternate pin function
	SDHC1_DATA1 (SDHC1 boot)	PIO7_20	Alternate pin function
	SDHC1_DATA2 (SDHC1 boot)	PIO7_21	Alternate pin function
	SDHC1_DATA3 (SDHC1 boot)	PIO7_22	Alternate pin function
uSDHC1	SDHC1_CARD_DET_N	PIO7_24	Function 1
	SDHC1_CLK	PIO7_17	Function 1
	SDHC1_CMD	PIO7_18	Function 1
	SDHC1_DATA0	PIO7_19	Function 1
	SDHC1_DATA1	PIO7_20	Function 1
	SDHC1_DATA2	PIO7_21	Function 1
	SDHC1_DATA3	PIO7_22	Function 1

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	SDHC1_DATA4	PIO7_13	Function 1
	SDHC1_DATA5	PIO7_14	Function 1
	SDHC1_DATA6	PIO7_15	Function 1
	SDHC1_DATA7	PIO7_16	Function 1
	SDHC1_RESET_N	PIO7_25	Function 1
	SDHC1_WR_PRT	PIO7_23	Function 1
	SDHC1_DS	PIO7_23	Function 5
UTICK0	UTICK0_CAP1	PIO1_13	Function 2
	UTICK0_CAP2	PIO0_31	Function 2
	UTICK0_CAP3	PIO1_10	Function 2
UTICK1	UTICK1_CAP0	PIO8_3	Function 3
	UTICK1_CAP1	PIO8_11	Function 3
	UTICK1_CAP2	PIO8_20	Function 3
	UTICK1_CAP3	PIO8_30	Function 3
XSPI0 (BOOT)	XSPI0_DATA0 (XSPI0 - NOR Flash boot)	PIO6_3	Alternate pin function
	XSPI0_DATA1 (XSPI0 - NOR Flash boot)	PIO6_4	Alternate pin function
	XSPI0_DATA2 (XSPI0 - NOR Flash boot)	PIO6_5	Alternate pin function
	XSPI0_DATA3 (XSPI0 - NOR Flash boot)	PIO6_6	Alternate pin function
	XSPI0_DATA4 (XSPI0 - NOR Flash boot)	PIO6_8	Alternate pin function
	XSPI0_DATA5 (XSPI0 - NOR Flash boot)	PIO6_9	Alternate pin function
	XSPI0_DATA6 (XSPI0 - NOR Flash boot)	PIO6_10	Alternate pin function
	XSPI0_DATA7 (XSPI0 - NOR Flash boot)	PIO6_11	Alternate pin function
	XSPI0_DQS0 (XSPI0 - NOR Flash boot)	PIO6_7	Alternate pin function
	XSPI0_SCLK0 (XSPI0 - NOR Flash boot)	PIO6_1	Alternate pin function
	XSPI0_SCLK0_N (XSPI0 - NOR Flash boot)	PIO6_0	Alternate pin function

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	XSPI0_SS0_N (XSPI0 - NOR Flash boot)	PIO6_2	Alternate pin function
XSPI0	XSPI0_DATA0	PIO6_3	Function 1
	XSPI0_DATA1	PIO6_4	Function 1
	XSPI0_DATA2	PIO6_5	Function 1
	XSPI0_DATA3	PIO6_6	Function 1
	XSPI0_DATA4	PIO6_8	Function 1
	XSPI0_DATA5	PIO6_9	Function 1
	XSPI0_DATA6	PIO6_10	Function 1
	XSPI0_DATA7	PIO6_11	Function 1
	XSPI0_DQS0	PIO6_7	Function 1
	XSPI0_SCLK0	PIO6_1	Function 1
	XSPI0_SCLK0_N	PIO6_0	Function 1
	XSPI0_SS0_N	PIO6_2	Function 1
	XSPI0_SS1_N	PIO6_12	Function 1
XSPI1 (BOOT)	XSPI1_DATA0 (XSPI1 - PSRAM boot)	PIO5_1	Alternate pin function
	XSPI1_DATA1 (XSPI1 - PSRAM boot)	PIO5_2	Alternate pin function
	XSPI1_DATA10 (XSPI1 - PSRAM boot)	PIO5_14	Alternate pin function
	XSPI1_DATA11 (XSPI1 - PSRAM boot)	PIO5_15	Alternate pin function
	XSPI1_DATA12 (XSPI1 - PSRAM boot)	PIO5_17	Alternate pin function
	XSPI1_DATA13 (XSPI1 - PSRAM boot)	PIO5_18	Alternate pin function
	XSPI1_DATA14 (XSPI1 - PSRAM boot)	PIO5_19	Alternate pin function
	XSPI1_DATA15 (XSPI1 - PSRAM boot)	PIO5_20	Alternate pin function
	XSPI1_DATA2 (XSPI1 - PSRAM boot)	PIO5_3	Alternate pin function
	XSPI1_DATA3 (XSPI1 - PSRAM boot)	PIO5_4	Alternate pin function
	XSPI1_DATA4 (XSPI1 - PSRAM boot)	PIO5_6	Alternate pin function
	XSPI1_DATA5 (XSPI1 - PSRAM boot)	PIO5_7	Alternate pin function
	XSPI1_DATA6 (XSPI1 - PSRAM boot)	PIO5_8	Alternate pin function
	XSPI1_DATA7 (XSPI1 - PSRAM boot)	PIO5_9	Alternate pin function
	XSPI1_DATA8 (XSPI1 - PSRAM boot)	PIO5_12	Alternate pin function
	XSPI1_DATA9 (XSPI1 - PSRAM boot)	PIO5_13	Alternate pin function
	XSPI1_DQS0 (XSPI1 - PSRAM boot)	PIO5_5	Alternate pin function
	XSPI1_DQS1 (XSPI1 - PSRAM boot)	PIO5_16	Alternate pin function

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	XSPI1_SCLK0 (XSPI1 - PSRAM boot)	PIO5_10	Alternate pin function
	XSPI1_SCLK0_N (XSPI1 - PSRAM boot)	PIO5_11	Alternate pin function
	XSPI1_SS0_N (XSPI1 - PSRAM boot)	PIO5_0	Alternate pin function
XSPI1	XSPI1_DATA0	PIO5_1	Function 1
	XSPI1_DATA1	PIO5_2	Function 1
	XSPI1_DATA10	PIO5_14	Function 1
	XSPI1_DATA11	PIO5_15	Function 1
	XSPI1_DATA12	PIO5_17	Function 1
	XSPI1_DATA13	PIO5_18	Function 1
	XSPI1_DATA14	PIO5_19	Function 1
	XSPI1_DATA15	PIO5_20	Function 1
	XSPI1_DATA2	PIO5_3	Function 1
	XSPI1_DATA3	PIO5_4	Function 1
	XSPI1_DATA4	PIO5_6	Function 1
	XSPI1_DATA5	PIO5_7	Function 1
	XSPI1_DATA6	PIO5_8	Function 1
	XSPI1_DATA7	PIO5_9	Function 1
	XSPI1_DATA8	PIO5_12	Function 1
	XSPI1_DATA9	PIO5_13	Function 1
	XSPI1_DQS0	PIO5_5	Function 1
	XSPI1_DQS1	PIO5_16	Function 1
	XSPI1_SCLK0	PIO5_10	Function 1
	XSPI1_SCLK0_N	PIO5_11	Function 1
	XSPI1_SS0_N	PIO5_0	Function 1
	XSPI1_SS1_N	PIO5_11	Function 3
XSPI2	XSPI2_DATA0	PIO4_1	Function 1
	XSPI2_DATA1	PIO4_2	Function 1
	XSPI2_DATA10	PIO4_14	Function 1
	XSPI2_DATA11	PIO4_15	Function 1
	XSPI2_DATA12	PIO4_17	Function 1
	XSPI2_DATA13	PIO4_18	Function 1
	XSPI2_DATA14	PIO4_19	Function 1

Table continues on the next page...

Table 17. Muxing options (324FOWLP) (continued)

Module	External signal	Pad	FSEL assignment
	XSPI2_DATA15	PIO4_20	Function 1
	XSPI2_DATA2	PIO4_3	Function 1
	XSPI2_DATA3	PIO4_4	Function 1
	XSPI2_DATA4	PIO4_6	Function 1
	XSPI2_DATA5	PIO4_7	Function 1
	XSPI2_DATA6	PIO4_8	Function 1
	XSPI2_DATA7	PIO4_9	Function 1
	XSPI2_DATA8	PIO4_12	Function 1
	XSPI2_DATA9	PIO4_13	Function 1
	XSPI2_DQS0	PIO4_5	Function 1
	XSPI2_DQS1	PIO4_16	Function 1
	XSPI2_SCLK0	PIO4_10	Function 1
	XSPI2_SCLK0_N	PIO4_11	Function 1
	XSPI2_SS1_N	PIO4_11	Function 3
	XSPI2_SS0_N	PIO4_0	Function 1

1. The MCLK indirectly connects SAI via CLKCTL, see CLKCTL for more details.

6.2.2 Muxing options (256WLCSP)

The pinmux selection is done through the FSEL field of the IOPCTL instance associated to the corresponding PIO pad. Alternate pin functions are selected through the AMENA field for analog inputs or through the corresponding boot mode selection detailed in the Boot chapter.

Table 18. Muxing Options (256WLCSP)

Module	External signal	Pad	FSEL assignment
ACMP0	CMP0_INA	PIO10_8	Alternate pin function
	CMP0_INB	PIO10_9	Alternate pin function
	CMP0_OUT	PIO10_5	Function 3
	CMP0_OUT	PIO8_6	Function 4
	CMP0_OUT	PIO8_19	Function 4
ADC0	ADC0_A0	PIO10_0	Alternate pin function
	ADC0_A1	PIO10_2	Alternate pin function
	ADC0_A2	PIO10_4	Alternate pin function
	ADC0_A4, SDADC_INP1	PIO10_12	Alternate pin function
	ADC0_B0	PIO10_1	Alternate pin function

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	ADC0_B1	PIO10_3	Alternate pin function
	ADC0_B2	PIO10_5	Alternate pin function
	ADC0_B4, SDADC_INN1	PIO10_13	Alternate pin function
BOOT	ISP boot select 0 00 - SDHC0 (eMMC or SD set in OTP) 01 - XSPI0 10 – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) 11 - XSPI1	PIO1_7	Alternate pin function
	ISP boot select 1 00 - SDHC0 (eMMC or SD set in OTP) 01 - XSPI0 10 – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) 11 - XSPI1	PIO1_8	Alternate pin function
CLKCTL	CLKIN	PIO1_9	Function 5
	CLKIN	PIO1_15	Function 5
	CLKOUT_VDD1	PIO1_9	Function 6
	CLKOUT_VDD1	PIO1_10	Function 6
	CLKOUT_VDD2	PIO0_14	Function 6
	CLKOUT_VDD2	PIO0_22	Function 6
	LOW_FREQ_CLKOUT	PIO2_14	Function 6
	LOW_FREQ_CLKOUT_N	PIO2_15	Function 6
	MCLK	PIO8_4	Function 3
	MCLK	PIO0_5	Function 6
	MCLK	PIO0_21	Function 6
	XTALIN	XTALINSYS	Function 0
	XTALOUT	XTALOUTSYS	Function 0
CTIMER_C	CTIMER_C_INP0	PIO0_10	Function 5
	CTIMER_C_INP10	PIO0_6	Function 4
	CTIMER_C_INP11	PIO0_7	Function 4
	CTIMER_C_INP12	PIO1_6	Function 4
	CTIMER_C_INP13	PIO1_7	Function 4

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	CTIMER_C_INP14	PIO1_8	Function 4
	CTIMER_C_INP15	PIO1_9	Function 4
	CTIMER_C_INP2	PIO0_14	Function 4
	CTIMER_C_INP4	PIO0_21	Function 4
	CTIMER_C_INP5	PIO0_22	Function 4
	CTIMER_C_INP9	PIO0_0	Function 4
CTIMER_S	CTIMER_S_INP0	PIO8_0	Function 2
	CTIMER_S_INP0	PIO8_5	Function 3
	CTIMER_S_INP1	PIO8_3	Function 2
	CTIMER_S_INP1	PIO8_12	Function 3
	CTIMER_S_INP2	PIO8_8	Function 2
	CTIMER_S_INP3	PIO8_10	Function 2
	CTIMER_S_INP3	PIO8_13	Function 3
	CTIMER_S_INP4	PIO8_16	Function 2
	CTIMER_S_INP4	PIO8_18	Function 3
	CTIMER_S_INP5	PIO8_17	Function 2
	CTIMER_S_INP6	PIO8_19	Function 3
	CTIMER_S_INP7	PIO8_20	Function 2
	CTIMER_S_INP9	PIO8_30	Function 2
CTIMER0	CTIMER0_MAT0	PIO0_1	Function 4
	CTIMER0_MAT1	PIO0_2	Function 4
	CTIMER0_MAT2	PIO0_3	Function 4
	CTIMER0_MAT3	PIO0_4	Function 4
CTIMER1	CTIMER1_MAT0	PIO0_10	Function 4
	CTIMER1_MAT1	PIO0_11	Function 4
	CTIMER1_MAT1	PIO1_12	Function 4
	CTIMER1_MAT2	PIO0_12	Function 4
	CTIMER1_MAT3	PIO0_13	Function 4
CTIMER2	CTIMER2_MAT0	PIO0_5	Function 4
	CTIMER2_MAT1	PIO1_10	Function 4
	CTIMER2_MAT2	PIO0_19	Function 4
	CTIMER2_MAT2	PIO1_11	Function 4

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	CTIMER2_MAT3	PIO0_20	Function 4
	CTIMER2_MAT3	PIO1_13	Function 4
CTIMER3	CTIMER3_MAT0	PIO1_14	Function 4
	CTIMER3_MAT1	PIO1_15	Function 4
	CTIMER3_MAT2	PIO1_16	Function 4
	CTIMER3_MAT3	PIO1_17	Function 4
CTIMER4	CTIMER4_MAT0	PIO0_31	Function 4
	CTIMER4_MAT0	PIO1_18	Function 4
	CTIMER4_MAT1	PIO1_0	Function 4
CTIMER5	CTIMER5_MAT0	PIO8_1	Function 2
	CTIMER5_MAT0	PIO8_18	Function 2
	CTIMER5_MAT0	PIO8_21	Function 2
	CTIMER5_MAT1	PIO8_2	Function 2
	CTIMER5_MAT1	PIO8_19	Function 2
	CTIMER5_MAT1	PIO8_22	Function 2
	CTIMER5_MAT2	PIO8_4	Function 2
	CTIMER5_MAT2	PIO8_23	Function 2
	CTIMER5_MAT3	PIO8_5	Function 2
	CTIMER5_MAT3	PIO8_24	Function 2
CTIMER6	CTIMER6_MAT0	PIO8_6	Function 2
	CTIMER6_MAT0	PIO8_25	Function 2
	CTIMER6_MAT1	PIO8_7	Function 2
	CTIMER6_MAT1	PIO8_26	Function 2
	CTIMER6_MAT2	PIO8_9	Function 2
	CTIMER6_MAT3	PIO8_11	Function 2
CTIMER7	CTIMER7_MAT0	PIO8_12	Function 2
	CTIMER7_MAT0	PIO8_31	Function 2
	CTIMER7_MAT1	PIO8_13	Function 2
	CTIMER7_MAT2	PIO8_14	Function 2
	CTIMER7_MAT3	PIO8_15	Function 2
Debug	JTAG_TCK/SWCLK	PIO8_0	Function 5
	JTAG_TD0/SWO	PIO8_3	Function 5

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	JTAG_TDI	PIO8_2	Function 5
	JTAG_TMS/SWDIO	PIO8_1	Function 5
	JTAG_TRSTN	PIO8_4	Function 5
	TRACECLK	PIO1_10	Function 5
	TRACEDATA0	PIO1_11	Function 5
	TRACEDATA1	PIO1_12	Function 5
	TRACEDATA2	PIO1_13	Function 5
	TRACEDATA3	PIO1_14	Function 5
EZH	EZH_PIO0	PIO2_0	Function 4
	EZH_PIO1	PIO2_1	Function 4
	EZH_PIO10	PIO2_10	Function 4
	EZH_PIO11	PIO2_11	Function 4
	EZH_PIO12	PIO2_12	Function 4
	EZH_PIO13	PIO2_13	Function 4
	EZH_PIO14	PIO2_14	Function 4
	EZH_PIO15	PIO2_15	Function 4
	EZH_PIO16	PIO3_0	Function 4
	EZH_PIO17	PIO3_1	Function 4
	EZH_PIO18	PIO3_2	Function 4
	EZH_PIO19	PIO3_3	Function 4
	EZH_PIO2	PIO2_2	Function 4
	EZH_PIO3	PIO2_3	Function 4
	EZH_PIO4	PIO2_4	Function 4
	EZH_PIO5	PIO2_5	Function 4
	EZH_PIO6	PIO2_6	Function 4
	EZH_PIO7	PIO2_7	Function 4
	EZH_PIO8	PIO2_8	Function 4
	EZH_PIO9	PIO2_9	Function 4
FLEXIO	FLEXIO_D0	PIO2_0	Function 5
	FLEXIO_D1	PIO2_1	Function 5
	FLEXIO_D10	PIO2_10	Function 5
	FLEXIO_D11	PIO2_11	Function 5

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	FLEXIO_D12	PIO2_12	Function 5
	FLEXIO_D13	PIO2_13	Function 5
	FLEXIO_D14	PIO2_14	Function 5
	FLEXIO_D15	PIO2_15	Function 5
	FLEXIO_D2	PIO2_2	Function 5
	FLEXIO_D3	PIO2_3	Function 5
	FLEXIO_D4	PIO2_4	Function 5
	FLEXIO_D5	PIO2_5	Function 5
	FLEXIO_D6	PIO2_6	Function 5
	FLEXIO_D7	PIO2_7	Function 5
	FLEXIO_D8	PIO2_8	Function 5
	FLEXIO_D9	PIO2_9	Function 5
GPIO	PIO0_0	PIO0_0	Function 0
	PIO0_1	PIO0_1	Function 0
	PIO0_10	PIO0_10	Function 0
	PIO0_11	PIO0_11	Function 0
	PIO0_12	PIO0_12	Function 0
	PIO0_13	PIO0_13	Function 0
	PIO0_14	PIO0_14	Function 0
	PIO0_19	PIO0_19	Function 0
	PIO0_2	PIO0_2	Function 0
	PIO0_20	PIO0_20	Function 0
	PIO0_21	PIO0_21	Function 0
	PIO0_22	PIO0_22	Function 0
	PIO0_3	PIO0_3	Function 0
	PIO0_31	PIO0_31	Function 0
	PIO0_4	PIO0_4	Function 0
	PIO0_5	PIO0_5	Function 0
	PIO0_6	PIO0_6	Function 0
	PIO0_7	PIO0_7	Function 0
	PIO1_0	PIO1_0	Function 0
	PIO1_10	PIO1_10	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO1_11	PIO1_11	Function 0
	PIO1_12	PIO1_12	Function 0
	PIO1_13	PIO1_13	Function 0
	PIO1_14	PIO1_14	Function 0
	PIO1_15	PIO1_15	Function 0
	PIO1_16	PIO1_16	Function 0
	PIO1_17	PIO1_17	Function 0
	PIO1_18	PIO1_18	Function 0
	PIO1_6	PIO1_6	Function 0
	PIO1_7	PIO1_7	Function 0
	PIO1_8	PIO1_8	Function 0
	PIO1_9	PIO1_9	Function 0
	PIO10_0	PIO10_0	Function 0
	PIO10_1	PIO10_1	Function 0
	PIO10_12	PIO10_12	Function 0
	PIO10_13	PIO10_13	Function 0
	PIO10_2	PIO10_2	Function 0
	PIO10_3	PIO10_3	Function 0
	PIO10_4	PIO10_4	Function 0
	PIO10_5	PIO10_5	Function 0
	PIO10_6	PIO10_6	Function 0
	PIO10_7	PIO10_7	Function 0
	PIO10_8	PIO10_8	Function 0
	PIO10_9	PIO10_9	Function 0
	PIO2_0	PIO2_0	Function 0
	PIO2_1	PIO2_1	Function 0
	PIO2_10	PIO2_10	Function 0
	PIO2_11	PIO2_11	Function 0
	PIO2_12	PIO2_12	Function 0
	PIO2_13	PIO2_13	Function 0
	PIO2_14	PIO2_14	Function 0
	PIO2_15	PIO2_15	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO2_2	PIO2_2	Function 0
	PIO2_3	PIO2_3	Function 0
	PIO2_4	PIO2_4	Function 0
	PIO2_5	PIO2_5	Function 0
	PIO2_6	PIO2_6	Function 0
	PIO2_7	PIO2_7	Function 0
	PIO2_8	PIO2_8	Function 0
	PIO2_9	PIO2_9	Function 0
	PIO3_0	PIO3_0	Function 0
	PIO3_1	PIO3_1	Function 0
	PIO3_2	PIO3_2	Function 0
	PIO3_3	PIO3_3	Function 0
	PIO4_0	PIO4_0	Function 0
	PIO4_1	PIO4_1	Function 0
	PIO4_10	PIO4_10	Function 0
	PIO4_11	PIO4_11	Function 0
	PIO4_12	PIO4_12	Function 0
	PIO4_13	PIO4_13	Function 0
	PIO4_14	PIO4_14	Function 0
	PIO4_15	PIO4_15	Function 0
	PIO4_16	PIO4_16	Function 0
	PIO4_17	PIO4_17	Function 0
	PIO4_18	PIO4_18	Function 0
	PIO4_19	PIO4_19	Function 0
	PIO4_2	PIO4_2	Function 0
	PIO4_20	PIO4_20	Function 0
	PIO4_3	PIO4_3	Function 0
	PIO4_4	PIO4_4	Function 0
	PIO4_5	PIO4_5	Function 0
	PIO4_6	PIO4_6	Function 0
	PIO4_7	PIO4_7	Function 0
	PIO4_8	PIO4_8	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO4_9	PIO4_9	Function 0
	PIO5_0	PIO5_0	Function 0
	PIO5_1	PIO5_1	Function 0
	PIO5_10	PIO5_10	Function 0
	PIO5_11	PIO5_11	Function 0
	PIO5_12	PIO5_12	Function 0
	PIO5_13	PIO5_13	Function 0
	PIO5_14	PIO5_14	Function 0
	PIO5_15	PIO5_15	Function 0
	PIO5_16	PIO5_16	Function 0
	PIO5_17	PIO5_17	Function 0
	PIO5_18	PIO5_18	Function 0
	PIO5_19	PIO5_19	Function 0
	PIO5_2	PIO5_2	Function 0
	PIO5_20	PIO5_20	Function 0
	PIO5_3	PIO5_3	Function 0
	PIO5_4	PIO5_4	Function 0
	PIO5_5	PIO5_5	Function 0
	PIO5_6	PIO5_6	Function 0
	PIO5_7	PIO5_7	Function 0
	PIO5_8	PIO5_8	Function 0
	PIO5_9	PIO5_9	Function 0
	PIO7_0	PIO7_0	Function 0
	PIO7_1	PIO7_1	Function 0
	PIO7_10	PIO7_10	Function 0
	PIO7_11	PIO7_11	Function 0
	PIO7_12	PIO7_12	Function 0
	PIO7_13	PIO7_13	Function 0
	PIO7_14	PIO7_14	Function 0
	PIO7_15	PIO7_15	Function 0
	PIO7_16	PIO7_16	Function 0
	PIO7_17	PIO7_17	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO7_18	PIO7_18	Function 0
	PIO7_19	PIO7_19	Function 0
	PIO7_2	PIO7_2	Function 0
	PIO7_20	PIO7_20	Function 0
	PIO7_21	PIO7_21	Function 0
	PIO7_22	PIO7_22	Function 0
	PIO7_23	PIO7_23	Function 0
	PIO7_25	PIO7_25	Function 0
	PIO7_3	PIO7_3	Function 0
	PIO7_4	PIO7_4	Function 0
	PIO7_5	PIO7_5	Function 0
	PIO7_6	PIO7_6	Function 0
	PIO7_7	PIO7_7	Function 0
	PIO7_8	PIO7_8	Function 0
	PIO7_9	PIO7_9	Function 0
	PIO8_0	PIO8_0	Function 0
	PIO8_1	PIO8_1	Function 0
	PIO8_10	PIO8_10	Function 0
	PIO8_11	PIO8_11	Function 0
	PIO8_12	PIO8_12	Function 0
	PIO8_13	PIO8_13	Function 0
	PIO8_14	PIO8_14	Function 0
	PIO8_15	PIO8_15	Function 0
	PIO8_16	PIO8_16	Function 0
	PIO8_17	PIO8_17	Function 0
	PIO8_18	PIO8_18	Function 0
	PIO8_19	PIO8_19	Function 0
	PIO8_2	PIO8_2	Function 0
	PIO8_20	PIO8_20	Function 0
	PIO8_21	PIO8_21	Function 0
	PIO8_22	PIO8_22	Function 0
	PIO8_23	PIO8_23	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	PIO8_24	PIO8_24	Function 0
	PIO8_25	PIO8_25	Function 0
	PIO8_26	PIO8_26	Function 0
	PIO8_3	PIO8_3	Function 0
	PIO8_30	PIO8_30	Function 0
	PIO8_31	PIO8_31	Function 0
	PIO8_4	PIO8_4	Function 0
	PIO8_5	PIO8_5	Function 0
	PIO8_6	PIO8_6	Function 0
	PIO8_7	PIO8_7	Function 0
	PIO8_8	PIO8_8	Function 0
	PIO8_9	PIO8_9	Function 0
I3C0	I3C0_PUR	PIO1_16	Function 5
	I3C0_SCL	PIO1_17	Function 5
	I3C0_SDA	PIO1_18	Function 5
	I3C1_PUR	PIO1_6	Function 5
	I3C1_SCL	PIO1_8	Function 5
	I3C1_SDA	PIO1_7	Function 5
	I3C2_PUR	PIO8_16	Function 4
	I3C2_SCL	PIO8_15	Function 4
	I3C2_SDA	PIO8_14	Function 4
	I3C3_PUR	PIO8_20	Function 4
	I3C3_SCL	PIO8_22	Function 4
	I3C3_SDA	PIO8_21	Function 4
LCDIF	LCD_D0	PIO2_6	Function 6
	LCD_D1	PIO2_7	Function 6
	LCD_D10	PIO4_14	Function 6
	LCD_D11	PIO4_15	Function 6
	LCD_D12	PIO4_17	Function 6
	LCD_D13	PIO4_18	Function 6
	LCD_D14	PIO4_19	Function 6
	LCD_D15	PIO4_20	Function 6

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LCD_D16	PIO4_6	Function 6
	LCD_D17	PIO4_7	Function 6
	LCD_D18	PIO4_8	Function 6
	LCD_D19	PIO4_9	Function 6
	LCD_D2	PIO2_8	Function 6
	LCD_D20	PIO4_1	Function 6
	LCD_D21	PIO4_2	Function 6
	LCD_D22	PIO4_3	Function 6
	LCD_D23	PIO4_4	Function 6
	LCD_D3	PIO2_9	Function 6
	LCD_D4	PIO2_10	Function 6
	LCD_D5	PIO2_11	Function 6
	LCD_D6	PIO2_12	Function 6
	LCD_D7	PIO2_13	Function 6
	LCD_D8	PIO4_12	Function 6
	LCD_D9	PIO4_13	Function 6
	LCD_DBI_CSX_AB	PIO2_0	Function 7
	LCD_DBI-D0	PIO2_6	Function 7
	LCD_DBI-D1	PIO2_7	Function 7
	LCD_DBI-D10	PIO4_14	Function 7
	LCD_DBI-D11	PIO4_15	Function 7
	LCD_DBI-D12	PIO4_17	Function 7
	LCD_DBI-D13	PIO4_18	Function 7
	LCD_DBI-D14	PIO4_19	Function 7
	LCD_DBI-D15	PIO4_20	Function 7
	LCD_DBI-D2	PIO2_8	Function 7
	LCD_DBI-D3	PIO2_9	Function 7
	LCD_DBI-D4	PIO2_10	Function 7
	LCD_DBI-D5	PIO2_11	Function 7
	LCD_DBI-D6	PIO2_12	Function 7
	LCD_DBI-D7	PIO2_13	Function 7
	LCD_DBI-D8	PIO4_12	Function 7

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LCD_DBI-D9	PIO4_13	Function 7
	LCD_DBI-DATA_OEN	PIO2_2	Function 7
	LCD_DBI-DCX-AB	PIO2_1	Function 7
	LCD_DBI-E	PIO2_5	Function 7
	LCD_DBI-RWDX	PIO2_3	Function 7
	LCD_DBI-WRX	PIO2_4	Function 7
	LCD_DOTCLK	PIO2_3	Function 6
	LCD_ENABLE	PIO2_2	Function 6
	LCD_HSYNC	PIO2_4	Function 6
	LCD_VSYNC	PIO2_5	Function 6
LP_FLEXCOMM0	LP_FLEXCOMM0_P0	PIO0_31	Function 1
	LP_FLEXCOMM0_P1	PIO1_0	Function 1
LP_FLEXCOMM0 (BOOT)	LPSPI0_PCS0 (SPI boot)	PIO3_3	Alternate pin function
	LPSPI0_SCK (SPI boot)	PIO3_1	Alternate pin function
	LPSPI0_SIN (SPI boot)	PIO3_2	Alternate pin function
	LPSPI0_SOUT (SPI boot)	PIO3_0	Alternate pin function
	LP_FLEXCOMM0_P0 (UART boot)	PIO0_31	Alternate pin function
	LP_FLEXCOMM0_P1 (UART boot)	PIO1_0	Alternate pin function
LP_FLEXCOMM1	LP_FLEXCOMM1_P0	PIO1_7	Function 1
	LP_FLEXCOMM1_P0	PIO7_10	Function 2
	LP_FLEXCOMM1_P1	PIO1_8	Function 1
	LP_FLEXCOMM1_P1	PIO7_11	Function 2
	LP_FLEXCOMM1_P2	PIO1_9	Function 1
	LP_FLEXCOMM1_P2	PIO7_12	Function 2
	LP_FLEXCOMM1_P3	PIO1_10	Function 1
	LP_FLEXCOMM1_P3	PIO7_13	Function 2
	LP_FLEXCOMM1_P4	PIO1_6	Function 1
	LP_FLEXCOMM1_P4	PIO7_19	Function 2
	LP_FLEXCOMM1_P5	PIO7_20	Function 2
	LP_FLEXCOMM1_P6	PIO7_21	Function 2
LP_FLEXCOMM10	LP_FLEXCOMM10_P0	PIO4_14	Function 2
	LP_FLEXCOMM10_P1	PIO4_15	Function 2

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM10_P2	PIO4_16	Function 2
	LP_FLEXCOMM10_P3	PIO4_17	Function 2
	LP_FLEXCOMM10_P4	PIO4_18	Function 2
	LP_FLEXCOMM10_P5	PIO4_19	Function 2
	LP_FLEXCOMM10_P6	PIO4_20	Function 2
LP_FLEXCOMM11	LP_FLEXCOMM11_P0	PIO5_14	Function 2
	LP_FLEXCOMM11_P1	PIO5_15	Function 2
	LP_FLEXCOMM11_P2	PIO5_16	Function 2
	LP_FLEXCOMM11_P3	PIO5_17	Function 2
	LP_FLEXCOMM11_P4	PIO5_18	Function 2
	LP_FLEXCOMM11_P5	PIO5_19	Function 2
	LP_FLEXCOMM11_P6	PIO5_20	Function 2
LP_FLEXCOMM12	LP_FLEXCOMM12_P0	PIO5_7	Function 2
	LP_FLEXCOMM12_P0	PIO7_2	Function 2
	LP_FLEXCOMM12_P1	PIO5_8	Function 2
	LP_FLEXCOMM12_P1	PIO7_3	Function 2
	LP_FLEXCOMM12_P2	PIO5_9	Function 2
	LP_FLEXCOMM12_P2	PIO7_4	Function 2
	LP_FLEXCOMM12_P3	PIO5_6	Function 2
	LP_FLEXCOMM12_P3	PIO7_5	Function 2
	LP_FLEXCOMM12_P4	PIO5_11	Function 2
	LP_FLEXCOMM12_P4	PIO7_6	Function 2
	LP_FLEXCOMM12_P5	PIO5_12	Function 2
	LP_FLEXCOMM12_P5	PIO7_7	Function 2
	LP_FLEXCOMM12_P6	PIO5_13	Function 2
	LP_FLEXCOMM12_P6	PIO7_8	Function 2
LP_FLEXCOMM13	LP_FLEXCOMM13_P0	PIO5_3	Function 2
	LP_FLEXCOMM13_P1	PIO5_4	Function 2
	LP_FLEXCOMM13_P2	PIO5_5	Function 2
	LP_FLEXCOMM13_P3	PIO5_10	Function 2
	LP_FLEXCOMM13_P4	PIO5_0	Function 2
	LP_FLEXCOMM13_P5	PIO5_1	Function 2

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM13_P5	PIO7_25	Function 2
	LP_FLEXCOMM13_P6	PIO5_2	Function 2
LP_FLEXCOMM14 (LPSPI14)	LPSPI14_PCS0	PIO3_3	Function 1
	LPSPI14_SCK	PIO3_1	Function 1
	LPSPI14_SIN	PIO3_2	Function 1
	LPSPI14_SOUT	PIO3_0	Function 1
LP_FLEXCOMM15 (LPI2C15)	PMIC_I2C_SCL	PMIC_I2C_SCL	Function 0
	PMIC_I2C_SCL	PMIC_I2C_SCL	Alternate pin function
	PMIC_I2C_SDA	PMIC_I2C_SDA	Function 0
	PMIC_I2C_SDA	PMIC_I2C_SDA	Alternate pin function
LP_FLEXCOMM16 (LPSPI16)	LPSPI16_PCS0	PIO2_5	Function 1
	LPSPI16_PCS1	PIO2_8	Function 1
	LPSPI16_PCS2	PIO2_7	Function 1
	LPSPI16_PCS3	PIO2_6	Function 1
	LPSPI16_SCK	PIO2_3	Function 1
	LPSPI16_SIN	PIO2_4	Function 1
	LPSPI16_SOUT	PIO2_2	Function 1
LP_FLEXCOMM17	LP_FLEXCOMM17_P0	PIO8_5	Function 1
	LP_FLEXCOMM17_P1	PIO8_6	Function 1
	LP_FLEXCOMM17_P2	PIO8_2	Function 1
	LP_FLEXCOMM17_P2	PIO8_12	Function 4
	LP_FLEXCOMM17_P3	PIO8_3	Function 1
	LP_FLEXCOMM17_P3	PIO8_13	Function 4
	LP_FLEXCOMM17_P4	PIO8_4	Function 1
	LP_FLEXCOMM17_P4	PIO8_11	Function 4
	LP_FLEXCOMM17_P5	PIO8_0	Function 1
	LP_FLEXCOMM17_P6	PIO8_1	Function 1
LP_FLEXCOMM18	LP_FLEXCOMM18_P0	PIO8_7	Function 1
	LP_FLEXCOMM18_P1	PIO8_8	Function 1
	LP_FLEXCOMM18_P2	PIO8_9	Function 1
	LP_FLEXCOMM18_P3	PIO8_10	Function 1
	LP_FLEXCOMM18_P4	PIO8_11	Function 1

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM18_P5	PIO8_12	Function 1
	LP_FLEXCOMM18_P6	PIO8_13	Function 1
LP_FLEXCOMM19	LP_FLEXCOMM19_P0	PIO8_14	Function 1
	LP_FLEXCOMM19_P0	PIO8_23	Function 3
	LP_FLEXCOMM19_P1	PIO8_15	Function 1
	LP_FLEXCOMM19_P1	PIO8_24	Function 3
	LP_FLEXCOMM19_P2	PIO8_16	Function 1
	LP_FLEXCOMM19_P2	PIO8_25	Function 3
	LP_FLEXCOMM19_P3	PIO8_17	Function 1
	LP_FLEXCOMM19_P3	PIO8_26	Function 3
	LP_FLEXCOMM19_P4	PIO8_18	Function 1
	LP_FLEXCOMM19_P5	PIO8_19	Function 1
	LP_FLEXCOMM19_P6	PIO8_20	Function 1
LP_FLEXCOMM2	LP_FLEXCOMM2_P0	PIO1_11	Function 1
	LP_FLEXCOMM2_P0	PIO7_17	Function 2
	LP_FLEXCOMM2_P1	PIO1_12	Function 1
	LP_FLEXCOMM2_P1	PIO7_18	Function 2
	LP_FLEXCOMM2_P2	PIO1_14	Function 1
	LP_FLEXCOMM2_P2	PIO7_22	Function 2
	LP_FLEXCOMM2_P3	PIO1_15	Function 1
	LP_FLEXCOMM2_P3	PIO7_23	Function 2
	LP_FLEXCOMM2_P4	PIO1_13	Function 1
	LP_FLEXCOMM2_P4	PIO7_20	Function 3
	LP_FLEXCOMM2_P5	PIO7_21	Function 3
	LP_FLEXCOMM2_P6	PIO7_25	Function 3
LP_FLEXCOMM20	LP_FLEXCOMM20_P0	PIO8_30	Function 1
	LP_FLEXCOMM20_P1	PIO8_31	Function 1
	LP_FLEXCOMM20_P3	PIO8_21	Function 1
	LP_FLEXCOMM20_P4	PIO8_22	Function 1
LP_FLEXCOMM3	LP_FLEXCOMM3_P0	PIO1_18	Function 1
	LP_FLEXCOMM3_P0	PIO8_21	Function 3
	LP_FLEXCOMM3_P1	PIO1_17	Function 1

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM3_P1	PIO8_22	Function 3
	LP_FLEXCOMM3_P2	PIO1_16	Function 1
	LP_FLEXCOMM3_P4	PIO7_0	Function 2
	LP_FLEXCOMM3_P5	PIO7_1	Function 2
	LP_FLEXCOMM3_P6	PIO7_9	Function 2
LP_FLEXCOMM3 (BOOT)	LP_FLEXCOMM3_P0 (I2C boot)	PIO1_18	Alternate pin function
	LP_FLEXCOMM3_P1 (I2C boot)	PIO1_17	Alternate pin function
LP_FLEXCOMM4	LP_FLEXCOMM4_P0	PIO0_3	Function 1
	LP_FLEXCOMM4_P1	PIO0_4	Function 1
	LP_FLEXCOMM4_P2	PIO0_6	Function 1
	LP_FLEXCOMM4_P3	PIO0_7	Function 1
	LP_FLEXCOMM4_P4	PIO0_5	Function 1
LP_FLEXCOMM5	LP_FLEXCOMM5_P0	PIO0_10	Function 1
	LP_FLEXCOMM5_P1	PIO0_11	Function 1
	LP_FLEXCOMM5_P2	PIO0_12	Function 1
	LP_FLEXCOMM5_P3	PIO0_13	Function 1
	LP_FLEXCOMM5_P4	PIO0_14	Function 1
LP_FLEXCOMM6	LP_FLEXCOMM6_P0	PIO0_21	Function 1
	LP_FLEXCOMM6_P1	PIO0_22	Function 1
	LP_FLEXCOMM6_P2	PIO0_19	Function 1
	LP_FLEXCOMM6_P3	PIO0_20	Function 1
LP_FLEXCOMM7	LP_FLEXCOMM7_P0	PIO7_14	Function 2
	LP_FLEXCOMM7_P1	PIO7_15	Function 2
	LP_FLEXCOMM7_P2	PIO7_16	Function 2
	LP_FLEXCOMM7_P3	PIO7_8	Function 3
	LP_FLEXCOMM7_P3	PIO7_19	Function 3
	LP_FLEXCOMM7_P4	PIO7_9	Function 3
	LP_FLEXCOMM7_P5	PIO7_1	Function 3
	LP_FLEXCOMM7_P6	PIO7_0	Function 3
LP_FLEXCOMM8	LP_FLEXCOMM8_P0	PIO4_0	Function 2
	LP_FLEXCOMM8_P0	PIO0_6	Function 5
	LP_FLEXCOMM8_P1	PIO4_1	Function 2

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	LP_FLEXCOMM8_P1	PIO0_7	Function 5
	LP_FLEXCOMM8_P2	PIO0_0	Function 1
	LP_FLEXCOMM8_P2	PIO4_2	Function 2
	LP_FLEXCOMM8_P3	PIO0_1	Function 1
	LP_FLEXCOMM8_P3	PIO4_3	Function 2
	LP_FLEXCOMM8_P4	PIO0_2	Function 1
	LP_FLEXCOMM8_P4	PIO4_4	Function 2
	LP_FLEXCOMM8_P5	PIO4_5	Function 2
	LP_FLEXCOMM8_P6	PIO4_6	Function 2
LP_FLEXCOMM9	LP_FLEXCOMM9_P0	PIO4_7	Function 2
	LP_FLEXCOMM9_P0	PIO0_2	Function 7
	LP_FLEXCOMM9_P1	PIO4_8	Function 2
	LP_FLEXCOMM9_P1	PIO0_5	Function 7
	LP_FLEXCOMM9_P2	PIO4_9	Function 2
	LP_FLEXCOMM9_P3	PIO4_10	Function 2
	LP_FLEXCOMM9_P4	PIO4_11	Function 2
	LP_FLEXCOMM9_P5	PIO4_12	Function 2
	LP_FLEXCOMM9_P6	PIO4_13	Function 2
MICFIL	PDM_CLK	PIO10_0	Function 5
	PDM_CLK	PIO10_8	Function 5
	PDM_CLK	PIO10_13	Function 5
	PDM_DATA01	PIO10_1	Function 5
	PDM_DATA01	PIO10_9	Function 5
	PDM_DATA23	PIO10_2	Function 5
	PDM_DATA45	PIO10_3	Function 5
	PDM_DATA67	PIO10_4	Function 5
	PDM_DATA67	PIO10_12	Function 5
MIPI_DSI_HOST	MIPI_DSI_CLKN	MIPI_DSI_CLKN	Function 0
	MIPI_DSI_CLKN	MIPI_DSI_CLKN	Alternate pin function
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	Function 0
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	Alternate pin function
	MIPI_DSI_D0N	MIPI_DSI_D0N	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	MIPI_DSI_D0N	MIPI_DSI_D0N	Alternate pin function
	MIPI_DSI_D0P	MIPI_DSI_D0P	Function 0
	MIPI_DSI_D0P	MIPI_DSI_D0P	Alternate pin function
	MIPI_DSI_D1N	MIPI_DSI_D1N	Function 0
	MIPI_DSI_D1N	MIPI_DSI_D1N	Alternate pin function
	MIPI_DSI_D1P	MIPI_DSI_D1P	Function 0
	MIPI_DSI_D1P	MIPI_DSI_D1P	Alternate pin function
OSC32KNP	32KHZ_CLKOUT	PIO1_3	Function 6
	32KHZ_CLKOUT	PIO1_6	Function 6
	32KHZ_CLKOUT	PIO1_13	Function 6
	EXTAL	XTALIN32KHZ	Function 0
	XTAL	XTALOUT32KHZ	Function 0
PMC	PMIC_IRQN	PMIC_IRQN	Function 0
	PMIC_IRQN	PMIC_IRQN	Alternate pin function
	PMIC_MODE0	PMIC_MODE0	Function 0
	PMIC_MODE0	PMIC_MODE0	Alternate pin function
	PMIC_MODE1	PMIC_MODE1	Function 0
	PMIC_MODE1	PMIC_MODE1	Alternate pin function
	RESETN	RESETN	Function 0
	RESETN	RESETN	Alternate pin function
RTC	RTCXIN	XTALINSYS	Alternate pin function
	RTCXOUT	XTALOUTSYS	Alternate pin function
SAI0[1]	SAI0_RX_BCLK	PIO0_0	Function 5
	SAI0_RX_DATA	PIO0_2	Function 5
	SAI0_RX_SYNC	PIO0_1	Function 5
	SAI0_TX_BCLK	PIO0_3	Function 5
	SAI0_TX_DATA	PIO0_4	Function 5
	SAI0_TX_SYNC	PIO0_5	Function 5
SAI1[1]	SAI1_RX_DATA	PIO0_1	Function 6
	SAI1_RX_SYNC	PIO0_19	Function 5
	SAI1_TX_BCLK	PIO0_20	Function 5
	SAI1_TX_BCLK	PIO0_0	Function 6

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	SAI1_TX_DATA	PIO0_21	Function 5
	SAI1_TX_SYNC	PIO0_22	Function 5
SAI2[1]	SAI2_RX_DATA	PIO0_14	Function 5
	SAI2_RX_DATA	PIO0_10	Function 6
	SAI2_TX_BCLK	PIO0_11	Function 5
	SAI2_TX_DATA	PIO0_12	Function 5
	SAI2_TX_SYNC	PIO0_13	Function 5
SAI3[1]	SAI3_RX_BCLK	PIO8_23	Function 1
	SAI3_RX_DATA	PIO8_24	Function 1
	SAI3_RX_SYNC	PIO8_25	Function 1
	SAI3_RX_SYNC	PIO8_17	Function 3
	SAI3_TX_BCLK	PIO8_26	Function 1
	SAI3_TX_BCLK	PIO8_15	Function 3
	SAI3_TX_DATA	PIO8_16	Function 3
	SAI3_TX_SYNC	PIO8_14	Function 3
SCT0	SCT0_GPIN0	PIO0_0	Function 2
	SCT0_GPIN0	PIO0_14	Function 2
	SCT0_GPIN1	PIO0_1	Function 2
	SCT0_GPIN2	PIO0_2	Function 2
	SCT0_GPIN2	PIO0_19	Function 2
	SCT0_GPIN3	PIO0_3	Function 2
	SCT0_GPIN3	PIO0_20	Function 2
	SCT0_GPIN3	PIO1_12	Function 2
	SCT0_GPIN4	PIO0_4	Function 2
	SCT0_GPIN4	PIO0_10	Function 2
	SCT0_GPIN4	PIO0_22	Function 2
	SCT0_GPIN4	PIO1_6	Function 2
	SCT0_GPIN5	PIO0_5	Function 2
	SCT0_GPIN5	PIO0_11	Function 2
	SCT0_GPIN5	PIO1_7	Function 2
	SCT0_GPIN6	PIO0_6	Function 2
	SCT0_GPIN6	PIO0_12	Function 2

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	SCT0_GPIN6	PIO0_21	Function 2
	SCT0_GPIN6	PIO1_8	Function 2
	SCT0_GPIN7	PIO0_7	Function 2
	SCT0_GPIN7	PIO0_13	Function 2
	SCT0_GPIN7	PIO1_9	Function 2
	SCT0_OUT0	PIO0_0	Function 3
	SCT0_OUT0	PIO1_6	Function 3
	SCT0_OUT1	PIO0_1	Function 3
	SCT0_OUT1	PIO1_7	Function 3
	SCT0_OUT2	PIO0_2	Function 3
	SCT0_OUT2	PIO0_19	Function 3
	SCT0_OUT2	PIO1_8	Function 3
	SCT0_OUT3	PIO0_3	Function 3
	SCT0_OUT3	PIO0_20	Function 3
	SCT0_OUT3	PIO1_9	Function 3
	SCT0_OUT4	PIO0_4	Function 3
	SCT0_OUT4	PIO0_10	Function 3
	SCT0_OUT4	PIO0_22	Function 3
	SCT0_OUT5	PIO0_5	Function 3
	SCT0_OUT5	PIO0_11	Function 3
	SCT0_OUT6	PIO0_6	Function 3
	SCT0_OUT6	PIO0_12	Function 3
	SCT0_OUT6	PIO0_21	Function 3
	SCT0_OUT7	PIO0_7	Function 3
	SCT0_OUT7	PIO0_13	Function 3
	SCT0_OUT8	PIO0_14	Function 3
	SCT0_OUT8	PIO0_31	Function 3
	SCT0_OUT9	PIO1_0	Function 3
	SCT0_OUT9	PIO1_12	Function 3
SDADC	SDADC_INN0	PIO10_7	Alternate pin function
	SDADC_INP0	PIO10_6	Alternate pin function
USB0	USB0_DM	USB0_DM	Function 0

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	USB0_DP	USB0_DP	Function 0
	USB0_OVERCURRENTN	PIO1_17	Function 2
	USB0_OVERCURRENTN	PIO2_0	Function 2
	USB0_PORTPWRN	PIO1_18	Function 2
	USB0_PORTPWRN	PIO2_1	Function 2
	USB0_VBUS	USB0_VBUS	Function 0
USB1	USB1_DM	USB0_DM	Alternate pin function
	USB1_DP	USB0_DP	Alternate pin function
	USB1_VBUS	USB0_VBUS	Alternate pin function
uSDHC0	SDHC0_CARD_DET_N	PIO7_11	Function 1
	SDHC0_CLK	PIO7_0	Function 1
	SDHC0_CMD	PIO7_1	Function 1
	SDHC0_DATA0	PIO7_6	Function 1
	SDHC0_DATA1	PIO7_7	Function 1
	SDHC0_DATA2	PIO7_8	Function 1
	SDHC0_DATA3	PIO7_9	Function 1
	SDHC0_DATA4	PIO7_2	Function 1
	SDHC0_DATA5	PIO7_3	Function 1
	SDHC0_DATA6	PIO7_4	Function 1
	SDHC0_DATA7	PIO7_5	Function 1
	SDHC0_DS	PIO7_10	Function 5
	SDHC0_RESET_N	PIO7_12	Function 1
	SDHC0_WR_PRT	PIO7_10	Function 1
uSDHC0 (BOOT)	SDHC0_CLK (SDHC0 boot)	PIO7_0	Alternate pin function
	SDHC0_CMD (SDHC0 boot)	PIO7_1	Alternate pin function
	SDHC0_DATA0 (SDHC0 boot)	PIO7_6	Alternate pin function
	SDHC0_DATA1 (SDHC0 boot)	PIO7_7	Alternate pin function
	SDHC0_DATA2 (SDHC0 boot)	PIO7_8	Alternate pin function
	SDHC0_DATA3 (SDHC0 boot)	PIO7_9	Alternate pin function
uSDHC1	SDHC1_CLK	PIO7_17	Function 1
	SDHC1_CMD	PIO7_18	Function 1
	SDHC1_DATA0	PIO7_19	Function 1

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	SDHC1_DATA1	PIO7_20	Function 1
	SDHC1_DATA2	PIO7_21	Function 1
	SDHC1_DATA3	PIO7_22	Function 1
	SDHC1_DATA4	PIO7_13	Function 1
	SDHC1_DATA5	PIO7_14	Function 1
	SDHC1_DATA6	PIO7_15	Function 1
	SDHC1_DATA7	PIO7_16	Function 1
	SDHC1_DS	PIO7_23	Function 5
	SDHC1_RESET_N	PIO7_25	Function 1
	SDHC1_WR_PRT	PIO7_23	Function 1
uSDHC1 (BOOT)	SDHC1_CLK (SDHC1 boot)	PIO7_17	Alternate pin function
	SDHC1_CMD (SDHC1 boot)	PIO7_18	Alternate pin function
	SDHC1_DATA0 (SDHC1 boot)	PIO7_19	Alternate pin function
	SDHC1_DATA1 (SDHC1 boot)	PIO7_20	Alternate pin function
	SDHC1_DATA2 (SDHC1 boot)	PIO7_21	Alternate pin function
	SDHC1_DATA3 (SDHC1 boot)	PIO7_22	Alternate pin function
UTICK0	UTICK0_CAP1	PIO1_13	Function 2
	UTICK0_CAP2	PIO0_31	Function 2
	UTICK0_CAP3	PIO1_10	Function 2
UTICK1	UTICK1_CAP0	PIO8_3	Function 3
	UTICK1_CAP1	PIO8_11	Function 3
	UTICK1_CAP2	PIO8_20	Function 3
	UTICK1_CAP3	PIO8_30	Function 3
XSPI1	XSPI1_DATA0	PIO5_1	Function 1
	XSPI1_DATA1	PIO5_2	Function 1
	XSPI1_DATA10	PIO5_14	Function 1
	XSPI1_DATA11	PIO5_15	Function 1
	XSPI1_DATA12	PIO5_17	Function 1
	XSPI1_DATA13	PIO5_18	Function 1
	XSPI1_DATA14	PIO5_19	Function 1
	XSPI1_DATA15	PIO5_20	Function 1
	XSPI1_DATA2	PIO5_3	Function 1

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	XSPI1_DATA3	PIO5_4	Function 1
	XSPI1_DATA4	PIO5_6	Function 1
	XSPI1_DATA5	PIO5_7	Function 1
	XSPI1_DATA6	PIO5_8	Function 1
	XSPI1_DATA7	PIO5_9	Function 1
	XSPI1_DATA8	PIO5_12	Function 1
	XSPI1_DATA9	PIO5_13	Function 1
	XSPI1_DQS0	PIO5_5	Function 1
	XSPI1_DQS1	PIO5_16	Function 1
	XSPI1_SCLK0	PIO5_10	Function 1
	XSPI1_SCLK0_N	PIO5_11	Function 1
	XSPI1_SS0_N	PIO5_0	Function 1
	XSPI1_SS1_N	PIO5_11	Function 3
XSPI1 (BOOT)	XSPI1_DATA0 (XSPI1-PSRAM boot)	PIO5_1	Alternate pin function
	XSPI1_DATA1 (XSPI1-PSRAM boot)	PIO5_2	Alternate pin function
	XSPI1_DATA10 (XSPI1-PSRAM boot)	PIO5_14	Alternate pin function
	XSPI1_DATA11 (XSPI1-PSRAM boot)	PIO5_15	Alternate pin function
	XSPI1_DATA12 (XSPI1-PSRAM boot)	PIO5_17	Alternate pin function
	XSPI1_DATA13 (XSPI1-PSRAM boot)	PIO5_18	Alternate pin function
	XSPI1_DATA14 (XSPI1-PSRAM boot)	PIO5_19	Alternate pin function
	XSPI1_DATA15 (XSPI1-PSRAM boot)	PIO5_20	Alternate pin function
	XSPI1_DATA2 (XSPI1-PSRAM boot)	PIO5_3	Alternate pin function
	XSPI1_DATA3 (XSPI1-PSRAM boot)	PIO5_4	Alternate pin function
	XSPI1_DATA4 (XSPI1-PSRAM boot)	PIO5_6	Alternate pin function
	XSPI1_DATA5 (XSPI1-PSRAM boot)	PIO5_7	Alternate pin function
	XSPI1_DATA6 (XSPI1-PSRAM boot)	PIO5_8	Alternate pin function
	XSPI1_DATA7 (XSPI1-PSRAM boot)	PIO5_9	Alternate pin function
	XSPI1_DATA8 (XSPI1-PSRAM boot)	PIO5_12	Alternate pin function
	XSPI1_DATA9 (XSPI1-PSRAM boot)	PIO5_13	Alternate pin function
	XSPI1_DQS0 (XSPI1-PSRAM boot)	PIO5_5	Alternate pin function
	XSPI1_DQS1 (XSPI1-PSRAM boot)	PIO5_16	Alternate pin function
	XSPI1_SCLK0 (XSPI1-PSRAM boot)	PIO5_10	Alternate pin function

Table continues on the next page...

Table 18. Muxing Options (256WLCSP) (continued)

Module	External signal	Pad	FSEL assignment
	XSPI1_SCLK0_N (XSPI1-PSRAM boot)	PIO5_11	Alternate pin function
	XSPI1_SS0_N (XSPI1-PSRAM boot)	PIO5_0	Alternate pin function
XSPI2	XSPI2_DATA0	PIO4_1	Function 1
	XSPI2_DATA1	PIO4_2	Function 1
	XSPI2_DATA10	PIO4_14	Function 1
	XSPI2_DATA11	PIO4_15	Function 1
	XSPI2_DATA12	PIO4_17	Function 1
	XSPI2_DATA13	PIO4_18	Function 1
	XSPI2_DATA14	PIO4_19	Function 1
	XSPI2_DATA15	PIO4_20	Function 1
	XSPI2_DATA2	PIO4_3	Function 1
	XSPI2_DATA3	PIO4_4	Function 1
	XSPI2_DATA4	PIO4_6	Function 1
	XSPI2_DATA5	PIO4_7	Function 1
	XSPI2_DATA6	PIO4_8	Function 1
	XSPI2_DATA7	PIO4_9	Function 1
	XSPI2_DATA8	PIO4_12	Function 1
	XSPI2_DATA9	PIO4_13	Function 1
	XSPI2_DQS0	PIO4_5	Function 1
	XSPI2_DQS1	PIO4_16	Function 1
	XSPI2_SCLK0	PIO4_10	Function 1
	XSPI2_SCLK0_N	PIO4_11	Function 1
	XSPI2_SS0_N	PIO4_0	Function 1
	XSPI2_SS1_N	PIO4_11	Function 3

1. The MCLK indirectly connects SAI via CLKCTL, see CLKCTL for more details.

6.3 Pin assignments

6.3.1 Pin assignments (324FOWLP)

Table 19. Pin Assignments (324FOWLP)

Pin name	FSEL assignments	Pad settings
PIO0_0	Function 0 - PIO0_0 Function 1 - LP_FLEXCOMM8_P2	DRIVE - 100 Ω IIENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT0 Function 4 - CTIMER_C_INP9 Function 5 - SAI0_RX_BCLK Function 6 - SAI1_TX_BCLK	ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_1	Function 0 - PIO0_1 Function 1 - LP_FLEXCOMM8_P3 Function 2 - SCT0_GPIN1 Function 3 - SCT0_OUT1 Function 4 - CTIMER0_MAT0 Function 5 - SAI0_RX_SYNC Function 6 - SAI1_RX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_2	Function 0 - PIO0_2 Function 1 - LP_FLEXCOMM8_P4 Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT2 Function 4 - CTIMER0_MAT1 Function 5 - SAI0_RX_DATA Function 7 - LP_FLEXCOMM9_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_3	Function 0 - PIO0_3 Function 1 - LP_FLEXCOMM4_P0 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT3 Function 4 - CTIMER0_MAT2 Function 5 - SAI0_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_4	Function 0 - PIO0_4 Function 1 - LP_FLEXCOMM4_P1 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER0_MAT3 Function 5 - SAI0_TX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_5	Function 0 - PIO0_5	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM4_P4 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT5 Function 4 - CTIMER2_MAT0 Function 5 - SAI0_TX_SYNC Function 6 - MCLK Function 7 - LP_FLEXCOMM9_P1	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_6	Function 0 - PIO0_6 Function 1 - LP_FLEXCOMM4_P2 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER_C_INP10 Function 5 - LP_FLEXCOMM8_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_7	Function 0 - PIO0_7 Function 1 - LP_FLEXCOMM4_P3 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT7 Function 4 - CTIMER_C_INP11 Function 5 - LP_FLEXCOMM8_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_9	Function 0 - PIO0_9 Function 1 - LP_FLEXCOMM4_P6 Function 2 - SCT0_GPIN1 Function 3 - SCT0_OUT1 Function 4 - CTIMER0_MAT0 Function 5 - LP_FLEXCOMM1_P5	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_10	Function 0 - PIO0_10 Function 1 - LP_FLEXCOMM5_P0 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER1_MAT0 Function 5 - CTIMER_C_INP0 Function 6 - SAI2_RX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO0_11	Function 0 - PIO0_11 Function 1 - LP_FLEXCOMM5_P1 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT5 Function 4 - CTIMER1_MAT1 Function 5 - SAI2_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_12	Function 0 - PIO0_12 Function 1 - LP_FLEXCOMM5_P2 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER1_MAT2 Function 5 - SAI2_TX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_13	Function 0 - PIO0_13 Function 1 - LP_FLEXCOMM5_P3 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT7 Function 4 - CTIMER1_MAT3 Function 5 - SAI2_TX_SYNC	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_14	Function 0 - PIO0_14 Function 1 - LP_FLEXCOMM5_P4 Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT8 Function 4 - CTIMER_C_INP2 Function 5 - SAI2_RX_DATA Function 6 - CLKOUT_VDD2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_15	Function 0 - PIO0_15 Function 1 - LP_FLEXCOMM5_P5 Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT2 Function 4 - CTIMER_C_INP3 Function 5 - SAI2_RX_BCLK Function 6 - LP_FLEXCOMM2_P5	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO0_16	Function 0 - PIO0_16 Function 1 - LP_FLEXCOMM5_P6 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT3 Function 4 - CTIMER0_MAT1 Function 5 - SAI2_RX_SYNC Function 6 - LP_FLEXCOMM2_P6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_17	Function 0 - PIO0_17 Function 1 - LP_FLEXCOMM6_P5 Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT0 Function 4 - CTIMER2_MAT0 Function 5 - SAI1_RX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_18	Function 0 - PIO0_18 Function 1 - LP_FLEXCOMM6_P6 Function 2 - SCT0_GPIN1 Function 3 - SCT0_OUT1 Function 4 - CTIMER2_MAT1 Function 5 - SAI1_RX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_19	Function 0 - PIO0_19 Function 1 - LP_FLEXCOMM6_P2 Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT2 Function 4 - CTIMER2_MAT2 Function 5 - SAI1_RX_SYNC	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_20	Function 0 - PIO0_20 Function 1 - LP_FLEXCOMM6_P3 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT3 Function 4 - CTIMER2_MAT3 Function 5 - SAI1_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_21	Function 0 - PIO0_21	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM6_P0 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER_C_INP4 Function 5 - SAI1_TX_DATA Function 6 - MCLK	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_22	Function 0 - PIO0_22 Function 1 - LP_FLEXCOMM6_P1 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER_C_INP5 Function 5 - SAI1_TX_SYNC Function 6 - CLKOUT_VDD2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_31	Function 0 - PIO0_31 Function 1 - LP_FLEXCOMM0_P0 Function 2 - UTICK0_CAP2 Function 3 - SCT0_OUT8 Function 4 - CTIMER4_MAT0 Alternate pin function - LP_FLEXCOMM0_P0 (UART boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_0	Function 0 - PIO1_0 Function 1 - LP_FLEXCOMM0_P1 Function 3 - SCT0_OUT9 Function 4 - CTIMER4_MAT1 Alternate pin function - LP_FLEXCOMM0_P1 (UART boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_2	Function 0 - PIO1_2 Function 1 - LP_FLEXCOMM0_P3 Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT6 Function 4 - CTIMER4_MAT3 Function 5 - LP_FLEXCOMM9_P5 Function 6 - CLKOUT_VDD1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO1_3	Function 0 - PIO1_3 Function 1 - LP_FLEXCOMM0_P4 Function 2 - SCT0_GPIN1 Function 3 - SCT0_OUT7 Function 4 - CTIMER_C_INP8 Function 5 - LP_FLEXCOMM9_P6 Function 6 - 32KHZ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_4	Function 0 - PIO1_4 Function 1 - LP_FLEXCOMM0_P5 Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT8 Function 4 - CTIMER1_MAT0 Function 5 - LP_FLEXCOMM9_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_5	Function 0 - PIO1_5 Function 1 - LP_FLEXCOMM0_P6 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT9 Function 4 - CTIMER1_MAT1 Function 5 - LP_FLEXCOMM9_P4	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_6	Function 0 - PIO1_6 Function 1 - LP_FLEXCOMM1_P4 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT0 Function 4 - CTIMER_C_INP12 Function 5 - I3C1_PUR Function 6 - 32KHZ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_7	Function 0 - PIO1_7 Function 1 - LP_FLEXCOMM1_P0 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT1 Function 4 - CTIMER_C_INP13 Function 5 - I3C1_SDA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - ISP boot select 0 • 11 - XSPI1 • 10 – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) • 01 - XSPI0 • 00 - SDHC0 (eMMC or SD set in OTP)	
PIO1_8	Function 0 - PIO1_8 Function 1 - LP_FLEXCOMM1_P1 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT2 Function 4 - CTIMER_C_INP14 Function 5 - I3C1_SCL Alternate pin function - ISP boot select 1 • 11 - XSPI1 • 10 – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) • 01 - XSPI0 • 00 - SDHC0 (eMMC or SD set in OTP)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_9	Function 0 - PIO1_9 Function 1 - LP_FLEXCOMM1_P2 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT3 Function 4 - CTIMER_C_INP15 Function 5 - CLKIN Function 6 - CLKOUT_VDD1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_10	Function 0 - PIO1_10 Function 1 - LP_FLEXCOMM1_P3 Function 2 - UTICK0_CAP3 Function 4 - CTIMER2_MAT1 Function 5 - TRACECLK Function 6 - CLKOUT_VDD1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_11	Function 0 - PIO1_11 Function 1 - LP_FLEXCOMM2_P0 Function 4 - CTIMER2_MAT2 Function 5 - TRACEDATA0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_12	Function 0 - PIO1_12 Function 1 - LP_FLEXCOMM2_P1 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT9 Function 4 - CTIMER1_MAT1 Function 5 - TRACEDATA1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_13	Function 0 - PIO1_13 Function 1 - LP_FLEXCOMM2_P4 Function 2 - UTICK0_CAP1 Function 4 - CTIMER2_MAT3 Function 5 - TRACEDATA2 Function 6 - 32KHZ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_14	Function 0 - PIO1_14 Function 1 - LP_FLEXCOMM2_P2 Function 4 - CTIMER3_MAT0 Function 5 - TRACEDATA3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_15	Function 0 - PIO1_15 Function 1 - LP_FLEXCOMM2_P3 Function 4 - CTIMER3_MAT1 Function 5 - CLKIN	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_16	Function 0 - PIO1_16 Function 1 - LP_FLEXCOMM3_P2 Function 4 - CTIMER3_MAT2 Function 5 - I3C0_PUR	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO1_17	Function 0 - PIO1_17 Function 1 - LP_FLEXCOMM3_P1 Function 2 - USB0_OVERCURRENTN Function 4 - CTIMER3_MAT3 Function 5 - I3C0_SCL Alternate pin function - LP_FLEXCOMM3_P1 (I2C boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_18	Function 0 - PIO1_18 Function 1 - LP_FLEXCOMM3_P0 Function 2 - USB0_PORTPWRN Function 4 - CTIMER4_MAT0 Function 5 - I3C0_SDA Alternate pin function - LP_FLEXCOMM3_P0 (I2C boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_19	Function 0 - PIO1_19 Function 1 - LP_FLEXCOMM3_P3 Function 2 - MCLK Function 3 - FREQME_GPIO_B_CLK Function 4 - CTIMER4_MAT1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_0	Function 0 - PIO2_0 Function 2 - USB0_OVERCURRENTN Function 4 - EZH_PIO0 Function 5 - FLEXIO_D0 Function 7 - LCD_DBI_CSX_AB	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_1	Function 0 - PIO2_1 Function 2 - USB0_PORTPWRN Function 4 - EZH_PIO1 Function 5 - FLEXIO_D1 Function 7 - LCD_DBI-DCX-AB	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_2	Function 0 - PIO2_2 Function 1 - LPSPI16_SOUT Function 4 - EZH_PIO2 Function 5 - FLEXIO_D2 Function 6 - LCD_ENABLE Function 7 - LCD_DBI-DATA_OEN	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_3	Function 0 - PIO2_3 Function 1 - LPSPI16_SCK Function 4 - EZH_PIO3 Function 5 - FLEXIO_D3 Function 6 - LCD_DOTCLK Function 7 - LCD_DBI-RWDX	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_4	Function 0 - PIO2_4 Function 1 - LPSPI16_SIN Function 4 - EZH_PIO4 Function 5 - FLEXIO_D4 Function 6 - LCD_HSYNC Function 7 - LCD_DBI-WRX	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_5	Function 0 - PIO2_5 Function 1 - LPSPI16_PCS0 Function 4 - EZH_PIO5	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 5 - FLEXIO_D5 Function 6 - LCD_VSYNC Function 7 - LCD_DBI-E	AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_6	Function 0 - PIO2_6 Function 1 - LPSPI16_PCS3 Function 4 - EZH_PIO6 Function 5 - FLEXIO_D6 Function 6 - LCD_D0 Function 7 - LCD_DBI-D0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_7	Function 0 - PIO2_7 Function 1 - LPSPI16_PCS2 Function 4 - EZH_PIO7 Function 5 - FLEXIO_D7 Function 6 - LCD_D1 Function 7 - LCD_DBI-D1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_8	Function 0 - PIO2_8 Function 1 - LPSPI16_PCS1 Function 4 - EZH_PIO8 Function 5 - FLEXIO_D8 Function 6 - LCD_D2 Function 7 - LCD_DBI-D2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO2_9	Function 0 - PIO2_9 Function 4 - EZH_PIO9 Function 5 - FLEXIO_D9 Function 6 - LCD_D3 Function 7 - LCD_DBI-D3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_10	Function 0 - PIO2_10 Function 4 - EZH_PIO10 Function 5 - FLEXIO_D10 Function 6 - LCD_D4 Function 7 - LCD_DBI-D4	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_11	Function 0 - PIO2_11 Function 4 - EZH_PIO11 Function 5 - FLEXIO_D11 Function 6 - LCD_D5 Function 7 - LCD_DBI-D5	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_12	Function 0 - PIO2_12 Function 4 - EZH_PIO12 Function 5 - FLEXIO_D12 Function 6 - LCD_D6 Function 7 - LCD_DBI-D6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_13	Function 0 - PIO2_13 Function 4 - EZH_PIO13 Function 5 - FLEXIO_D13 Function 6 - LCD_D7 Function 7 - LCD_DBI-D7	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_14	Function 0 - PIO2_14 Function 4 - EZH_PIO14 Function 5 - FLEXIO_D14 Function 6 - LOW_FREQ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_15	Function 0 - PIO2_15 Function 4 - EZH_PIO15 Function 5 - FLEXIO_D15 Function 6 - LOW_FREQ_CLKOUT_N	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_0	Function 0 - PIO3_0 Function 1 - LPSPI14_SOUT Function 4 - EZH_PIO16	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - LPSPI0_SOUT (SPI boot)	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_1	Function 0 - PIO3_1 Function 1 - LPSPI14_SCK Function 4 - EZH_PIO17 Alternate pin function - LPSPI0_SCK (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_2	Function 0 - PIO3_2 Function 1 - LPSPI14_SIN Function 4 - EZH_PIO18 Alternate pin function - LPSPI0_SIN (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_3	Function 0 - PIO3_3 Function 1 - LPSPI14_PCS0 Function 4 - EZH_PIO19 Alternate pin function - LPSPI0_PCS0 (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_4	Function 0 - PIO3_4 Function 1 - LPSPI14_PCS3 Function 2 - LP_FLEXCOMM0_P4 Function 3 - LP_FLEXCOMM3_P2 Function 4 - EZH_PIO20	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_5	Function 0 - PIO3_5 Function 1 - LPSPI14_PCS2 Function 2 - LP_FLEXCOMM0_P5 Function 3 - LP_FLEXCOMM3_P3 Function 4 - EZH_PIO21	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO3_6	Function 0 - PIO3_6 Function 1 - LPSPI14_PCS1 Function 2 - LP_FLEXCOMM0_P6 Function 3 - LP_FLEXCOMM9_P2 Function 4 - EZH_PIO22 Function 5 - I3C0_PUR	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_7	Function 0 - PIO3_7 Function 1 - LP_FLEXCOMM0_P2 Function 3 - LP_FLEXCOMM5_P5 Function 4 - EZH_PIO23	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_8	Function 0 - PIO3_8 Function 1 - LP_FLEXCOMM0_P3 Function 3 - LP_FLEXCOMM5_P6 Function 4 - EZH_PIO24	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_9	Function 0 - PIO3_9 Function 1 - LP_FLEXCOMM6_P2 Function 3 - LP_FLEXCOMM3_P4 Function 4 - EZH_PIO25	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_10	Function 0 - PIO3_10 Function 1 - LP_FLEXCOMM6_P3 Function 2 - USB0_OVERCURRENTN Function 3 - LP_FLEXCOMM3_P5 Function 4 - EZH_PIO26	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_11	Function 0 - PIO3_11	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM6_P4 Function 2 - USB0_PORTPWRN Function 3 - LP_FLEXCOMM9_P3 Function 4 - EZH_PIO27 Function 5 - I3C1_PUR	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_0	Function 0 - PIO4_0 Function 1 - XSPI2_SS0_N Function 2 - LP_FLEXCOMM8_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_1	Function 0 - PIO4_1 Function 1 - XSPI2_DATA0 Function 2 - LP_FLEXCOMM8_P1 Function 6 - LCD_D20	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_2	Function 0 - PIO4_2 Function 1 - XSPI2_DATA1 Function 2 - LP_FLEXCOMM8_P2 Function 6 - LCD_D21	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_3	Function 0 - PIO4_3 Function 1 - XSPI2_DATA2 Function 2 - LP_FLEXCOMM8_P3 Function 6 - LCD_D22	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_4	Function 0 - PIO4_4 Function 1 - XSPI2_DATA3 Function 2 - LP_FLEXCOMM8_P4	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 6 - LCD_D23	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_5	Function 0 - PIO4_5 Function 1 - XSPI2_DQS0 Function 2 - LP_FLEXCOMM8_P5	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_6	Function 0 - PIO4_6 Function 1 - XSPI2_DATA4 Function 2 - LP_FLEXCOMM8_P6 Function 6 - LCD_D16	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_7	Function 0 - PIO4_7 Function 1 - XSPI2_DATA5 Function 2 - LP_FLEXCOMM9_P0 Function 6 - LCD_D17	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_8	Function 0 - PIO4_8 Function 1 - XSPI2_DATA6 Function 2 - LP_FLEXCOMM9_P1 Function 6 - LCD_D18	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_9	Function 0 - PIO4_9 Function 1 - XSPI2_DATA7 Function 2 - LP_FLEXCOMM9_P2 Function 6 - LCD_D19	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO4_10	Function 0 - PIO4_10 Function 1 - XSPI2_SCLK0 Function 2 - LP_FLEXCOMM9_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_11	Function 0 - PIO4_11 Function 1 - XSPI2_SCLK0_N Function 2 - LP_FLEXCOMM9_P4 Function 3 - XSPI2_SS1_N	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO4_12	Function 0 - PIO4_12 Function 1 - XSPI2_DATA8 Function 2 - LP_FLEXCOMM9_P5 Function 6 - LCD_D8 Function 7 - LCD_DBI-D8	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_13	Function 0 - PIO4_13 Function 1 - XSPI2_DATA9 Function 2 - LP_FLEXCOMM9_P6 Function 6 - LCD_D9 Function 7 - LCD_DBI-D9	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_14	Function 0 - PIO4_14 Function 1 - XSPI2_DATA10 Function 2 - LP_FLEXCOMM10_P0 Function 6 - LCD_D10 Function 7 - LCD_DBI-D10	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_15	Function 0 - PIO4_15	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI2_DATA11 Function 2 - LP_FLEXCOMM10_P1 Function 6 - LCD_D11 Function 7 - LCD_DBI-D11	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_16	Function 0 - PIO4_16 Function 1 - XSPI2_DQS1 Function 2 - LP_FLEXCOMM10_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_17	Function 0 - PIO4_17 Function 1 - XSPI2_DATA12 Function 2 - LP_FLEXCOMM10_P3 Function 6 - LCD_D12 Function 7 - LCD_DBI-D12	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_18	Function 0 - PIO4_18 Function 1 - XSPI2_DATA13 Function 2 - LP_FLEXCOMM10_P4 Function 6 - LCD_D13 Function 7 - LCD_DBI-D13	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_19	Function 0 - PIO4_19 Function 1 - XSPI2_DATA14 Function 2 - LP_FLEXCOMM10_P5 Function 6 - LCD_D14 Function 7 - LCD_DBI-D14	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_20	Function 0 - PIO4_20 Function 1 - XSPI2_DATA15 Function 2 - LP_FLEXCOMM10_P6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 6 - LCD_D15 Function 7 - LCD_DBI-D15	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_0	Function 0 - PIO5_0 Function 1 - XSPI1_SS0_N Function 2 - LP_FLEXCOMM13_P4 Alternate pin function - XSPI1_SS0_N (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO5_1	Function 0 - PIO5_1 Function 1 - XSPI1_DATA0 Function 2 - LP_FLEXCOMM13_P5 Alternate pin function - XSPI1_DATA0 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_2	Function 0 - PIO5_2 Function 1 - XSPI1_DATA1 Function 2 - LP_FLEXCOMM13_P6 Alternate pin function - XSPI1_DATA1 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_3	Function 0 - PIO5_3 Function 1 - XSPI1_DATA2 Function 2 - LP_FLEXCOMM13_P0 Alternate pin function - XSPI1_DATA2 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_4	Function 0 - PIO5_4 Function 1 - XSPI1_DATA3 Function 2 - LP_FLEXCOMM13_P1 Alternate pin function - XSPI1_DATA3 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO5_5	Function 0 - PIO5_5 Function 1 - XSPI1_DQS0 Function 2 - LP_FLEXCOMM13_P2 Alternate pin function - XSPI1_DQS0 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_6	Function 0 - PIO5_6 Function 1 - XSPI1_DATA4 Function 2 - LP_FLEXCOMM12_P3 Alternate pin function - XSPI1_DATA4 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_7	Function 0 - PIO5_7 Function 1 - XSPI1_DATA5 Function 2 - LP_FLEXCOMM12_P0 Alternate pin function - XSPI1_DATA5 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_8	Function 0 - PIO5_8 Function 1 - XSPI1_DATA6 Function 2 - LP_FLEXCOMM12_P1 Alternate pin function - XSPI1_DATA6 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_9	Function 0 - PIO5_9 Function 1 - XSPI1_DATA7 Function 2 - LP_FLEXCOMM12_P2 Alternate pin function - XSPI1_DATA7 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_10	Function 0 - PIO5_10	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI1_SCLK0 Function 2 - LP_FLEXCOMM13_P3 Alternate pin function - XSPI1_SCLK0 (XSPI1-PSRAM boot)	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_11	Function 0 - PIO5_11 Function 1 - XSPI1_SCLK0_N Function 2 - LP_FLEXCOMM12_P4 Function 3 - XSPI1_SS1_N Alternate pin function - XSPI1_SCLK0_N (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO5_12	Function 0 - PIO5_12 Function 1 - XSPI1_DATA8 Function 2 - LP_FLEXCOMM12_P5 Alternate pin function - XSPI1_DATA8 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_13	Function 0 - PIO5_13 Function 1 - XSPI1_DATA9 Function 2 - LP_FLEXCOMM12_P6 Alternate pin function - XSPI1_DATA9 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_14	Function 0 - PIO5_14 Function 1 - XSPI1_DATA10 Function 2 - LP_FLEXCOMM11_P0 Alternate pin function - XSPI1_DATA10 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_15	Function 0 - PIO5_15 Function 1 - XSPI1_DATA11 Function 2 - LP_FLEXCOMM11_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - XSPI1_DATA11 (XSPI1-PSRAM boot)	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_16	Function 0 - PIO5_16 Function 1 - XSPI1_DQS1 Function 2 - LP_FLEXCOMM11_P2 Alternate pin function - XSPI1_DQS1 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_17	Function 0 - PIO5_17 Function 1 - XSPI1_DATA12 Function 2 - LP_FLEXCOMM11_P3 Alternate pin function - XSPI1_DATA12 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_18	Function 0 - PIO5_18 Function 1 - XSPI1_DATA13 Function 2 - LP_FLEXCOMM11_P4 Alternate pin function - XSPI1_DATA13 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_19	Function 0 - PIO5_19 Function 1 - XSPI1_DATA14 Function 2 - LP_FLEXCOMM11_P5 Alternate pin function - XSPI1_DATA14 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_20	Function 0 - PIO5_20 Function 1 - XSPI1_DATA15 Function 2 - LP_FLEXCOMM11_P6 Alternate pin function - XSPI1_DATA15 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO6_0	Function 0 - PIO6_0 Function 1 - XSPI0_SCLK0_N Function 2 - LP_FLEXCOMM7_P2 Alternate pin function - XSPI0_SCLK0_N (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_1	Function 0 - PIO6_1 Function 1 - XSPI0_SCLK0 Function 2 - LP_FLEXCOMM10_P0 Alternate pin function - XSPI0_SCLK0 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_2	Function 0 - PIO6_2 Function 1 - XSPI0_SS0_N Function 2 - LP_FLEXCOMM10_P1 Alternate pin function - XSPI0_SS0_N (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO6_3	Function 0 - PIO6_3 Function 1 - XSPI0_DATA0 Function 2 - LP_FLEXCOMM10_P2 Alternate pin function - XSPI0_DATA0 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_4	Function 0 - PIO6_4 Function 1 - XSPI0_DATA1 Function 2 - LP_FLEXCOMM10_P3 Alternate pin function - XSPI0_DATA1 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_5	Function 0 - PIO6_5	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI0_DATA2 Function 2 - LP_FLEXCOMM10_P4 Alternate pin function - XSPI0_DATA2 (XSPI0-NOR Flash boot)	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_6	Function 0 - PIO6_6 Function 1 - XSPI0_DATA3 Function 2 - LP_FLEXCOMM10_P5 Alternate pin function - XSPI0_DATA3 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_7	Function 0 - PIO6_7 Function 1 - XSPI0_DQS0 Function 2 - LP_FLEXCOMM10_P6 Alternate pin function - XSPI0_DQS0 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_8	Function 0 - PIO6_8 Function 1 - XSPI0_DATA4 Function 2 - LP_FLEXCOMM13_P0 Function 3 - LP_FLEXCOMM11_P2 Alternate pin function - XSPI0_DATA4 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_9	Function 0 - PIO6_9 Function 1 - XSPI0_DATA5 Function 2 - LP_FLEXCOMM13_P1 Function 3 - LP_FLEXCOMM11_P3 Alternate pin function - XSPI0_DATA5 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_10	Function 0 - PIO6_10 Function 1 - XSPI0_DATA6 Function 2 - LP_FLEXCOMM13_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 3 - LP_FLEXCOMM11_P0 Alternate pin function - XSPI0_DATA6 (XSPI0-NOR Flash boot)	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_11	Function 0 - PIO6_11 Function 1 - XSPI0_DATA7 Function 2 - LP_FLEXCOMM13_P3 Function 3 - LP_FLEXCOMM11_P1 Alternate pin function - XSPI0_DATA7 (XSPI0-NOR Flash boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO6_12	Function 0 - PIO6_12 Function 1 - XSPI0_SS1_N	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO7_0	Function 0 - PIO7_0 Function 1 - SDHC0_CLK Function 2 - LP_FLEXCOMM3_P4 Function 3 - LP_FLEXCOMM7_P6 Alternate pin function - SDHC0_CLK (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_1	Function 0 - PIO7_1 Function 1 - SDHC0_CMD Function 2 - LP_FLEXCOMM3_P5 Function 3 - LP_FLEXCOMM7_P5 Alternate pin function - SDHC0_CMD (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_2	Function 0 - PIO7_2 Function 1 - SDHC0_DATA4 Function 2 - LP_FLEXCOMM12_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO7_3	Function 0 - PIO7_3 Function 1 - SDHC0_DATA5 Function 2 - LP_FLEXCOMM12_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_4	Function 0 - PIO7_4 Function 1 - SDHC0_DATA6 Function 2 - LP_FLEXCOMM12_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_5	Function 0 - PIO7_5 Function 1 - SDHC0_DATA7 Function 2 - LP_FLEXCOMM12_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_6	Function 0 - PIO7_6 Function 1 - SDHC0_DATA0 Function 2 - LP_FLEXCOMM12_P4 Alternate pin function - SDHC0_DATA0 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_7	Function 0 - PIO7_7 Function 1 - SDHC0_DATA1 Function 2 - LP_FLEXCOMM12_P5 Alternate pin function - SDHC0_DATA1 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_8	Function 0 - PIO7_8	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - SDHC0_DATA2 Function 2 - LP_FLEXCOMM12_P6 Function 3 - LP_FLEXCOMM7_P3 Alternate pin function - SDHC0_DATA2 (SDHC0 boot)	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_9	Function 0 - PIO7_9 Function 1 - SDHC0_DATA3 Function 2 - LP_FLEXCOMM3_P6 Function 3 - LP_FLEXCOMM7_P4 Alternate pin function - SDHC0_DATA3 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_10	Function 0 - PIO7_10 Function 1 - SDHC0_WR_PRT Function 2 - LP_FLEXCOMM1_P0 Function 5 - SDHC0_DS	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_11	Function 0 - PIO7_11 Function 1 - SDHC0_CARD_DET_N Function 2 - LP_FLEXCOMM1_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_12	Function 0 - PIO7_12 Function 1 - SDHC0_RESET_N Function 2 - LP_FLEXCOMM1_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_13	Function 0 - PIO7_13 Function 1 - SDHC1_DATA4 Function 2 - LP_FLEXCOMM1_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_14	Function 0 - PIO7_14 Function 1 - SDHC1_DATA5 Function 2 - LP_FLEXCOMM7_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_15	Function 0 - PIO7_15 Function 1 - SDHC1_DATA6 Function 2 - LP_FLEXCOMM7_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_16	Function 0 - PIO7_16 Function 1 - SDHC1_DATA7 Function 2 - LP_FLEXCOMM7_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_17	Function 0 - PIO7_17 Function 1 - SDHC1_CLK Function 2 - LP_FLEXCOMM2_P0 Alternate pin function - SDHC1_CLK (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_18	Function 0 - PIO7_18 Function 1 - SDHC1_CMD Function 2 - LP_FLEXCOMM2_P1 Alternate pin function - SDHC1_CMD (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO7_19	Function 0 - PIO7_19 Function 1 - SDHC1_DATA0 Function 2 - LP_FLEXCOMM1_P4 Function 3 - LP_FLEXCOMM7_P3 Alternate pin function - SDHC1_DATA0 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_20	Function 0 - PIO7_20 Function 1 - SDHC1_DATA1 Function 2 - LP_FLEXCOMM1_P5 Function 3 - LP_FLEXCOMM2_P4 Alternate pin function - SDHC1_DATA1 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_21	Function 0 - PIO7_21 Function 1 - SDHC1_DATA2 Function 2 - LP_FLEXCOMM1_P6 Function 3 - LP_FLEXCOMM2_P5 Alternate pin function - SDHC1_DATA2 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_22	Function 0 - PIO7_22 Function 1 - SDHC1_DATA3 Function 2 - LP_FLEXCOMM2_P2 Alternate pin function - SDHC1_DATA3 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_23	Function 0 - PIO7_23 Function 1 - SDHC1_WR_PRT Function 2 - LP_FLEXCOMM2_P3 Function 5 - SDHC1_DS	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_24	Function 0 - PIO7_24	DRIVE - 100 Ω

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - SDHC1_CARD_DET_N Function 2 - LP_FLEXCOMM13_P4	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_25	Function 0 - PIO7_25 Function 1 - SDHC1_RESET_N Function 2 - LP_FLEXCOMM13_P5 Function 3 - LP_FLEXCOMM2_P6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_0	Function 0 - PIO8_0 Function 1 - LP_FLEXCOMM17_P5 Function 2 - CTIMER_S_INP0 Function 5 - JTAG_TCK/SWCLK	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-down PUPDENA - Enabled
PIO8_1	Function 0 - PIO8_1 Function 1 - LP_FLEXCOMM17_P6 Function 2 - CTIMER5_MAT0 Function 5 - JTAG_TMS/SWDIO	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_2	Function 0 - PIO8_2 Function 1 - LP_FLEXCOMM17_P2 Function 2 - CTIMER5_MAT1 Function 5 - JTAG_TDI	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_3	Function 0 - PIO8_3 Function 1 - LP_FLEXCOMM17_P3 Function 2 - CTIMER_S_INP1 Function 3 - UTICK1_CAP0 Function 5 - JTAG_TD0/SWO	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_4	Function 0 - PIO8_4 Function 1 - LP_FLEXCOMM17_P4 Function 2 - CTIMER5_MAT2 Function 3 - MCLK Function 5 - JTAG_TRSTN	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_5	Function 0 - PIO8_5 Function 1 - LP_FLEXCOMM17_P0 Function 2 - CTIMER5_MAT3 Function 3 - CTIMER_S_INP0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_6	Function 0 - PIO8_6 Function 1 - LP_FLEXCOMM17_P1 Function 2 - CTIMER6_MAT0 Function 4 - CMP0_OUT	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO8_7	Function 0 - PIO8_7 Function 1 - LP_FLEXCOMM18_P0 Function 2 - CTIMER6_MAT1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_8	Function 0 - PIO8_8 Function 1 - LP_FLEXCOMM18_P1 Function 2 - CTIMER_S_INP2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_9	Function 0 - PIO8_9 Function 1 - LP_FLEXCOMM18_P2 Function 2 - CTIMER6_MAT2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_10	Function 0 - PIO8_10 Function 1 - LP_FLEXCOMM18_P3 Function 2 - CTIMER_S_INP3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO8_11	Function 0 - PIO8_11 Function 1 - LP_FLEXCOMM18_P4 Function 2 - CTIMER6_MAT3 Function 3 - UTICK1_CAP1 Function 4 - LP_FLEXCOMM17_P4	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_12	Function 0 - PIO8_12 Function 1 - LP_FLEXCOMM18_P5 Function 2 - CTIMER7_MAT0 Function 3 - CTIMER_S_INP1 Function 4 - LP_FLEXCOMM17_P2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_13	Function 0 - PIO8_13 Function 1 - LP_FLEXCOMM18_P6 Function 2 - CTIMER7_MAT1 Function 3 - CTIMER_S_INP3 Function 4 - LP_FLEXCOMM17_P3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_14	Function 0 - PIO8_14 Function 1 - LP_FLEXCOMM19_P0 Function 2 - CTIMER7_MAT2 Function 3 - SAI3_TX_SYNC Function 4 - I3C2_SDA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_15	Function 0 - PIO8_15	IIENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM19_P1 Function 2 - CTIMER7_MAT3 Function 3 - SAI3_TX_BCLK Function 4 - I3C2_SCL	ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_16	Function 0 - PIO8_16 Function 1 - LP_FLEXCOMM19_P2 Function 2 - CTIMER_S_INP4 Function 3 - SAI3_TX_DATA Function 4 - I3C2_PUR	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_17	Function 0 - PIO8_17 Function 1 - LP_FLEXCOMM19_P3 Function 2 - CTIMER_S_INP5 Function 3 - SAI3_RX_SYNC	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_18	Function 0 - PIO8_18 Function 1 - LP_FLEXCOMM19_P4 Function 2 - CTIMER5_MAT0 Function 3 - CTIMER_S_INP4	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_19	Function 0 - PIO8_19 Function 1 - LP_FLEXCOMM19_P5	IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - CTIMER5_MAT1 Function 3 - CTIMER_S_INP6 Function 4 - CMP0_OUT	AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_20	Function 0 - PIO8_20 Function 1 - LP_FLEXCOMM19_P6 Function 2 - CTIMER_S_INP7 Function 3 - UTICK1_CAP2 Function 4 - I3C3_PUR	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_21	Function 0 - PIO8_21 Function 1 - LP_FLEXCOMM20_P3 Function 2 - CTIMER5_MAT0 Function 3 - LP_FLEXCOMM3_P0 Function 4 - I3C3_SDA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_22	Function 0 - PIO8_22 Function 1 - LP_FLEXCOMM20_P4 Function 2 - CTIMER5_MAT1 Function 3 - LP_FLEXCOMM3_P1 Function 4 - I3C3_SCL	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_23	Function 0 - PIO8_23 Function 1 - SAI3_RX_BCLK Function 2 - CTIMER5_MAT2	IIENA - Disabled ODENA - Disabled AMENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 3 - LP_FLEXCOMM19_P0	FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_24	Function 0 - PIO8_24 Function 1 - SAI3_RX_DATA Function 2 - CTIMER5_MAT3 Function 3 - LP_FLEXCOMM19_P1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_25	Function 0 - PIO8_25 Function 1 - SAI3_RX_SYNC Function 2 - CTIMER6_MAT0 Function 3 - LP_FLEXCOMM19_P2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_26	Function 0 - PIO8_26 Function 1 - SAI3_TX_BCLK Function 2 - CTIMER6_MAT1 Function 3 - LP_FLEXCOMM19_P3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_27	Function 0 - PIO8_27 Function 1 - SAI3_TX_DATA Function 2 - CTIMER6_MAT2 Function 3 - LP_FLEXCOMM19_P4	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_28	Function 0 - PIO8_28 Function 1 - SAI3_TX_SYNC Function 2 - CTIMER6_MAT3 Function 3 - LP_FLEXCOMM19_P5	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_30	Function 0 - PIO8_30 Function 1 - LP_FLEXCOMM20_P0 Function 2 - CTIMER_S_INP9 Function 3 - UTICK1_CAP3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_31	Function 0 - PIO8_31 Function 1 - LP_FLEXCOMM20_P1 Function 2 - CTIMER7_MAT0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_0	Function 0 - PIO10_0 Function 5 - PDM_CLK Alternate pin function - ADC0_A0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_1	Function 0 - PIO10_1 Function 5 - PDM_DATA01 Alternate pin function - ADC0_B0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_2	Function 0 - PIO10_2 Function 5 - PDM_DATA23 Alternate pin function - ADC0_A1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_3	Function 0 - PIO10_3 Function 5 - PDM_DATA45 Function - ADC0_B1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_4	Function 0 - PIO10_4 Function 5 - PDM_DATA67 Alternate pin function - ADC0_A2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_5	Function 0 - PIO10_5 Function 3 - CMP0_OUT Alternate pin function - ADC0_B2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_6	Function 0 - PIO10_6 Alternate pin function - SDADC_INP0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_7	Function 0 - PIO10_7 Alternate pin function - SDADC_INN0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_8	Function 0 - PIO10_8 Function 5 - PDM_CLK Alternate pin function - CMP0_INA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO10_9	Function 0 - PIO10_9 Function 5 - PDM_DATA01 Alternate pin function - CMP0_INB	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_12	Function 0 - PIO10_12 Function 5 - PDM_DATA67 Alternate pin function - ADC0_A4, SDADC_INP1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_13	Function 0 - PIO10_13 Function 5 - PDM_CLK Alternate pin function - ADC0_B4, SDADC_INN1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_14	Function 0 - PIO10_14 Function 5 - PDM_DATA01 Alternate pin function - ADC0_A5, SDADC_INP2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
PIO10_15	Function 0 - PIO10_15 Function 5 - PDM_DATA23 Alternate pin function - ADC0_B5, SDADC_INN2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_16	Function 0 - PIO10_16 Function 5 - PDM_DATA45 Alternate pin function - CMP0_INC, SDADC_INP3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_17	Function 0 - PIO10_17 Function 5 - PDM_DATA67 Alternate pin function - CMP0_IND, SDADC_INN3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
USB0_VBUS	Function 0 - USB0_VBUS Alternate pin function - USB1_VBUS	-
USB0_DM	Function 0 - USB0_DM Alternate pin function - USB1_DM	-
USB0_DP	Function 0 - USB0_DP Alternate pin function - USB1_DP	-
EUSB_DM	Function 0 - EUSB_DM Alternate pin function - EUSB_DM	-

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
EUSB_DP	Function 0 - EUSB_DP Alternate pin function - EUSB_DP	-
PMIC_MODE1	Function 0 - PMIC_MODE1 Alternate pin function - PMIC_MODE1	-
PMIC_MODE0	Function 0 - PMIC_MODE0 Alternate pin function - PMIC_MODE0	-
PMIC_I2C_SDA	Function 0 - PMIC_I2C_SDA Alternate pin function - PMIC_I2C_SDA	-
PMIC_I2C_SCL	Function 0 - PMIC_I2C_SCL Alternate pin function - PMIC_I2C_SCL	-
PMIC_IRQN	Function 0 - PMIC_IRQN Alternate pin function - PMIC_IRQN	-
XTALIN32KHZ	Function 0 - XTALIN32KHZ Alternate pin function - XTALIN	-
XTALOUT32KHZ	Function 0 - XTALOUT32KHZ Alternate pin function - XTALOUT	-
XTALINSYS	Function 0 - XTALINSYS Alternate pin function - RTCXIN	-
XTALOUTSYS	Function 0 - XTALOUTSYS Alternate pin function - RTCXOUT	-
RESETN	Function 0 - RESETN Alternate pin function - RESETN	-
MIPI_DSI_CLKP	Function 0 - MIPI_DSI_CLKP Alternate pin function - MIPI_DSI_CLKP	-
MIPI_DSI_CLKN	Function 0 - MIPI_DSI_CLKN Alternate pin function - MIPI_DSI_CLKN	-
MIPI_DSI_D0P	Function 0 - MIPI_DSI_D0P Alternate pin function - MIPI_DSI_D0P	-
MIPI_DSI_D0N	Function 0 - MIPI_DSI_D0N	-

Table continues on the next page...

Table 19. Pin Assignments (324FOWLP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - MIPI_DSI_D0N	
MIPI_DSI_D1P	Function 0 - MIPI_DSI_D1P Alternate pin function - MIPI_DSI_D1P	-
MIPI_DSI_D1N	Function 0 - MIPI_DSI_D1N Alternate pin function - MIPI_DSI_D1N	-

6.3.2 Pin assignments (256WLCSP)

Table 20. Pin assignments (256WLCSP)

Pin name	FSEL assignments	Pad settings
PIO0_0	Function 0 - PIO0_0 Function 1 - LP_FLEXCOMM8_P2 Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT0 Function 4 - CTIMER_C_INP9 Function 5 - SAI0_RX_BCLK Function 6 - SAI1_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_1	Function 0 - PIO0_1 Function 1 - LP_FLEXCOMM8_P3 Function 2 - SCT0_GPIN1 Function 3 - SCT0_OUT1 Function 4 - CTIMER0_MAT0 Function 5 - SAI0_RX_SYNC Function 6 - SAI1_RX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_2	Function 0 - PIO0_2 Function 1 - LP_FLEXCOMM8_P4 Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT2 Function 4 - CTIMER0_MAT1 Function 5 - SAI0_RX_DATA Function 7 - LP_FLEXCOMM9_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_3	Function 0 - PIO0_3 Function 1 - LP_FLEXCOMM4_P0	DRIVE - 100 Ω IIENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT3 Function 4 - CTIMER0_MAT2 Function 5 - SAI0_TX_BCLK	ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_4	Function 0 - PIO0_4 Function 1 - LP_FLEXCOMM4_P1 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER0_MAT3 Function 5 - SAI0_TX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_5	Function 0 - PIO0_5 Function 1 - LP_FLEXCOMM4_P4 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT5 Function 4 - CTIMER2_MAT0 Function 5 - SAI0_TX_SYNC Function 6 - MCLK Function 7 - LP_FLEXCOMM9_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_6	Function 0 - PIO0_6 Function 1 - LP_FLEXCOMM4_P2 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER_C_INP10 Function 5 - LP_FLEXCOMM8_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_7	Function 0 - PIO0_7 Function 1 - LP_FLEXCOMM4_P3 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT7 Function 4 - CTIMER_C_INP11 Function 5 - LP_FLEXCOMM8_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_10	Function 0 - PIO0_10 Function 1 - LP_FLEXCOMM5_P0	DRIVE - 100 Ω IIENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER1_MAT0 Function 5 - CTIMER_C_INP0 Function 6 - SAI2_RX_DATA	ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_11	Function 0 - PIO0_11 Function 1 - LP_FLEXCOMM5_P1 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT5 Function 4 - CTIMER1_MAT1 Function 5 - SAI2_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_12	Function 0 - PIO0_12 Function 1 - LP_FLEXCOMM5_P2 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER1_MAT2 Function 5 - SAI2_TX_DATA	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_13	Function 0 - PIO0_13 Function 1 - LP_FLEXCOMM5_P3 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT7 Function 4 - CTIMER1_MAT3 Function 5 - SAI2_TX_SYNC	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_14	Function 0 - PIO0_14 Function 1 - LP_FLEXCOMM5_P4 Function 2 - SCT0_GPIN0 Function 3 - SCT0_OUT8 Function 4 - CTIMER_C_INP2 Function 5 - SAI2_RX_DATA Function 6 - CLKOUT_VDD2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_19	Function 0 - PIO0_19 Function 1 - LP_FLEXCOMM6_P2	DRIVE - 100 Ω IIENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - SCT0_GPIN2 Function 3 - SCT0_OUT2 Function 4 - CTIMER2_MAT2 Function 5 - SAI1_RX_SYNC	ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_20	Function 0 - PIO0_20 Function 1 - LP_FLEXCOMM6_P3 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT3 Function 4 - CTIMER2_MAT3 Function 5 - SAI1_TX_BCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_21	Function 0 - PIO0_21 Function 1 - LP_FLEXCOMM6_P0 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT6 Function 4 - CTIMER_C_INP4 Function 5 - SAI1_TX_DATA Function 6 - MCLK	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_22	Function 0 - PIO0_22 Function 1 - LP_FLEXCOMM6_P1 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT4 Function 4 - CTIMER_C_INP5 Function 5 - SAI1_TX_SYNC Function 6 - CLKOUT_VDD2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO0_31	Function 0 - PIO0_31 Function 1 - LP_FLEXCOMM0_P0 Function 2 - UTICK0_CAP2 Function 3 - SCT0_OUT8 Function 4 - CTIMER4_MAT0 Alternate pin function - LP_FLEXCOMM0_P0 (UART boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_0	Function 0 - PIO1_0 Function 1 - LP_FLEXCOMM0_P1	DRIVE - 100 Ω IIENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 3 - SCT0_OUT9 Function 4 - CTIMER4_MAT1 Alternate pin function - LP_FLEXCOMM0_P1 (UART boot)	ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_6	Function 0 - PIO1_6 Function 1 - LP_FLEXCOMM1_P4 Function 2 - SCT0_GPIN4 Function 3 - SCT0_OUT0 Function 4 - CTIMER_C_INP12 Function 5 - I3C1_PUR Function 6 - 32KHZ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_7	Function 0 - PIO1_7 Function 1 - LP_FLEXCOMM1_P0 Function 2 - SCT0_GPIN5 Function 3 - SCT0_OUT1 Function 4 - CTIMER_C_INP13 Function 5 - I3C1_SDA Alternate pin function - ISP boot select 0 • 11b - XSPI1 • 10b – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) • 01b - XSPI0 • 00b - SDHC0 (eMMC or SD set in OTP)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_8	Function 0 - PIO1_8 Function 1 - LP_FLEXCOMM1_P1 Function 2 - SCT0_GPIN6 Function 3 - SCT0_OUT2 Function 4 - CTIMER_C_INP14 Function 5 - I3C1_SCL Alternate pin function - ISP boot select 1 • 11b - XSPI1 • 10b – Auto ISP (UART, SPI, I2C, USB-HID, eUSB) • 01b - XSPI0 • 00b - SDHC0 (eMMC or SD set in OTP)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
PIO1_9	Function 0 - PIO1_9 Function 1 - LP_FLEXCOMM1_P2 Function 2 - SCT0_GPIN7 Function 3 - SCT0_OUT3 Function 4 - CTIMER_C_INP15 Function 5 - CLKIN Function 6 - CLKOUT_VDD1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_10	Function 0 - PIO1_10 Function 1 - LP_FLEXCOMM1_P3 Function 2 - UTICK0_CAP3 Function 4 - CTIMER2_MAT1 Function 5 - TRACECLK Function 6 - CLKOUT_VDD1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_11	Function 0 - PIO1_11 Function 1 - LP_FLEXCOMM2_P0 Function 4 - CTIMER2_MAT2 Function 5 - TRACEDATA0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_12	Function 0 - PIO1_12 Function 1 - LP_FLEXCOMM2_P1 Function 2 - SCT0_GPIN3 Function 3 - SCT0_OUT9 Function 4 - CTIMER1_MAT1 Function 5 - TRACEDATA1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_13	Function 0 - PIO1_13 Function 1 - LP_FLEXCOMM2_P4 Function 2 - UTICK0_CAP1 Function 4 - CTIMER2_MAT3 Function 5 - TRACEDATA2 Function 6 - 32KHZ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_14	Function 0 - PIO1_14	DRIVE - 100 Ω

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM2_P2 Function 4 - CTIMER3_MAT0 Function 5 - TRACEDATA3	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_15	Function 0 - PIO1_15 Function 1 - LP_FLEXCOMM2_P3 Function 4 - CTIMER3_MAT1 Function 5 - CLKIN	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_16	Function 0 - PIO1_16 Function 1 - LP_FLEXCOMM3_P2 Function 4 - CTIMER3_MAT2 Function 5 - I3C0_PUR	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_17	Function 0 - PIO1_17 Function 1 - LP_FLEXCOMM3_P1 Function 2 - USB0_OVERCURRENTN Function 4 - CTIMER3_MAT3 Function 5 - I3C0_SCL Alternate pin function - LP_FLEXCOMM3_P1 (I2C boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO1_18	Function 0 - PIO1_18 Function 1 - LP_FLEXCOMM3_P0 Function 2 - USB0_PORTPWRN Function 4 - CTIMER4_MAT0 Function 5 - I3C0_SDA Alternate pin function - LP_FLEXCOMM3_P0 (I2C boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_0	Function 0 - PIO2_0 Function 2 - USB0_OVERCURRENTN Function 4 - EZH_PIO0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 5 - FLEXIO_D0 Function 7 - LCD_DBI_CSX_AB	AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_1	Function 0 - PIO2_1 Function 2 - USB0_PORTPWRN Function 4 - EZH_PIO1 Function 5 - FLEXIO_D1 Function 7 - LCD_DBI-DCX-AB	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_2	Function 0 - PIO2_2 Function 1 - LPSPI16_SOUT Function 4 - EZH_PIO2 Function 5 - FLEXIO_D2 Function 6 - LCD_ENABLE Function 7 - LCD_DBI-DATA_OEN	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_3	Function 0 - PIO2_3 Function 1 - LPSPI16_SCK Function 4 - EZH_PIO3 Function 5 - FLEXIO_D3 Function 6 - LCD_DOTCLK Function 7 - LCD_DBI-RWDX	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
PIO2_4	Function 0 - PIO2_4 Function 1 - LPSPI16_SIN Function 4 - EZH_PIO4 Function 5 - FLEXIO_D4 Function 6 - LCD_HSYNC Function 7 - LCD_DBI-WRX	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_5	Function 0 - PIO2_5 Function 1 - LPSPI16_PCS0 Function 4 - EZH_PIO5 Function 5 - FLEXIO_D5 Function 6 - LCD_VSYNC Function 7 - LCD_DBI-E	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_6	Function 0 - PIO2_6 Function 1 - LPSPI16_PCS3 Function 4 - EZH_PIO6 Function 5 - FLEXIO_D6 Function 6 - LCD_D0 Function 7 - LCD_DBI-D0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_7	Function 0 - PIO2_7 Function 1 - LPSPI16_PCS2 Function 4 - EZH_PIO7 Function 5 - FLEXIO_D7 Function 6 - LCD_D1 Function 7 - LCD_DBI-D1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_8	Function 0 - PIO2_8 Function 1 - LPSPI16_PCS1 Function 4 - EZH_PIO8 Function 5 - FLEXIO_D8 Function 6 - LCD_D2 Function 7 - LCD_DBI-D2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_9	Function 0 - PIO2_9 Function 4 - EZH_PIO9 Function 5 - FLEXIO_D9 Function 6 - LCD_D3 Function 7 - LCD_DBI-D3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_10	Function 0 - PIO2_10 Function 4 - EZH_PIO10 Function 5 - FLEXIO_D10 Function 6 - LCD_D4 Function 7 - LCD_DBI-D4	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_11	Function 0 - PIO2_11 Function 4 - EZH_PIO11 Function 5 - FLEXIO_D11	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 6 - LCD_D5 Function 7 - LCD_DBI-D5	AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_12	Function 0 - PIO2_12 Function 4 - EZH_PIO12 Function 5 - FLEXIO_D12 Function 6 - LCD_D6 Function 7 - LCD_DBI-D6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_13	Function 0 - PIO2_13 Function 4 - EZH_PIO13 Function 5 - FLEXIO_D13 Function 6 - LCD_D7 Function 7 - LCD_DBI-D7	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO2_14	Function 0 - PIO2_14 Function 4 - EZH_PIO14 Function 5 - FLEXIO_D14 Function 6 - LOW_FREQ_CLKOUT	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
PIO2_15	Function 0 - PIO2_15 Function 4 - EZH_PIO15 Function 5 - FLEXIO_D15 Function 6 - LOW_FREQ_CLKOUT_N	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_0	Function 0 - PIO3_0 Function 1 - LPSPI14_SOUT Function 4 - EZH_PIO16 Alternate pin function - LPSPI0_SOUT (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_1	Function 0 - PIO3_1 Function 1 - LPSPI14_SCK Function 4 - EZH_PIO17 Alternate pin function - LPSPI0_SCK (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_2	Function 0 - PIO3_2 Function 1 - LPSPI14_SIN Function 4 - EZH_PIO18 Alternate pin function - LPSPI0_SIN (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO3_3	Function 0 - PIO3_3 Function 1 - LPSPI14_PCS0 Function 4 - EZH_PIO19 Alternate pin function - LPSPI0_PCS0 (SPI boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO4_0	Function 0 - PIO4_0 Function 1 - XSPI2_SS0_N Function 2 - LP_FLEXCOMM8_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_1	Function 0 - PIO4_1 Function 1 - XSPI2_DATA0 Function 2 - LP_FLEXCOMM8_P1 Function 6 - LCD_D20	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_2	Function 0 - PIO4_2 Function 1 - XSPI2_DATA1 Function 2 - LP_FLEXCOMM8_P2 Function 6 - LCD_D21	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_3	Function 0 - PIO4_3 Function 1 - XSPI2_DATA2 Function 2 - LP_FLEXCOMM8_P3 Function 6 - LCD_D22	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_4	Function 0 - PIO4_4 Function 1 - XSPI2_DATA3 Function 2 - LP_FLEXCOMM8_P4 Function 6 - LCD_D23	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_5	Function 0 - PIO4_5	DRIVE - 100 Ω

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI2_DQS0 Function 2 - LP_FLEXCOMM8_P5	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_6	Function 0 - PIO4_6 Function 1 - XSPI2_DATA4 Function 2 - LP_FLEXCOMM8_P6 Function 6 - LCD_D16	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_7	Function 0 - PIO4_7 Function 1 - XSPI2_DATA5 Function 2 - LP_FLEXCOMM9_P0 Function 6 - LCD_D17	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_8	Function 0 - PIO4_8 Function 1 - XSPI2_DATA6 Function 2 - LP_FLEXCOMM9_P1 Function 6 - LCD_D18	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_9	Function 0 - PIO4_9 Function 1 - XSPI2_DATA7 Function 2 - LP_FLEXCOMM9_P2 Function 6 - LCD_D19	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_10	Function 0 - PIO4_10 Function 1 - XSPI2_SCLK0 Function 2 - LP_FLEXCOMM9_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_11	Function 0 - PIO4_11 Function 1 - XSPI2_SCLK0_N Function 2 - LP_FLEXCOMM9_P4 Function 3 - XSPI2_SS1_N	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO4_12	Function 0 - PIO4_12 Function 1 - XSPI2_DATA8 Function 2 - LP_FLEXCOMM9_P5 Function 6 - LCD_D8 Function 7 - LCD_DBI-D8	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_13	Function 0 - PIO4_13 Function 1 - XSPI2_DATA9 Function 2 - LP_FLEXCOMM9_P6 Function 6 - LCD_D9 Function 7 - LCD_DBI-D9	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_14	Function 0 - PIO4_14 Function 1 - XSPI2_DATA10 Function 2 - LP_FLEXCOMM10_P0 Function 6 - LCD_D10 Function 7 - LCD_DBI-D10	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_15	Function 0 - PIO4_15 Function 1 - XSPI2_DATA11 Function 2 - LP_FLEXCOMM10_P1 Function 6 - LCD_D11 Function 7 - LCD_DBI-D11	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO4_16	Function 0 - PIO4_16 Function 1 - XSPI2_DQS1 Function 2 - LP_FLEXCOMM10_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_17	Function 0 - PIO4_17 Function 1 - XSPI2_DATA12 Function 2 - LP_FLEXCOMM10_P3 Function 6 - LCD_D12 Function 7 - LCD_DBI-D12	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_18	Function 0 - PIO4_18 Function 1 - XSPI2_DATA13 Function 2 - LP_FLEXCOMM10_P4 Function 6 - LCD_D13 Function 7 - LCD_DBI-D13	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_19	Function 0 - PIO4_19 Function 1 - XSPI2_DATA14 Function 2 - LP_FLEXCOMM10_P5 Function 6 - LCD_D14 Function 7 - LCD_DBI-D14	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO4_20	Function 0 - PIO4_20 Function 1 - XSPI2_DATA15 Function 2 - LP_FLEXCOMM10_P6 Function 6 - LCD_D15 Function 7 - LCD_DBI-D15	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_0	Function 0 - PIO5_0	DRIVE - 100 Ω

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI1_SS0_N Function 2 - LP_FLEXCOMM13_P4 Alternate pin function - XSPI1_SS0_N (XSPI1-PSRAM boot)	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO5_1	Function 0 - PIO5_1 Function 1 - XSPI1_DATA0 Function 2 - LP_FLEXCOMM13_P5 Alternate pin function - XSPI1_DATA0 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_2	Function 0 - PIO5_2 Function 1 - XSPI1_DATA1 Function 2 - LP_FLEXCOMM13_P6 Alternate pin function - XSPI1_DATA1 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_3	Function 0 - PIO5_3 Function 1 - XSPI1_DATA2 Function 2 - LP_FLEXCOMM13_P0 Alternate pin function - XSPI1_DATA2 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_4	Function 0 - PIO5_4 Function 1 - XSPI1_DATA3 Function 2 - LP_FLEXCOMM13_P1 Alternate pin function - XSPI1_DATA3 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_5	Function 0 - PIO5_5 Function 1 - XSPI1_DQS0 Function 2 - LP_FLEXCOMM13_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - XSPI1_DQS0 (XSPI1-PSRAM boot)	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_6	Function 0 - PIO5_6 Function 1 - XSPI1_DATA4 Function 2 - LP_FLEXCOMM12_P3 Alternate pin function - XSPI1_DATA4 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_7	Function 0 - PIO5_7 Function 1 - XSPI1_DATA5 Function 2 - LP_FLEXCOMM12_P0 Alternate pin function - XSPI1_DATA5 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_8	Function 0 - PIO5_8 Function 1 - XSPI1_DATA6 Function 2 - LP_FLEXCOMM12_P1 Alternate pin function - XSPI1_DATA6 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_9	Function 0 - PIO5_9 Function 1 - XSPI1_DATA7 Function 2 - LP_FLEXCOMM12_P2 Alternate pin function - XSPI1_DATA7 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_10	Function 0 - PIO5_10 Function 1 - XSPI1_SCLK0 Function 2 - LP_FLEXCOMM13_P3 Alternate pin function - XSPI1_SCLK0 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO5_11	Function 0 - PIO5_11 Function 1 - XSPI1_SCLK0_N Function 2 - LP_FLEXCOMM12_P4 Function 3 - XSPI1_SS1_N Alternate pin function - XSPI1_SCLK0_N (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO5_12	Function 0 - PIO5_12 Function 1 - XSPI1_DATA8 Function 2 - LP_FLEXCOMM12_P5 Alternate pin function - XSPI1_DATA8 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_13	Function 0 - PIO5_13 Function 1 - XSPI1_DATA9 Function 2 - LP_FLEXCOMM12_P6 Alternate pin function - XSPI1_DATA9 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_14	Function 0 - PIO5_14 Function 1 - XSPI1_DATA10 Function 2 - LP_FLEXCOMM11_P0 Alternate pin function - XSPI1_DATA10 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_15	Function 0 - PIO5_15 Function 1 - XSPI1_DATA11 Function 2 - LP_FLEXCOMM11_P1 Alternate pin function - XSPI1_DATA11 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_16	Function 0 - PIO5_16	DRIVE - 100 Ω

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - XSPI1_DQS1 Function 2 - LP_FLEXCOMM11_P2 Alternate pin function - XSPI1_DQS1 (XSPI1-PSRAM boot)	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_17	Function 0 - PIO5_17 Function 1 - XSPI1_DATA12 Function 2 - LP_FLEXCOMM11_P3 Alternate pin function - XSPI1_DATA12 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_18	Function 0 - PIO5_18 Function 1 - XSPI1_DATA13 Function 2 - LP_FLEXCOMM11_P4 Alternate pin function - XSPI1_DATA13 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_19	Function 0 - PIO5_19 Function 1 - XSPI1_DATA14 Function 2 - LP_FLEXCOMM11_P5 Alternate pin function - XSPI1_DATA14 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO5_20	Function 0 - PIO5_20 Function 1 - XSPI1_DATA15 Function 2 - LP_FLEXCOMM11_P6 Alternate pin function - XSPI1_DATA15 (XSPI1-PSRAM boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_0	Function 0 - PIO7_0 Function 1 - SDHC0_CLK Function 2 - LP_FLEXCOMM3_P4	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 3 - LP_FLEXCOMM7_P6 Alternate pin function - SDHC0_CLK (SDHC0 boot)	IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_1	Function 0 - PIO7_1 Function 1 - SDHC0_CMD Function 2 - LP_FLEXCOMM3_P5 Function 3 - LP_FLEXCOMM7_P5 Alternate pin function - SDHC0_CMD (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_2	Function 0 - PIO7_2 Function 1 - SDHC0_DATA4 Function 2 - LP_FLEXCOMM12_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_3	Function 0 - PIO7_3 Function 1 - SDHC0_DATA5 Function 2 - LP_FLEXCOMM12_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_4	Function 0 - PIO7_4 Function 1 - SDHC0_DATA6 Function 2 - LP_FLEXCOMM12_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_5	Function 0 - PIO7_5 Function 1 - SDHC0_DATA7 Function 2 - LP_FLEXCOMM12_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO7_6	Function 0 - PIO7_6 Function 1 - SDHC0_DATA0 Function 2 - LP_FLEXCOMM12_P4 Alternate pin function - SDHC0_DATA0 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_7	Function 0 - PIO7_7 Function 1 - SDHC0_DATA1 Function 2 - LP_FLEXCOMM12_P5 Alternate pin function - SDHC0_DATA1 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_8	Function 0 - PIO7_8 Function 1 - SDHC0_DATA2 Function 2 - LP_FLEXCOMM12_P6 Function 3 - LP_FLEXCOMM7_P3 Alternate pin function - SDHC0_DATA2 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_9	Function 0 - PIO7_9 Function 1 - SDHC0_DATA3 Function 2 - LP_FLEXCOMM3_P6 Function 3 - LP_FLEXCOMM7_P4 Alternate pin function - SDHC0_DATA3 (SDHC0 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_10	Function 0 - PIO7_10 Function 1 - SDHC0_WR_PRT Function 2 - LP_FLEXCOMM1_P0 Function 5 - SDHC0_DS	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_11	Function 0 - PIO7_11	DRIVE - 100 Ω

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - SDHC0_CARD_DET_N Function 2 - LP_FLEXCOMM1_P1	IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_12	Function 0 - PIO7_12 Function 1 - SDHC0_RESET_N Function 2 - LP_FLEXCOMM1_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_13	Function 0 - PIO7_13 Function 1 - SDHC1_DATA4 Function 2 - LP_FLEXCOMM1_P3	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_14	Function 0 - PIO7_14 Function 1 - SDHC1_DATA5 Function 2 - LP_FLEXCOMM7_P0	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_15	Function 0 - PIO7_15 Function 1 - SDHC1_DATA6 Function 2 - LP_FLEXCOMM7_P1	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_16	Function 0 - PIO7_16 Function 1 - SDHC1_DATA7 Function 2 - LP_FLEXCOMM7_P2	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_17	Function 0 - PIO7_17 Function 1 - SDHC1_CLK Function 2 - LP_FLEXCOMM2_P0 Alternate pin function - SDHC1_CLK (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_18	Function 0 - PIO7_18 Function 1 - SDHC1_CMD Function 2 - LP_FLEXCOMM2_P1 Alternate pin function - SDHC1_CMD (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_19	Function 0 - PIO7_19 Function 1 - SDHC1_DATA0 Function 2 - LP_FLEXCOMM1_P4 Function 3 - LP_FLEXCOMM7_P3 Alternate pin function - SDHC1_DATA0 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_20	Function 0 - PIO7_20 Function 1 - SDHC1_DATA1 Function 2 - LP_FLEXCOMM1_P5 Function 3 - LP_FLEXCOMM2_P4 Alternate pin function - SDHC1_DATA1 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_21	Function 0 - PIO7_21 Function 1 - SDHC1_DATA2 Function 2 - LP_FLEXCOMM1_P6 Function 3 - LP_FLEXCOMM2_P5 Alternate pin function - SDHC1_DATA2 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO7_22	Function 0 - PIO7_22 Function 1 - SDHC1_DATA3 Function 2 - LP_FLEXCOMM2_P2 Alternate pin function - SDHC1_DATA3 (SDHC1 boot)	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_23	Function 0 - PIO7_23 Function 1 - SDHC1_WR_PRT Function 2 - LP_FLEXCOMM2_P3 Function 5 - SDHC1_DS	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO7_25	Function 0 - PIO7_25 Function 1 - SDHC1_RESET_N Function 2 - LP_FLEXCOMM13_P5 Function 3 - LP_FLEXCOMM2_P6	DRIVE - 100 Ω IIENA - Disabled ODENA - Disabled IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_0	Function 0 - PIO8_0 Function 1 - LP_FLEXCOMM17_P5 Function 2 - CTIMER_S_INP0 Function 5 - JTAG_TCK/SWCLK	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-down PUPDENA - Enabled
PIO8_1	Function 0 - PIO8_1 Function 1 - LP_FLEXCOMM17_P6 Function 2 - CTIMER5_MAT0 Function 5 - JTAG_TMS/SWDIO	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Enabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_2	Function 0 - PIO8_2 Function 1 - LP_FLEXCOMM17_P2 Function 2 - CTIMER5_MAT1 Function 5 - JTAG_TDI	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_3	Function 0 - PIO8_3 Function 1 - LP_FLEXCOMM17_P3 Function 2 - CTIMER_S_INP1 Function 3 - UTICK1_CAP0 Function 5 - JTAG_TD0/SWO	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_4	Function 0 - PIO8_4 Function 1 - LP_FLEXCOMM17_P4 Function 2 - CTIMER5_MAT2 Function 3 - MCLK Function 5 - JTAG_TRSTN	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Enabled PUPDSEL - Pull-up PUPDENA - Enabled
PIO8_5	Function 0 - PIO8_5 Function 1 - LP_FLEXCOMM17_P0 Function 2 - CTIMER5_MAT3 Function 3 - CTIMER_S_INP0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_6	Function 0 - PIO8_6 Function 1 - LP_FLEXCOMM17_P1 Function 2 - CTIMER6_MAT0 Function 4 - CMP0_OUT	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_7	Function 0 - PIO8_7 Function 1 - LP_FLEXCOMM18_P0 Function 2 - CTIMER6_MAT1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_8	Function 0 - PIO8_8 Function 1 - LP_FLEXCOMM18_P1 Function 2 - CTIMER_S_INP2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_9	Function 0 - PIO8_9 Function 1 - LP_FLEXCOMM18_P2 Function 2 - CTIMER6_MAT2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDENA - Disabled
PIO8_10	Function 0 - PIO8_10 Function 1 - LP_FLEXCOMM18_P3 Function 2 - CTIMER_S_INP3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_11	Function 0 - PIO8_11 Function 1 - LP_FLEXCOMM18_P4 Function 2 - CTIMER6_MAT3 Function 3 - UTICK1_CAP1 Function 4 - LP_FLEXCOMM17_P4	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_12	Function 0 - PIO8_12 Function 1 - LP_FLEXCOMM18_P5 Function 2 - CTIMER7_MAT0 Function 3 - CTIMER_S_INP1 Function 4 - LP_FLEXCOMM17_P2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_13	Function 0 - PIO8_13 Function 1 - LP_FLEXCOMM18_P6 Function 2 - CTIMER7_MAT1 Function 3 - CTIMER_S_INP3 Function 4 - LP_FLEXCOMM17_P3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
PIO8_14	Function 0 - PIO8_14 Function 1 - LP_FLEXCOMM19_P0 Function 2 - CTIMER7_MAT2 Function 3 - SAI3_TX_SYNC Function 4 - I3C2_SDA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_15	Function 0 - PIO8_15 Function 1 - LP_FLEXCOMM19_P1 Function 2 - CTIMER7_MAT3 Function 3 - SAI3_TX_BCLK Function 4 - I3C2_SCL	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_16	Function 0 - PIO8_16 Function 1 - LP_FLEXCOMM19_P2 Function 2 - CTIMER_S_INP4 Function 3 - SAI3_TX_DATA Function 4 - I3C2_PUR	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_17	Function 0 - PIO8_17 Function 1 - LP_FLEXCOMM19_P3 Function 2 - CTIMER_S_INP5 Function 3 - SAI3_RX_SYNC	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_18	Function 0 - PIO8_18	IIENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 1 - LP_FLEXCOMM19_P4 Function 2 - CTIMER5_MAT0 Function 3 - CTIMER_S_INP4	ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_19	Function 0 - PIO8_19 Function 1 - LP_FLEXCOMM19_P5 Function 2 - CTIMER5_MAT1 Function 3 - CTIMER_S_INP6 Function 4 - CMP0_OUT	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_20	Function 0 - PIO8_20 Function 1 - LP_FLEXCOMM19_P6 Function 2 - CTIMER_S_INP7 Function 3 - UTICK1_CAP2 Function 4 - I3C3_PUR	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_21	Function 0 - PIO8_21 Function 1 - LP_FLEXCOMM20_P3 Function 2 - CTIMER5_MAT0 Function 3 - LP_FLEXCOMM3_P0 Function 4 - I3C3_SDA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_22	Function 0 - PIO8_22 Function 1 - LP_FLEXCOMM20_P4	IIENA - Disabled ODENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 2 - CTIMER5_MAT1 Function 3 - LP_FLEXCOMM3_P1 Function 4 - I3C3_SCL	AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_23	Function 0 - PIO8_23 Function 1 - SAI3_RX_BCLK Function 2 - CTIMER5_MAT2 Function 3 - LP_FLEXCOMM19_P0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_24	Function 0 - PIO8_24 Function 1 - SAI3_RX_DATA Function 2 - CTIMER5_MAT3 Function 3 - LP_FLEXCOMM19_P1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_25	Function 0 - PIO8_25 Function 1 - SAI3_RX_SYNC Function 2 - CTIMER6_MAT0 Function 3 - LP_FLEXCOMM19_P2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_26	Function 0 - PIO8_26 Function 1 - SAI3_TX_BCLK Function 2 - CTIMER6_MAT1	IIENA - Disabled ODENA - Disabled AMENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Function 3 - LP_FLEXCOMM19_P3	FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_30	Function 0 - PIO8_30 Function 1 - LP_FLEXCOMM20_P0 Function 2 - CTIMER_S_INP9 Function 3 - UTICK1_CAP3	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO8_31	Function 0 - PIO8_31 Function 1 - LP_FLEXCOMM20_P1 Function 2 - CTIMER7_MAT0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_0	Function 0 - PIO10_0 Function 5 - PDM_CLK Alternate pin function - ADC0_A0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_1	Function 0 - PIO10_1 Function 5 - PDM_DATA01 Alternate pin function - ADC0_B0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_2	Function 0 - PIO10_2 Function 5 - PDM_DATA23 Alternate pin function - ADC0_A1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_3	Function 0 - PIO10_3 Function 5 - PDM_DATA45 Alternate pin function - ADC0_B1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_4	Function 0 - PIO10_4 Function 5 - PDM_DATA67 Alternate pin function - ADC0_A2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_5	Function 0 - PIO10_5 Function 3 - CMP0_OUT Alternate pin function - ADC0_B2	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode)

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_6	Function 0 - PIO10_6 Alternate pin function - SDADC_INP0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_7	Function 0 - PIO10_7 Alternate pin function - SDADC_INN0	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_8	Function 0 - PIO10_8 Function 5 - PDM_CLK Alternate pin function - CMP0_INA	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_9	Function 0 - PIO10_9 Function 5 - PDM_DATA01 Alternate pin function - CMP0_INB	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
		PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_12	Function 0 - PIO10_12 Function 5 - PDM_DATA67 Alternate pin function - ADC0_A4, SDADC_INP1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
PIO10_13	Function 0 - PIO10_13 Function 5 - PDM_CLK Alternate pin function - ADC0_B4, SDADC_INN1	IIENA - Disabled ODENA - Disabled AMENA - Disabled FULLDRIVE - Normal output drive SLEWRATE - Disabled (standard mode) IBENA - Disabled PUPDSEL - Pull-down PUPDENA - Disabled
USB0_VBUS	Function 0 - USB0_VBUS Alternate pin function - USB1_VBUS	-
USB0_DM	Function 0 - USB0_DM Alternate pin function - USB1_DM	-
USB0_DP	Function 0 - USB0_DP Alternate pin function - USB1_DP	-
PMIC_MODE1	Function 0 - PMIC_MODE1 Alternate pin function - PMIC_MODE1	-
PMIC_MODE0	Function 0 - PMIC_MODE0 Alternate pin function - PMIC_MODE0	-
PMIC_I2C_SDA	Function 0 - PMIC_I2C_SDA Alternate pin function - PMIC_I2C_SDA	-
PMIC_I2C_SCL	Function 0 - PMIC_I2C_SCL	-

Table continues on the next page...

Table 20. Pin assignments (256WLCSP) (continued)

Pin name	FSEL assignments	Pad settings
	Alternate pin function - PMIC_I2C_SCL	
PMIC_IRQN	Function 0 - PMIC_IRQN Alternate pin function - PMIC_IRQN	-
XTALIN32KHZ	Function 0 - XTALIN32KHZ Alternate pin function - XTALIN	-
XTALOUT32KHZ	Function 0 - XTALOUT32KHZ Alternate pin function - XTALOUT	-
XTALINSYS	Function 0 - XTALINSYS Alternate pin function - RTCXIN	-
XTALOUTSYS	Function 0 - XTALOUTSYS Alternate pin function - RTCXOUT	-
RESETN	Function 0 - RESETN Alternate pin function - RESETN	-
MIPI_DSI_CLKP	Function 0 - MIPI_DSI_CLKP Alternate pin function - MIPI_DSI_CLKP	-
MIPI_DSI_CLKN	Function 0 - MIPI_DSI_CLKN Alternate pin function - MIPI_DSI_CLKN	-
MIPI_DSI_D0P	Function 0 - MIPI_DSI_D0P Alternate pin function - MIPI_DSI_D0P	-
MIPI_DSI_D0N	Function 0 - MIPI_DSI_D0N Alternate pin function - MIPI_DSI_D0N	-
MIPI_DSI_D1P	Function 0 - MIPI_DSI_D1P Alternate pin function - MIPI_DSI_D1P	-
MIPI_DSI_D1N	Function 0 - MIPI_DSI_D1N Alternate pin function - MIPI_DSI_D1N	-

Chapter 7
Cortex-M33 Platform Overview

7.1 Overview

There are two CM33 cores modules in this chip. CPU0 is the control processor in the Compute domain. CPU1 is the control processor in the Sense domain.

Each Cortex-M33 implements a modified Harvard memory architecture using two 32-bit bus interfaces: the code and system buses. The bus interfaces are activated by address range and can include both instruction fetches and operand data references on a given bus port. The code bus is typically used for instruction fetching and data accesses of PC-relative data, while the system bus is typically used for operand data references to the on- and off-chip memories and peripheral accesses. The bus structure fully supports concurrent instruction fetch and data accesses, but the Cortex-M33 implementations can generate both types of references on each bus.

7.1.1 CPU0 and CPU1 block diagrams

Block diagrams are shown below:

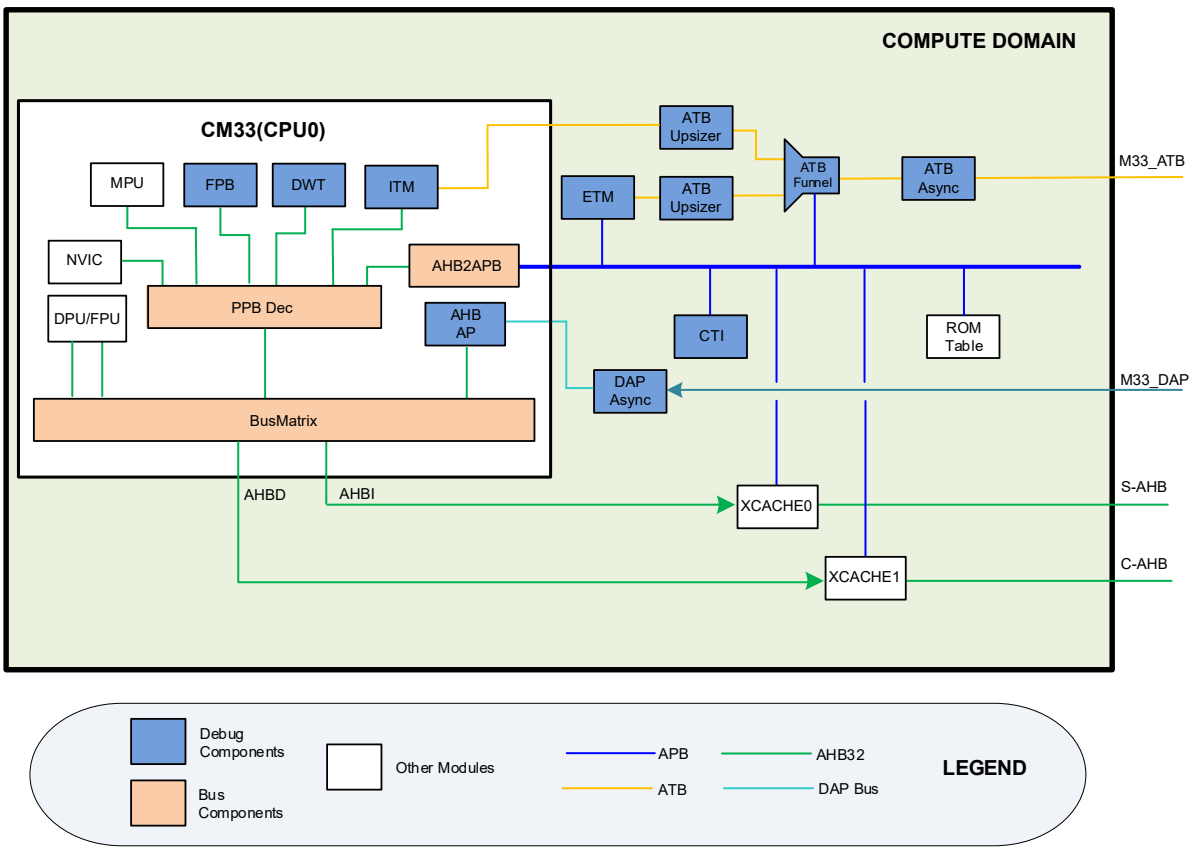


Figure 9. CPU0 platform diagram

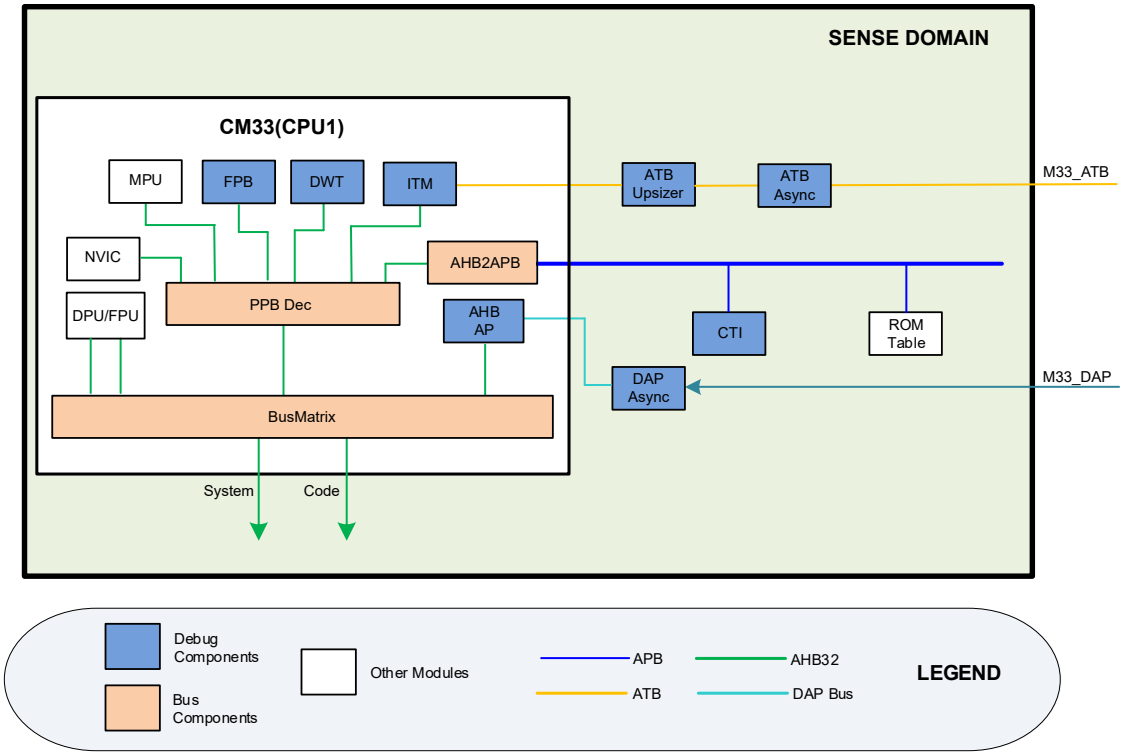


Figure 10. CPU1 platform diagram

7.1.2 Cortex-M33 configuration

Table 21. CM33 configuration for CPU0 and CPU1

Feature	CPU0	CPU1
Domain	COMPUTE	SENSE
Arm TrustZone®	supported	supported
Arm Cortex-M33 built-in Memory Protection Unit (MPU) supporting eight regions	supported	supported
Hardware Floating Point Unit (FPU)	supported	supported
Arm Cortex-M33 built-in Nested Vectored Interrupt Controller (NVIC)	supported	supported
Non-maskable Interrupt (NMI) input	supported	supported
ETM trace (with external buffer), eight breakpoints, four watchpoints	supported	not supported
Instrumentation Trace Macrocell (ITM)	supported	supported
Cortex M33 system tick timer	supported	supported

NOTE

It is recommended that performance critical code be located such that it fetches from the code bus interface as defined by addresses < 0x2000_0000.

7.2 Functional description

7.2.1 CPU0 and CPU1

- CPU0 is the control processor in the Compute domain
  - It is the main control CPU for the chip
  - Responsible for conducting the booting process
  - Responsible for releasing HiFi-4 DSP out of reset and enabling the DSP clock
  - Interfaces with external memories including xSPI Flash, USB, SPI, I2C, UART and eMMC
- CPU1 is the control processor in the Sense domain
  - Accesses all peripherals in the Sense domain and the shared region (peripherals accessible for both Sense and Compute domains)
  - Responsible for releasing HiFi-1 DSP out of reset and enabling the DSP clock

NOTE

Some shared, system-level resources are configured and managed by CPU0.

7.2.2 Nested Vectored Interrupt Controller (NVIC)

CPU0 and CPU1 each include a Nested Vectored Interrupt Controller (NVIC). They both support 8 priority levels for interrupts. The NVIC IPR registers define 3 bits per IRQ.

7.2.3 Floating Point Unit (FPU)

CPU0 and CPU1 both support the Floating Point Unit (FPU). The FPU implements FPv5 floating-point extensions, which supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. The FPU also provides conversions between fixed-point and floating point data formats, and floating-point constant instructions.

The FPU is compliant with the ANSI/IEEE Std 754-2008 standard.

7.2.4 Memory Protection Unit (MPU)

CPU0 and CPU1 support the Memory Protection Unit (MPU). The MPU supports independent attribute settings for each region and ability to export memory attributes.

In addition, a Secure MPU is also supported.

A processor generates a fault if the core access hits the overlapping regions in the case of overlapping memory regions.

Instruction accesses and data accesses in a given region have the same region settings.

7.2.5 Debug

See the Debug overview for more information about the CM33 debug components.

7.2.6 Code bus access

CPU0 and CPU1 code bus can access memory and peripherals as shown in the bus matrix block diagram. CPU0 code bus access is routed to XCACHE1. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable accesses or forwarding the cache write-through and cache miss accesses to the downstream memories through the controller port of this cache controller.

7.2.7 System bus access

All system bus accesses are routed to the target address in destination memories through multilayer AHB matrix target port, see System diagram for more details. CPU0 system bus access is routed to XCACHE0.

7.2.8 Access control

CPU0 code and system bus accesses are checked by the core access control logic, that is, IDAU/SAU and MPU. All requests that miss or bypass the cache are checked by downstream secure AHB bus logic. The caches include protection control signals (HPROT[3:0]) and processing domain bits as part of the tags. If a fetch address hits the cache but the protection control and/or domain bits are different, the cache controller forces a miss with the allocate location the same as the address hit location in the cache. This policy allows all the downstream checks to take place, and this new miss is loaded in the cache with the updated protection control and domain bits overwriting the line with the same address. This keeps the cache coherent while always checking accesses that need to see the downstream checks.

CPU1 access is controlled by secure AHB control.

The CPU0 and CPU1 access to the on chip memory is checked in the AHBSC. See i.MX RT700 Security Reference Manual for more details about AHBSC.

Chapter 8
External Memory Controllers

8.1 Overview

This chip has the following external memory interfaces and controllers:

- Ultra Secured Digital Host Controller (uSDHC) supporting SD, SDIO, eMMC
- External Serial Peripheral Interface (XSPI) Controller supporting flash and PSRAM memory
- Two Universal Serial Bus (USB) Controllers supporting flash memory

8.1.1 System-Level diagram

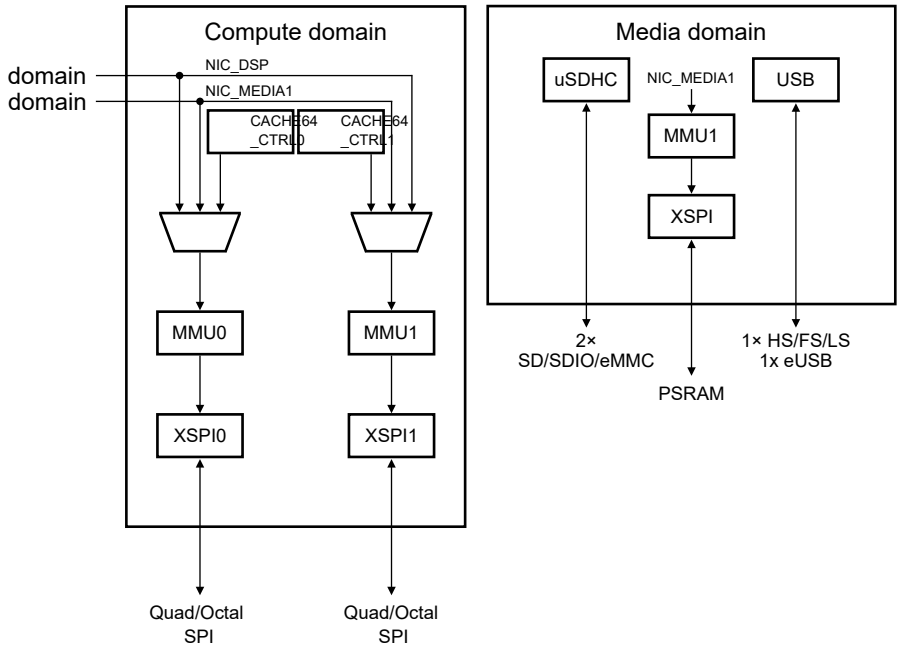


Figure 11. External memory controllers block diagram

8.2 Modules

Table 22. Modules

Modules	Instances	Description	Main features
Ultra Secured Digital Host Controller (uSDHC)	Media domain x2	uSDHC acts as a bridge, passing host bus transactions to the eMMC, SD card, and SDIO by sending commands and performing data accesses to/from the cards.	Conforms to the SD Host Controller Standard Specification version 2.0/3.0 Compatible with the eMMC System Specification version 4.2/4.3/4.4/4.41/4.5 Compatible with the SDIO Specification version 2.0/3.0

Table continues on the next page...

Table 22. Modules (continued)

Modules	Instances	Description	Main features
			Support for various SD, eMMC, SD card, or SDIO data transfer modes Configurable data buffer to transfer data Internal DMA eMMC/SD card detection Power management
External Serial Peripheral Interface (XSPI)	Compute domain x2 Media domain x1	XSPI acts as an interface to a single serial flash memory device.	Support for up to hex mode for data transfers and up to octal mode operation transfers Support for up to eight bidirectional data lines on XSPI0, and up to 16 on XSPI1 and XSPI2 Support for Double Data Rate (DDR)/Double Transfer Rate (DTR) mode Flash memory data strobe signal to support data sampling in DDR and Single Data Rate (SDR) mode Support for PSRAM Memory mapped read/write access by AHB bus Programmable sequence engine
Universal Serial Bus (USB)	Media domain x2	USB controller block provides high performance USB functionality that conforms to the Universal Serial Bus Specification Revision 2.0.	EHCI-compatible for host mode High Speed (HS), Full Speed (FS) and Low Speed (LS) operation in host mode Support for HS and FS operation in device mode Support for low-power suspend mode Support for low-power sleep mode Embedded DMA controller

Chapter 9
Boot ROM

9.1 Overview

NOTE

This chapter is also available in the i.MX RT700 Security Reference Manual, with differences related to security functionality. The chapter in the Reference Manual does not show the security relevant information in it.

ROM bootloader runs from internal read-only memory (ROM). After a power-on reset or warm reset, the ROM bootloader is executed.

The 256 KB on-chip ROM bootloader supports the following features:

- Prime boot from external memories including XSPI NOR, XSPI NAND and eMMC
- Recovery boot from LP_FLEXCOMM (LPSPI) NOR memory.
- In-System Programming (ISP) via USB0 (HS USB), USB1 (eUSB), LPUART, LPSPI and LPI2C.
- Debug Mailbox (DBGMB)
- ROM APIs.

The ROM bootloader resides in the memory region starting at address 0x1300_0000. The diagram below illustrates the memory map utilized during ROM execution. During bootloader operation, the first 84 KB of SRAM (0x3000_0000 – 0x3001_4FFF) must be reserved and not used by other components. As a result, load-to-RAM images should be placed in memory regions starting from 0x0001_5000 (secure alias: 0x1001_5000) or 0x2001_5000 (secure alias: 0x3001_5000). Once the system has successfully booted, the reserved 84 KB of SRAM becomes available for general use.

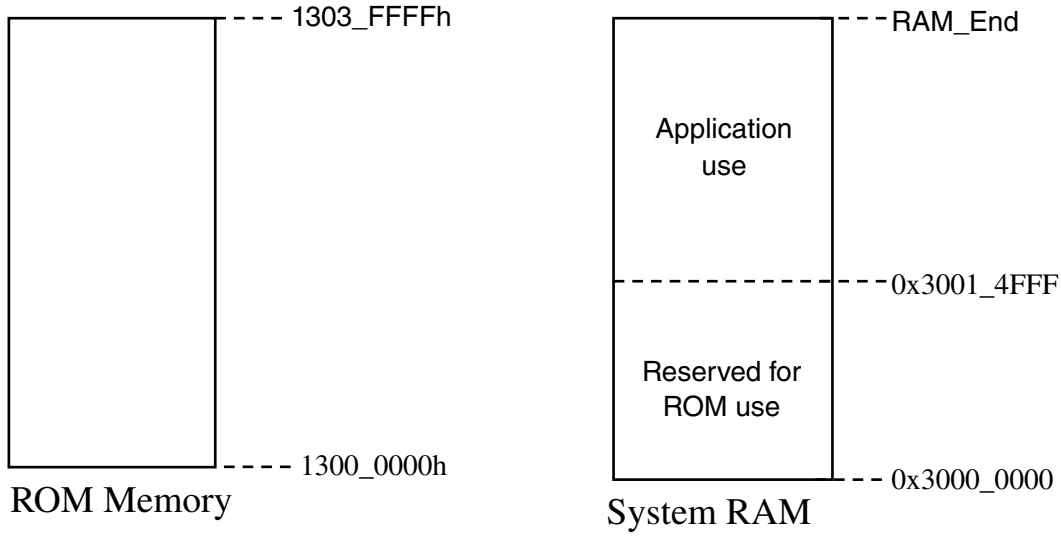


Figure 12. ROM bootloader memory usage

9.2 High level boot flow

On this chip, Boot ROM first downloads boot image from external memory to RAM based on boot mode and information in the image header. For execute-in-place (XIP) boot image, Boot ROM will skip the download operation. Then, Boot ROM validates the boot image. Lifecycle, boot configuration and information in the image decide which validation method is used for this image. If the boot image is valid, Boot ROM jumps to the image and executes.

Following are the boot modes available for the boot ROM:

- Prime boot from XSPI NOR
- Prime boot from serial NAND connected to XSPI
- Prime boot from eMMC
- Recovery boot from LPSPI NOR

Boot mode can be selected using Boot pins, ROM scratch register or fuses. Recovery boot is executed when prime boot fails, and recovery boot is enabled in fuse.

Boot ROM can do simple image check, CRC check or ECDSA authentication at Develop lifecycle when secure boot is not enforced.

If secure boot is enforced or lifecycle is after Develop:

- for XIP image, ECDSA authentication is enforced
- for copy-to-RAM image, it must be in secure binary (SB3) format on original external memory. CRC check or ECDSA authentication can be done for image downloaded to RAM.

The following figure shows the high level boot flow:

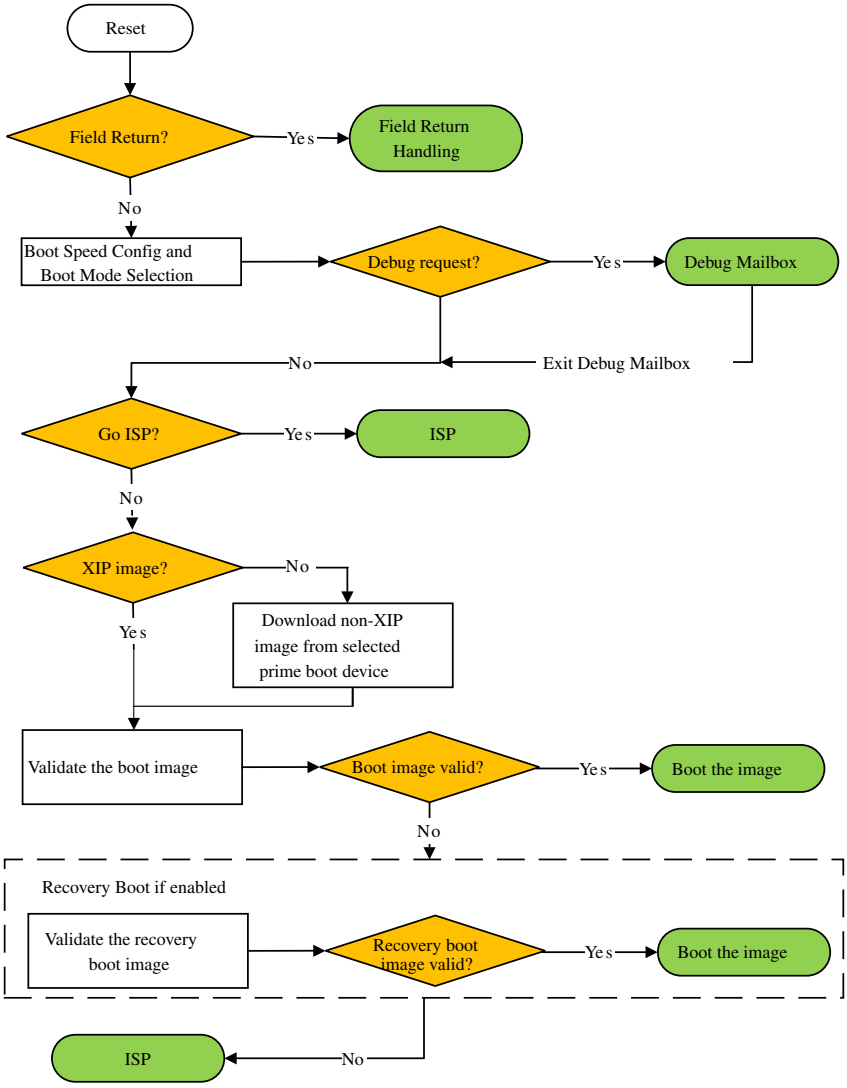


Figure 13. High level boot flow

9.3 Pin allocation

9.3.1 Boot pin allocation

PIO1_8 and PIO1_7 are allocated as Boot pins which select the prime boot source or forces entry to ISP mode. Boot pins work only when, in BOOT_CFG0, BOOT_SRC[1:0] = 00b and ISP_MODE_PIN_DIS = 0b.

Table 23. Boot pin allocation

PIO1_8	PIO1_7	Definition
0	0	Boot from eMMC connected to uSDHC0
0	1	Boot from serial NOR connected to XSPI0
1	0	Enter ISP (UART, SPI_target, I2C_target, USB0, USB1)
1	1	Boot from serial NOR connected to XSPI1

9.3.2 Boot interface pin allocation

Boot ROM provides flexibility that you can configure the boot interface according to your hardware design.

When fuse BOOT_SRC[1:0] is not set (00b), Boot pin can be used to determine which source is used to boot. Fuse BOOT_SRC and BOOT_IF_INST can be configured together to determine the boot source.

For prime boot from serial NOR or NAND, the memory can be connected to XSPI0 or XSPI1. Boot ROM also supports copying code from the boot medium to an external pSRAM connected to XSPI0 or XSPI1.

For prime boot from eMMC, the memory can be connected to uSDHC0 or uSDHC1.

Boot ROM supports recovery boot from 1-bit serial NOR connected to LP_FLEXCOMM (LPSPI) if fuse REC_BOOT_EN is set. You can choose any one instance among LP_FLEXCOMM0 to LP_FLEXCOMM13 to connect serial NOR by setting fuse REC_BOOT_FC. You can select Chip-Select (CS) pin in CS0 to CS3 by using fuse REC_BOOT_FC_PCS and select LP_FLEXCOMM (LPSPI) baudrate by using fuse REC_BOOT_FC_BAUDRATE.

Detailed pin allocation for each boot interface is described in the table below.

Table 24. Boot interface pin allocation

<table>
<tr><th>Boot interface</th><th>Instance and memory supported</th><th>Pin allocation</th></tr>
<tr><td rowspan="2">XSPI</td><td>XSPI0
Quad NOR
Octal NOR
8-bit pSRAM
Quad NAND</td><td>Table 25. XSPI0 pin signals
<table><tr><td>PIO6_0</td><td>XSPI0_SCLK0_N</td></tr><tr><td>PIO6_1</td><td>XSPI0_SCLK0</td></tr><tr><td>PIO6_2</td><td>XSPI0_SS0_N</td></tr><tr><td>PIO6_3</td><td>XSPI0_DATA0</td></tr><tr><td>PIO6_4</td><td>XSPI0_DATA1</td></tr><tr><td>PIO6_5</td><td>XSPI0_DATA2</td></tr><tr><td>PIO6_6</td><td>XSPI0_DATA3</td></tr><tr><td>PIO6_7</td><td>XSPI0_DQS0</td></tr><tr><td>PIO6_8</td><td>XSPI0_DATA4</td></tr><tr><td>PIO6_9</td><td>XSPI0_DATA5</td></tr><tr><td>PIO6_10</td><td>XSPI0_DATA6</td></tr><tr><td>PIO6_11</td><td>XSPI0_DATA7</td></tr><tr><td>PIO6_12</td><td>XSPI0_SS1_N</td></tr></table></td></tr>
<tr><td>XSPI1
Quad NOR
Octal NOR
8/16-bit pSRAM
Quad NAND</td><td>Table 26. XSPI1 pin signals
<table><tr><td>PIO5_0</td><td>XSPI1_SS0_N</td></tr><tr><td>PIO5_1</td><td>XSPI1_DATA0</td></tr><tr><td>PIO5_2</td><td>XSPI1_DATA1</td></tr></table></td></tr>
</table>

Table 26. XSPI1 pin signals (continued)

Boot interface	Instance and memory supported	Pin allocation
		Table 26. XSPI1 pin signals (continued)
		PIO5_3 XSPI1_DATA2
		PIO5_4 XSPI1_DATA3
		PIO5_5 XSPI1_DQS0
		PIO5_6 XSPI1_DATA4
		PIO5_7 XSPI1_DATA5
		PIO5_8 XSPI1_DATA6
		PIO5_9 XSPI1_DATA7
		PIO5_10 XSPI1_SCLK0
		PIO5_11 XSPI1_SCLK0_N
		PIO5_12 XSPI1_DATA8
		PIO5_13 XSPI1_DATA9
		PIO5_14 XSPI1_DATA10
		PIO5_15 XSPI1_DATA11
		PIO5_16 XSPI1_DQS1
		PIO5_17 XSPI1_DATA12
		PIO5_18 XSPI1_DATA13
		PIO5_19 XSPI1_DATA14
		PIO5_20 XSPI1_DATA15
uSDHC	uSDHC0 eMMC	Table 27. uSDHC0 pin signals
		PIO7_0 SDHC0_CLK
		PIO7_1 SDHC0_CMD
		PIO7_2 SDHC0_DATA4
		PIO7_3 SDHC0_DATA5
		PIO7_4 SDHC0_DATA6

Table 24. Boot interface pin allocation (continued)

<table>
<tr><th>Boot interface</th><th>Instance and memory supported</th><th>Pin allocation</th></tr>
<tr><td rowspan="3"></td><td></td><td>Table 27. uSDHC0 pin signals (continued)
<table><tr><td>PIO7_5</td><td>SDHC0_DATA7</td></tr><tr><td>PIO7_6</td><td>SDHC0_DATA0</td></tr><tr><td>PIO7_7</td><td>SDHC0_DATA1</td></tr><tr><td>PIO7_8</td><td>SDHC0_DATA2</td></tr><tr><td>PIO7_9</td><td>SDHC0_DATA3</td></tr><tr><td>PIO7_12</td><td>SDHC0_RESET_N</td></tr></table></td></tr>
<tr><td>uSDHC1
eMMC</td><td>Table 28. uSDHC1 pin signals
<table><tr><td>PIO7_13</td><td>SDHC1_DATA4</td></tr><tr><td>PIO7_14</td><td>SDHC1_DATA5</td></tr><tr><td>PIO7_15</td><td>SDHC1_DATA6</td></tr><tr><td>PIO7_16</td><td>SDHC1_DATA7</td></tr><tr><td>PIO7_17</td><td>SDHC1_CLK</td></tr><tr><td>PIO7_18</td><td>SDHC1_CMD</td></tr><tr><td>PIO7_19</td><td>SDHC1_DATA0</td></tr><tr><td>PIO7_20</td><td>SDHC1_DATA1</td></tr><tr><td>PIO7_21</td><td>SDHC1_DATA2</td></tr><tr><td>PIO7_22</td><td>SDHC1_DATA3</td></tr><tr><td>PIO7_25</td><td>SDHC1_RESET_N</td></tr></table></td></tr>
<tr><td>LP_FLEXCOMM
(configured as LPSPI)</td><td>LP_FLEXCOMMn (n is in 0 – 13)
single-bit NOR
1</td><td>Table 29. LP_FLEXCOMM0 pin signals
<table><tr><td>PIO0_31</td><td>LP_FLEXCOMM0_P0 (SDO)</td></tr><tr><td>PIO1_0</td><td>LP_FLEXCOMM0_P1 (SCK)</td></tr><tr><td>PIO1_2</td><td>LP_FLEXCOMM0_P3 (PCS[0])</td></tr></table></td></tr>
</table>

Table continues on the next page...

Table 24. Boot interface pin allocation (continued)

<table>
<tr><th>Boot interface</th><th>Instance and memory supported</th><th>Pin allocation</th></tr>
<tr><td></td><td></td><td>
Table 30. LP_FLEXCOMM1 pin signals
<table>
<tr><td>PIO1_7</td><td>LP_FLEXCOMM1_P0 (SDO)</td></tr>
<tr><td>PIO1_8</td><td>LP_FLEXCOMM1_P1 (SCK)</td></tr>
<tr><td>PIO1_9</td><td>LP_FLEXCOMM1_P2 (SDI)</td></tr>
<tr><td>PIO1_10</td><td>LP_FLEXCOMM1_P3 (PCS[0])</td></tr>
</table>
Table 31. LP_FLEXCOMM2 pin signals
<table>
<tr><td>PIO1_11</td><td>LP_FLEXCOMM2_P0 (SDO)</td></tr>
<tr><td>PIO1_12</td><td>LP_FLEXCOMM2_P1 (SCK)</td></tr>
<tr><td>PIO1_14</td><td>LP_FLEXCOMM2_P2 (SDI)</td></tr>
<tr><td>PIO1_15</td><td>LP_FLEXCOMM2_P3 (PCS[0])</td></tr>
</table>
Table 32. LP_FLEXCOMM3 pin signals
<table>
<tr><td>PIO1_18</td><td>LP_FLEXCOMM3_P0 (SDO)</td></tr>
<tr><td>PIO1_17</td><td>LP_FLEXCOMM3_P1 (SCK)</td></tr>
<tr><td>PIO1_16</td><td>LP_FLEXCOMM3_P2 (SDI)</td></tr>
<tr><td>PIO1_19</td><td>LP_FLEXCOMM3_P3 (PCS[0])</td></tr>
</table>
Table 33. LP_FLEXCOMM4 pin signals
<table>
<tr><td>PIO0_3</td><td>LP_FLEXCOMM4_P0 (SDO)</td></tr>
<tr><td>PIO0_4</td><td>LP_FLEXCOMM4_P1 (SCK)</td></tr>
<tr><td>PIO0_6</td><td>LP_FLEXCOMM4_P2 (SDI)</td></tr>
<tr><td>PIO0_7</td><td>LP_FLEXCOMM4_P3 (PCS[0])</td></tr>
</table>
Table 34. LP_FLEXCOMM5 pin signals
<table>
<tr><td>PIO0_10</td><td>LP_FLEXCOMM5_P0 (SDO)</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table 24. Boot interface pin allocation (continued)

<table>
<tr><th>Boot interface</th><th>Instance and memory supported</th><th>Pin allocation</th></tr>
<tr><td></td><td></td><td>
Table 34. LP_FLEXCOMM5 pin signals (continued)
<table>
<tr><td>PIO0_11</td><td>LP_FLEXCOMM5_P1 (SCK)</td></tr>
<tr><td>PIO0_12</td><td>LP_FLEXCOMM5_P2 (SDI)</td></tr>
<tr><td>PIO0_13</td><td>LP_FLEXCOMM5_P3 (PCS[0])</td></tr>
</table>
Table 35. LP_FLEXCOMM6 pin signals
<table>
<tr><td>PIO0_21</td><td>LP_FLEXCOMM6_P0 (SDO)</td></tr>
<tr><td>PIO0_22</td><td>LP_FLEXCOMM6_P1 (SCK)</td></tr>
<tr><td>PIO0_19</td><td>LP_FLEXCOMM6_P2 (SDI)</td></tr>
<tr><td>PIO0_20</td><td>LP_FLEXCOMM6_P3 (PCS[0])</td></tr>
</table>
Table 36. LP_FLEXCOMM7 pin signals
<table>
<tr><td>PIO0_24</td><td>LP_FLEXCOMM7_P0 (SDO)</td></tr>
<tr><td>PIO0_25</td><td>LP_FLEXCOMM7_P1 (SCK)</td></tr>
<tr><td>PIO0_26</td><td>LP_FLEXCOMM7_P2 (SDI)</td></tr>
<tr><td>PIO0_27</td><td>LP_FLEXCOMM7_P3 (PCS[0])</td></tr>
</table>
Table 37. LP_FLEXCOMM8 pin signals
<table>
<tr><td>PIO4_0</td><td>LP_FLEXCOMM8_P0 (SDO)</td></tr>
<tr><td>PIO4_1</td><td>LP_FLEXCOMM8_P1 (SCK)</td></tr>
<tr><td>PIO4_2</td><td>LP_FLEXCOMM8_P2 (SDI)</td></tr>
<tr><td>PIO4_3</td><td>LP_FLEXCOMM8_P3 (PCS[0])</td></tr>
</table>
Table 38. LP_FLEXCOMM9 pin signals
<table>
<tr><td>PIO4_7</td><td>LP_FLEXCOMM9_P0 (SDO)</td></tr>
<tr><td>PIO4_8</td><td>LP_FLEXCOMM9_P1 (SCK)</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table 24. Boot interface pin allocation (continued)

<table>
<tr><th>Boot interface</th><th>Instance and memory supported</th><th>Pin allocation</th></tr>
<tr><td></td><td></td><td>
Table 38. LP_FLEXCOMM9 pin signals (continued)
<table>
<tr><td>PIO4_9</td><td>LP_FLEXCOMM9_P2 (SDI)</td></tr>
<tr><td>PIO4_10</td><td>LP_FLEXCOMM9_P3 (PCS[0])</td></tr>
</table>
Table 39. LP_FLEXCOMM10 pin signals
<table>
<tr><td>PIO4_14</td><td>LP_FLEXCOMM10_P0 (SDO)</td></tr>
<tr><td>PIO4_15</td><td>LP_FLEXCOMM10_P1 (SCK)</td></tr>
<tr><td>PIO4_16</td><td>LP_FLEXCOMM10_P2 (SDI)</td></tr>
<tr><td>PIO4_17</td><td>LP_FLEXCOMM10_P3 (PCS[0])</td></tr>
</table>
Table 40. LP_FLEXCOMM11 pin signals
<table>
<tr><td>PIO5_14</td><td>LP_FLEXCOMM11_P0 (SDO)</td></tr>
<tr><td>PIO5_15</td><td>LP_FLEXCOMM11_P1 (SCK)</td></tr>
<tr><td>PIO5_16</td><td>LP_FLEXCOMM11_P2 (SDI)</td></tr>
<tr><td>PIO5_17</td><td>LP_FLEXCOMM11_P3 (PCS[0])</td></tr>
</table>
Table 41. LP_FLEXCOMM12 pin signals
<table>
<tr><td>PIO5_7</td><td>LP_FLEXCOMM12_P0 (SDO)</td></tr>
<tr><td>PIO5_8</td><td>LP_FLEXCOMM12_P1 (SCK)</td></tr>
<tr><td>PIO5_9</td><td>LP_FLEXCOMM12_P2 (SDI)</td></tr>
<tr><td>PIO5_6</td><td>LP_FLEXCOMM12_P3 (PCS[0])</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table 24. Boot interface pin allocation (continued)

Boot interface	Instance and memory supported	Pin allocation
		Table 42. LP_FLEXCOMM13 pin signals
		PIO5_3 — LP_FLEXCOMM13_P0 (SDO)
		PIO5_4 — LP_FLEXCOMM13_P1 (SCK)
		PIO5_5 — LP_FLEXCOMM13_P2 (SDI)
		PIO5_10 — LP_FLEXCOMM13_P3 (PCS[0])

9.3.3 ISP interface pin allocation

ROM bootloader provides in-system programming (ISP) utility that operates over a serial connection on the MCUs. It enables quick and easy programming of MCUs through the entire product lifecycle. Users can utilize host tools to download application to memory medium via serial interfaces including UART, USB0, USB1, SPI or I2C. Table below shows the pin allocation of ISP interfaces.

Table 43. ISP interface pin allocation

ISP interface	Pin allocation	
UART		
	PIO0_31	LP_FLEXCOMM0_P0/RXD
	PIO1_0	LP_FLEXCOMM0_P1/TXD
USB0		
	USB0_VBUS_DETECT	
	USB0_DP	
	USB0_DM	
USB1 (eUSB)		
	EUSB_DM	
	EUSB_DP	
SPI targets		
	PIO3_0	LPSPI14_SOUT
	PIO3_1	LPSPI14_SCK
	PIO3_2	LPSPI14_SIN
	PIO3_3	LPSPI14_PCS0

Table 43. ISP interface pin allocation (continued)

ISP interface	Pin allocation	
I2C targets		
	PIO1_17	LP_FLEXCOMM3_P1/SCL
	PIO1_18	LP_FLEXCOMM3_P0/SDA

9.4 Boot configuration and boot image

9.4.1 Boot speed

Three different boot speeds (compute_main_clk) are supported, which can be configured from fuse BOOT_SPEED[1:0]. Boot ROM configures clock based on configuration in BOOT_SPEED[1:0] early in the boot flow. The default boot speed is 192 MHz. Table below is the fuse definition.

Table 44. Fuse definition

Fuse	Fuse word	Bits	Definition
BOOT_SPEED	136 BOOT_CFG0	13:12 (2 bits)	System clock speed during boot. • 00 - Normal boot (192 MHz) – VDD2 = 1.0 V • 01 - High speed boot (300 MHz) – VDD2 = 1.1 V • 10 - Low speed boot (32 MHz) – VDD2 = 1.0 V • 11 - Normal boot (192 MHz) – VDD2 = 1.0 V **NOTE** If the VDD2 voltage is supplied externally, then one of the 1.0 V boot speed options must be used. The high speed boot option should not be selected.

9.4.2 Boot source selection from fuses

Fuse configuration for boot source selection in word 0 BOOT_CFG0:

Table 45. Fuse configuration for boot source selection

Fuse	Fuse word	Bits	Definition	Description
BOOT_SRC[1:0]	136	2	00 - Boot pins determine the source 01 - XSPI NOR flash 10 - uSDHC flash 11 - XSPI NAND flash	These two bits specify the boot source

Table continues on the next page...

Table 45. Fuse configuration for boot source selection (continued)

Fuse	Fuse word	Bits	Definition	Description
BOOT_IF_INST[2]		1	0 - Instance 0 (XSPI0 or uSDHC0) 1 - Instance 1 (XSPI1 or uSDHC1)	This bit specifies boot instance when XSPI or uSDHC is selected as boot source

Image offset definition for various boot media:

Table 46. Image offset definition for various boot media

Boot media	Image Offset
XSPI NOR boot	4000h
XSPI NAND boot	0000h
eMMC boot	1000h
LP_FLEXCOMM 1-bit NOR recovery boot	1000h

9.4.3 Image header

After completing boot mode detection, the bootloader starts validating the vector table and image header if the image is present in the boot media (external flash memory).

The boot ROM does the following for image validity check:

- Validates the image using header and fuse settings when secure boot is enabled. See the "Secure Boot ROM" chapter in the *Security Reference Manual* for details.
- Validates the image using CRC32 when secure boot is not enabled, if the CRC check is present in the image header.
- Validates whether SP and PC are valid RAM and NVM addresses of the chip.
- Validates the TZ-M image type if the aforementioned image checks pass.

The beginning of the image follows the format described in Table 47. The bootloader begins scanning for user images by examining the image type marker located at offset 24h. If the value matches any supported image type markers, the image header's validation begins, and after its completion, the qualification continues by examining the TZ-M "imageType" field:

- If it is a CRC image, you use the "image length" field value as the length to perform a CRC calculation based on the image in the flash memory (see Table 47 for more). The CRC calculation begins at offset 0h from the beginning of the image header and continues up to the number of bytes specified by the image length, which does not include the "offsetToExtendedHeader" field that makes up the CRC value field. This means that the calculated CRC skips the CRC value field. The result is then compared to the "offsetToExtendedHeader" entry in the structure and the image is considered valid if a match exists; otherwise, the image is considered invalid. The CRC calculation is not performed if the image is not a CRC image.
- If it is a signed image, the "image length" field value is used as the length to perform authentication (ECDSA), which is performed on the image in the external flash memory. The authentication begins at offset 0h from the beginning of the image header, and continues up to the number of bytes specified by the image length. The "offsetToExtendedHeader" field value points to the offset that holds the certificates.

Table 47. Image header in vector table

Offset (hex)	Size (bytes)	Symbol	Description
00	4	Initial SP	Stack pointer
04	4	Initial PC	Application's first execution instruction
08	24	Vector table	Cortex-M33 core's vector table entries
20	4	image length	Length of the current image (total length, including the signature), which is set to the actual image length if the image type is another value
24	4	imageType	Bit [7:0]: Image type • 0h - Plain image • 2h - Plain image with CRC • 4h - Signed image • 6h - SB3 manifest • Other values are reserved. Bit [12:8]: Reserved Bit [13]: Indicates whether the image includes TZM preset data: • 0 - Image does not contain TZM preset data. • 1 - Image contains TZM preset data. Bits[31:14]: Reserved.
28	4	offsetToExtendedHeader	Offset to extended header: • For a signed image (imageType = 4h), this is the offset to the certificate header block • For a CRC image (imageType = 2h), this is the crcChecksum value
2C	8	Vector table	Cortex-M33 core's vector table entries
34	4	imageExecutionAddress	The execution address of the image is: • Set to 0 for XIP image • Set to actual image execution address load_to_RAM image
38		Vector table	Cortex-M33 core's vector table entries

NOTE

Load_to_RAM image: The image must have a nonzero imageExecutionAddress and image_length in the image header.

9.4.4 Image version and anti-rollback

Boot ROM supports anti-rollback for signed image and SB3.1 image. It compares the version of image with the version value programmed in fuse and only boots the image with version equal to or greater than the version in fuse. See details of signed image and SB3 image structure in i.MXRT700 Security Reference Manual.

For signed image, ROM compares the image version with fuse SEC_FW_VER (word 128 - 131). Boot ROM does not boot image with version lower than the version in fuse. For fuse SEC_FW_VER (secure firmware version), value 1b means version 1, value 11b means version 2, value 111b means version 3, etc. For SB3.1 image, users can enable image version check using SB3.1 fwVersionCheck command. Boot ROM checks the version of SB3.1 image with the fuse required in checkFwVersion command, either NS_FW_VER (non-secure firmware version, word 112 - 127) or SEC_FW_VER. Fuse NS_FW_VER works similarly like SEC_FW_VER, value 1b means version 1, value 11b means version 2, value 111b means version 3, etc.

NOTE

- SEC_FW_VER fuse can be used for major or critical updates and NS_FW_VER fuse can be used for minor or non-critical updates.
- Only lower 16 bits are active for both fuse words (SEC_FW_VER/NS_FW_VER). Hence, there can be up to 64 updates when using SEC_FW_VER fuse and up to 256 updates when using NS_FW_VER fuse.
- If an application doesn't use SB3.1 fwVersionCheck command with kNBOOT_CNT_nonsecure counter ID, then fuse words 112-127 can be re-purposed as customer defined counters (CUST_CTR9 – CUST_CTR24). A potential use case is implementing non-volatile counters required by the TF-M framework.

9.5 Prime boot from XSPI NOR

Boot ROM supports prime boot from Quad NOR, Octal NOR and HyperFlash connected to XSPI instance 0 or 1. Both execute-in-place (XIP) and load-to-RAM boot image are supported for XSPI NOR. For load-to-RAM boot image, the image can be loaded to on-chip SRAM (starts from offset 15000h), or pSRAM connected to the other XSPI instance. For example, if XSPI0 is selected as boot source and NOR is connected to it, the ROM can load the boot image from XSPI0 NOR to pSRAM connected to XSPI1. Note that Boot ROM only supports XSPI instance 0 and 1.

At Develop (3h) lifecycle, XIP image and load-to-RAM image can be any image with valid image header. When secure boot is enforced or lifecycle is after Develop (3h), the XIP image must be correctly signed image to boot successfully, and copy-to-RAM image must be in SB3 format and the image to generate SB3 should either be CRC image or signed image.

There are three ways to start XSPI boot.

- BOOT_SRC[1:0] = 00b.
  - If BOOT_PIN_MODE = 01b, XSPI0 is selected as boot source.
  - If BOOT_PIN_MODE = 11b, XSPI1 is selected as boot source.
- BOOT_SRC[1:0] = 01b, Boot ROM directly boots from XSPI NOR, and XSPI instance is selected via BOOT_IF_INST
- Use runBootloader API with EB04_*n*000h (*n* is 0 or 1 to select XSPI instance) as the parameter

Once XSPI NOR boot is selected, Boot ROM configures XSPI based on FCB or AutoProbe value and tries to find the boot image at offset 4000h of the XSPI NOR. XSPI NOR image layout is like figure below, first 4000h is platform configuration, and XSPI NOR boot support dual image boot if fuses are configured correctly, see sections below for more details.

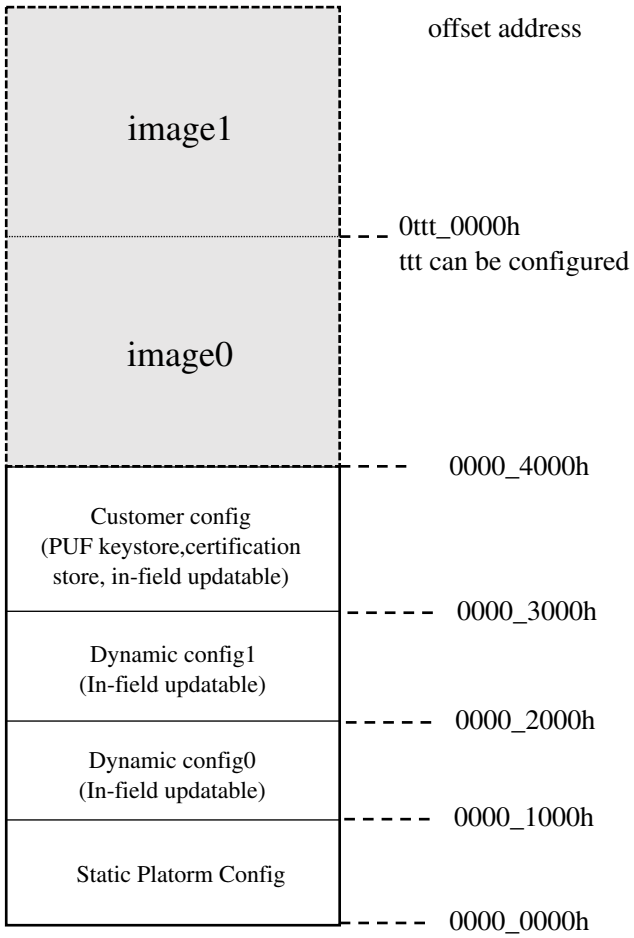


Figure 14. XSPI dual image boot

9.5.1 Platform configuration

First 4000h of XSPI NOR is platform configuration which is divided into two groups:

1. Static or factory programmed platform configuration including flash configuration block (FCB) of NOR, and optional configuration for pSRAM.

```
typedef struct {
 uint8_t xspi_fcb_block[768];// Configure structure for boot device connected to XSPI0/XSPI1
interface.
 uint8_t psram_config_block[768];// Configure structure for pSRAM device connected to XSPI0/
XSPI1 interface.
 uint8_t xspi2_fcb_block[768];// Configure structure for pSRAM device connected to XSPI2
interface. Only for users' usage, Boot ROM doesn't use this part
 uint8_t reserved[1792];// Reserved for future usage
} static_platform_config_t;
```

2. Dynamic or in-field updatable configuration.

```
typedef struct {
 uint32_t tag;// Check this tag to see if the dynamic config is programmed
```

```
 uint32_t cfg_version; // Version of this structure. Used for identifying the active
configuration by picking the page which has the higher number between ping & pong pages.
 uint32_t arb_version;// anti-roll back version to be compared with SEC_FW_VER or NS_FW_VER
monotonic counter fields in OTP.
 uint8_t reserved[4]; // Reserved area for future
}

typedef struct IPED_CTX_CFG_t{
uint32_t iv_lsb_erase_count;// IV for each context.
uint32_t iv_msb;// Random number generated on each erase operation.
uint64_t aad_data;//IPED context specific AAD data. This field used for GCM mode only. Ignored
in XEX & CTR modes.
uint32_t start_addr;// Start address of the context. The bit-field layout is same as
IPEDCTXaSTART register including GCM mode bit & AHB_ERROR bit.
uint32_t end_Addr;// End address of the context. The bit-field layout is same as IPEDCTXaEND
register. Lower bits for locking the context.
} enc_ctx_cfg[NUM_CTX]; //AES-GCM Encrypted. For RT700 NUM_CTX = 16.

typedef struct IPED_CTX_CFG_t psram_ctx_cfg[XSPI1_NUM_CTX];// For systems where XSPI0 is
connected to flash and XSPI1 connected to pSRAM. For RT700 XSPI1_NUM_CTX = 16.For systems where
flash is connected XSP1.
```

9.5.2 XSPI boot configuration

The boot ROM supports access to different Quad and Octal SPI NOR flash memory chips from various vendors via the XSPI interface using 1-bit, 4-bit (Quad), or 8-bit (Octal) mode. It uses the flash memory configuration block (FCB) located at offset 0h on the flash memory chip or flash memory auto-probe feature that fuse specifies:

- If XSPI_FLASH_AUTO_PROBE_EN in word 152 XSPI_CFG0 is blown, the bootloader performs the flash memory auto-probe sequence using parameters blown in:
  - The XSPI_FLASH_PROBE_TYPE field that defines the flash memory auto-probe type.
  - The XSPI_FLASH_TYPE fields that define the flash memory type.
  - The XSPI_FLASH_BOOT_FREQ field that defines the flash memory access speed.
- If XSPI_FLASH_AUTO_PROBE_EN is not blown, the bootloader looks at offset 0h on the flash memory chip; if data at offset 0h is equal to 4246_4346h, the bootloader reads the whole 576-byte FCB into the on-chip SRAM and configures the XSPI controller using this FCB accordingly.

  After performing the aforementioned operation, the bootloader starts performing the normal boot flow.

Table 48. XSPI boot configurations in fuse index 152 (XSPI_CFG0)

Fuse	Definition	Settings	Shipped value
XSPI_CFG0[0]	XSPI_FLASH_AUTO_PROBE_EN	Quad SPI or Octal SPI flash memory auto-probe feature enable 0 - Disable 1 - Enable	0
XSPI_CFG0[3:1]	XSPI_FLASH_PROBE_TYPE	Quad SPI or Octal SPI flash memory probe type 000 - Quad SPI NOR flash memory 001 - Macronix Octal flash memory	0

Table continues on the next page...

Table 48. XSPI boot configurations in fuse index 152 (XSPI_CFG0) (continued)

Fuse	Definition	Settings	Shipped value
		010 - Micron Octal flash memory 011 - Adesto Octal flash memory 100 - Octal SDR 101 - Octal DDR 110 - Reserved 111 - Reserved	
XSPI_CFG0[6:4]	XSPI_FLASH_TYPE	Typical serial NOR flash memory types 000 - Quad SPI flash memory address (by default, the chip supports 3B read) 001 - Reserved 010 - HyperFlash 1.8 V 011 - HyperFlash 3.3 V 100 - MXIC Octal DDR 101 - Micron Octal DDR 110 - Reserved 111 - Reserved	0
XSPI_CFG0[10:7]	XSPI_FLASH_DUMMY_CYCLES	Quad SPI or Octal SPI dummy cycles for read command 0000 - Dummy cycles probed automatically Other value - Number of dummy cycles	0
XSPI_CFG0[13:11]	XSPI_FLASH_BOOT_FREQ	Quad SPI or Octal SPI flash interface frequency 000 - 80 MHz 001 - 60 MHz 010 - 50 MHz 011 - 100 MHz 100 - 120 MHz 101 - 133 MHz 110 - 166 MHz 111 - 200 MHz	0
XSPI_CFG0[14]	XSPI_RESET_PIN_EN	Enable to reset flash memory by reset pin or JEDEC 0 - Disable	0

Table continues on the next page...

Table 48. XSPI boot configurations in fuse index 152 (XSPI_CFG0) (continued)

Fuse	Definition	Settings	Shipped value
		1 - Enable	
XSPI_CFG0[15]	XSPI_NOR_RESET_TYPE	This field is valid only if XSPI_RESET_PIN_EN is set. 0 - Pin reset 1 - JEDEC reset	0
XSPI_CFG0[18:16]	XSPI_RESET_PIN_PORT	GPIO port number. This field is valid only if XSPI_RESET_PIN_EN is set.	0
XSPI_CFG0[23:19]	XSPI_RESET_PIN_GPIO	GPIO pin number. This field is valid only if XSPI_RESET_PIN_EN is set.	0
XSPI_CFG0[25:24]	XSPI_HOLD_TIME	Wait time before accessing serial flash memory 00 - 500 µs 01 - 1 ms 10 - 3 ms 11 - 10 ms	0
XSPI_CFG0[29:26]	XSPI_PWR_HOLD_TIME	Delay (wait time) after POR before accessing Quad SPI or Octal SPI flash memory devices, in addition to the delay that the XSPI_HOLD_TIME field specifies 0000 - No delay 0001 - Additional 100 µs 0010 - Additional 500 µs 0011 - Additional 1 ms 0100 - Additional 10 ms 0101 - Additional 20 ms 0110 - Additional 40 ms 0111 - Additional 60 ms 1000 - Additional 80 ms 1001 - Additional 100 ms 1010 - Additional 120 ms 1011 - Additional 140 ms 1100 - Additional 160 ms 1101 - Additional 180 ms 1110 - Additional 200 ms	0

Table continues on the next page...

Table 48. XSPI boot configurations in fuse index 152 (XSPI_CFG0) (continued)

Fuse	Definition	Settings	Shipped value
		1111 - Additional 220 ms	
XSPI_CFG0[31:30]	XSPI_FLASH_DUMMY_CYCLES_UPPER	Quad/Octal-SPI dummy cycles upper 2 bits for read command.	0

Table 49. XSPI boot configurations in fuse index 153 (XSPI_CFG1)

Fuse	Definition	Settings
XSPI_CFG1[6:0]	XSPI_DELAY_CELL_NUM	Delay cell numbers for flash read 0 - Use the Delay cell value calcu Others - Use the Delay cell value
XSPI_CFG0[16:7]	XSPI_FLASH_SEC_IMAGE_OFFSET	Offset of the second image
XSPI_CFG1[23:17]	XSPI_FLASH_IMAGE_SIZE	Size of the second image
XSPI_CFG1[24]	XSPI_FLASH_ENHANCE_MODE	XSPI enhance mode 000 - Disable 101 - Force using interanal loopba Others - Reserved
XSPI_CFG1[31:25]	Reserved	

Table 50. XSPI FCB settings

Field	Offset (hex)	Size (bytes)	Description
tag	000	4	Configuration block tag Setting: 4246_4346h
version	004	4	Configuration block that the boot ROM uses internally Setting: 5601_0000h
Reserved	008	4	Reserved
readSampleClkSrc	00C	1	Read sampling clock source options: 0h – Internal sampling 1h – Reserved 2h – DQS pad loopback 3h – External DQS signal
csHoldTime	00D	1	CS hold time in terms of flash memory clock cycles Recommended value is 3h.

Table continues on the next page...

Table 50. XSPI FCB settings (continued)

Field	Offset (hex)	Size (bytes)	Description
csSetupTIme	00E	1	CS set up time in terms of flash memory clock cycles The recommended value is 3h.
columAddressWidth	00F	1	Column address width: • Set to 3h for HyperFlash NOR flash memories. • Set to 0h for other flash memory chips.
deviceModeCfgEnable	010	1	Enables the chip mode configuration sequence: 1h – Enable 0h – Disable
deviceModeType	011	1	Argument for Device mode configuration For example, Quad enable setting in Status Register2 for some Quad SPI flash memory devices: 0h – No mode change 1h – Quad enable (switch from SPI to Quad mode) 2h – Spi2Xpi (switch from SPI to DPI, QPI, or OPI mode) 3h – Xpi2Spi (switch from DPI, QPI, or OPI to SPI mode)
waitTimeCfgCommands	012	2	Wait time (in terms of 100 µs) for Device mode configuration command
deviceModeSeq	014	4	Device mode configuration sequence: Byte 0 – Number of required sequences Byte 1 – Sequence index
deviceModeArg	018	4	Argument for Device mode configuration For example, Quad enable setting in Status Register2 for some Quad SPI flash memory devices
configCmdEnable	01C	1	Configuration command enable: 0h – Ignores configuration command 1h – Enables configuration command
configModeType	01D	3	Configuration command type that supports up to three types; each type is combined with a configuration command sequence
configCmdSeqs	020	12	Configuration command that supports up to three sequences
Reserved	02C	4	Reserved
configCmdArgs	030	12	Configuration command that supports up to three arguments

Table continues on the next page...

Table 50. XSPI FCB settings (continued)

Field	Offset (hex)	Size (bytes)	Description
Reserved	03C	4	Reserved
controllerMiscOption	040	4	Miscellaneous controller configuration options: Bit 0 – Differential clock enable: set to 1 for HyperFlash NOR flash memory 1.8 V device and set to 0 for other devices Bit 3 – WordAddressableEnable: set to 1 for HyperFlash NOR flash memory and set to 0 for other devices Bit 4 – SafeConfigFreqEnable: set to 1 if expecting to configure the chip with a safe frequency Bit 6 – DDR mode enable: set to 1 if DDR read is expected Other bits – Reserved; set to 0
deviceType	044	1	Device type 1h – Serial NOR flash memory
sflashPadType	045	1	Data pad used in Read command: 1h – Single pad 2h – Dual pads 4h – Quad pads 8h– Octal pads
serialClkFreq	046	1	Serial clock frequency 1h – 30 MHz 2h – 50 MHz 3h – 60 MHz 4h – 80 MHz 5h – 100 MHz 6h - 120 MHz 7h - 133 MHz 8h - 166 MHz 9h - 200 MHz
lutCustomSeqEnable	047	1	LUT customization enable—required when you cannot perform the program and erase operations using one LUT sequence (currently, only one is applicable to HyperFlash NOR flash memory)
Reserved	048	8	Reserved for future use
sflashA1Size	050	4	Size of flash memory connected to A1

Table continues on the next page...

Table 50. XSPI FCB settings (continued)

Field	Offset (hex)	Size (bytes)	Description
sflashA2Size	054	4	Size of flash memory connected to A2
csPadSettingOverride	060	4	Pad override value for CS pin Use this value (if nonzero) to configure the CS pin; otherwise, use the default boot ROM setting
sclkPadSettingOverride	064	4	Serial clock pad setting override value
dataPadSettingOverride	068	4	Data pad setting override value
dqsPadSettingOverride	06C	4	DQS pad setting override value
timeoutInMs	070	4	Timeout value (in terms of ms) to terminate the busy check
commandInterval	074	4	CS deselect interval between two commands
dataValidTime	078	4	Clock edge to data valid time
busyOffset	07C	2	Busy offset, valid value: 0-1Fh
busyBitPolarity	07E	2	Busy flag polarity: 0h – Busy flag is 1 when the flash memory device is busy 1h – Busy flag is 0 when the flash memory device is busy
lookupTable	080	360	18 LUT sequences—each sequence consists of four words (see XSPI chapter for more information)
lutCustomSeq	1E8	48	Customizable LUT sequences
dllCraSdrVal	218	4	Customizable DLLCRA for SDR setting
smprSdrVal	21C	4	Customizable SMPR SDR setting
dllCraDdrVal	220	4	Customizable DLLCRA for DDR setting
smprDdrVal	224	4	Customizable SMPR DDR setting
pageSize	228	4	Page size of Serial NOR
sectorSize	22C	4	Sector size of Serial NOR
ipcmdSerialClkFreq	230	1	Serial clock frequency for IP commands: 0h – The same with read command Others – The same definition as that of the serial clock frequency
isUniformBlockSize	231	1	Indicates whether the sector size is the same as the block size:

Table continues on the next page...

Table 50. XSPI FCB settings (continued)

Field	Offset (hex)	Size (bytes)	Description
			0h – No 1h – Yes
isDataOrderSwapped	232	1	Data order (D0, D1, D2, D3) is swapped with (D1, D0, D3, D2)
Reserved	233	1	Reserved
serialNorType	234	1	Serial NOR flash memory type: 0, 1, 2, and 3
needExitNoCmdMode	235	1	Exits NoCmd mode before other IP commands
halfClkForNonReadCmd	236	1	Half serial clock for nonread command: true or false
needRestoreNoCmdMode	237	1	Restores NoCmd mode after IP command execution
blockSize	238	4	Flash memory block size
flashStateCtx	23C	4	Flash memory state context
Reserved	240	40	Reserved

9.5.2.1 Reset

During XSPI boot, the boot process requires the XSPI flash memory device to be in a certain mode, specially, 1-bit SPI compatible mode. The flash memory device is, by default, in this mode after a POR reset. That is because the power-up sequence resets it along with the boot ROM device. However, the flash memory device does not remain in 1-bit SPI compatible mode if you configure it in DPI, QPI, or Octal mode when any non-POR resets happen (for example, watchdog timer and external pin reset). In such a case, the boot process requires special processing to restore the flash memory device to 1-bit SPI compatible mode before accessing it. You can achieve this by using a GPIO to assert a reset pin on the flash memory device. The bootloader can perform the reset process and reset the flash memory device to 1-bit SPI compatible mode if the XSPI_RESET_PIN_EN field is a nonzero value in fuse, using the GPIO that the XSPI_RESET_PIN_PORT and XSPI_RESET_PIN_GPIO fields specify in combination.

NOTE

If using FCB mode and the FLASH is configured to QPI/OPI mode, you should always program the FLASH state context to the correct value in order to restore the FLASH to SPI mode after warm reset.

You can also achieve this by maintaining the FLASH state context via the FLASH context register (General Purpose (GPR) register in RTC_SS, address 4006_8C10h) or FCB flashStateCtx (See Table 50), which logs the below information:

- FLASH default working mode on POR reset
- Current FLASH working mode
- Sequence to reset to default working mode

Before accessing the FLASH device, ROM reads the context parameter and resets the FLASH to the default working mode via the specified reset pattern first. The FLASH device is in the default operating mode, and consequently, software can access FLASH as expected (including rebooting, configure-memory command, or the FLASH API call).

The following section explains the detailed information about the FLASH context register.

```
typedef union
{
  struct
```

```
  {
    uint8_t reserved ;
    uint8_t current_mode;
    uint8_t reserved ;
    uint8_t restore_sequence;
  } B;
  uint32_t U;
} FLASH_run_context_t;
```

Table 51. Flash context register

Field	Description	Definition
current_mode	FLASH Device current running mode	kFlashInstMode_ExtendedSpi = 0x0 kFlashInstMode_0_4_4_SDR = 0x1 kFlashInstMode_0_4_4_DDR = 0x2 kFlashInstMode_DPI_SDR = 0x21 kFlashInstMode_DPI_DDR = 0x22 kFlashInstMode_QPI_SDR = 0x41 kFlashInstMode_QPI_DDR = 0x42 kFlashInstMode_OPI_SDR = 0x81 kFlashInstMode_OPI_DDR = 0x82
restore_sequence	FLASH reset sequence of standard SPI mode	kRestoreSeqence_None = 0 kRestoreSeqence_QPI_4_0xFFs = 2 kRestoreSeqence_QPI_Mode_0x00 = 3 kRestoreSeqence_Send_F0 = 5 kRestoreSeqence_Send_66_99 = 6 kRestoreSeqence_Send_6699_9966 = 7 kRestoreSeqence_Send_06_FF = 8 kRestoreSeqence_QPI_5_0xFFs = 9 kRestoreSeqence_Send_QPI_8_0xFFs = 10 kRestoreSeqence_Wakep_0xAB = 11 kRestoreSeqence_Wakep_0xAB_54 = 12

Take IMXRT700 EVK board for example, if the user needs to reset the FLASH to XSPI mode, such as waking up the FLASH from deep power-down mode, or reconfigure the FLASH use ISP command, or use the ROM auto probe boot feature(when the FLASH is in OPI DDR mode), the user needs to do below actions.

FLASH Type: MX25UM51345G:

```
FLASH_run_context_t run_ctx = {.U = 0 };
run_ctx.B.current_mode = kFLASHInstMode_OPI_DDR; // Set the current FLASH mode
run_ctx.B.restore_sequence = kRestoreSeqence_Send_6699_9966; // Select the FLASH reset sequences
*(volatile uint32_t *)(0x40068C10) = run_ctx.U // Update the context register
```

After this operation, the context register stores the current running mode of the NOR FLASH. In ISP reconfiguring or rebooting, ROM uses the context value to decides the way to reset the FLASH. For the FLASH resetting sequence, please refer to the flash device data sheet for Send_66_99, Send_6699_9966, Send_06_FF.

9.5.2.2 External Memory Configuration Data (XMCD)

For comprehensive or feature-rich applications, the on-chip RAM may not be enough for the application use. Usually, such an application requires a large capacity of RAM and needs to run on the external RAM for high execution performance. However, enabling the external RAM on the chip is relatively complicated, and the settings are chip specific. The Boot ROM on the chip simplifies the external RAM configuration by introducing the XMCD.

The XMCD data structure resides out of the program image, and the offset is fixed to 300h.

XMCD consists of a XMCD header and a memory specific configuration block. On this chip, the following type of external RAM devices is supported and it has a specific configuration block.

- HyperRAM/APMemory via XSPI – XSPI RAM configuration block

9.5.2.2.1 XMCD Header

The header definition is as follows.

Table 52. XMCD Header Definition

[31:28] Tag	[27:24] Version	[23:20] Memory Interface	[19:16] Instance	[15:12] Configuration Block Type	[11:0] Configuration block size
Valid value = Ch	Fixed value of '0'	0 - XSPI	0 - XSPI0 1 - XSPI1	0 - Simplified Configuration Option Block 1 - Full Configuration Block	Configuration block size (including the XMCD Header itself).

9.5.2.2.2 XSPI RAM simplified configuration option clock

The XMCD offers a simplified XSPI RAM configuration option block which can meet the typical usage of the pSRAM (HyperRAM or APMemory) devices.

9.5.2.2.2.1 Simplified XSPI RAM Configuration Option Structure

The simplified XSPI RAM configuration option structure is shown in the following tables.

Table 53. Simplified XSPI RAM Configuration Option 0

[31:28] Tag	[27:24] Option Size	[23:20] device type	[19:16] Reserved	[15:12] Misc.	[11:8] Maximum frequency	[7:0] Size in MB
Fixed to Ch	0 - Option words = 1 1 - Option words = 2 Others - Reserved	0 - HyperRAM 1 - APMemory	Reserved	For HyperRAM 0 - Differential clock driven 1 - Single-ended clock driven	0 - Invalid 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz	0 - Auto detection Others - Size in MB

Table continues on the next page...

Table 53. Simplified XSPI RAM Configuration Option 0

[31:28] Tag	[27:24] Option Size	[23:20] device type	[19:16] Reserved	[15:12] Misc.	[11:8] Maximum frequency	[7:0] Size in MB
					5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz Others - Reserved	

Table 54. Simplified XSPI RAM Configuration Option 1

[31:28] RAM_CONNECTION	[27:24] Reserved	[23:20] Reserved	[19:16] Reserved	[15] Bus width	[14:8] Maximum CS Low time	[7:4] Write dummy cycles	[3:0] Read dummy cycles
0 - PORTA	Reserved	Reserved	Reserved	• 0 - X8 mode • 1 - X16 mode	Max CS low interval = value x 0.1 µs	0 - Auto detection Others - Specified dummy cycles	0 - Auto detection Others - Specified dummy cycles

9.5.2.2.2.2 Example of XMCD for HyperRAM Support

The following table shows an example of XMCD for HyperRAM support.

Table 55. Example XMCD block for HyperRAM

Offset	Field	Description
0h	C001_0008h	Tag = Ch Version = 0 Memory Interface: XSPI Instance: 1 - XSPI1 Configuration block type: Simplified Configuration block size: 8 (4-byte header + 4-byte option block, option word1 not used)
4h	C000_0700h	Tag = Ch Option_Size = 0 DeviceType: HyperRAM Reserved: 0

Table continues on the next page...

Table 55. Example XMCD block for HyperRAM (continued)

Offset	Field	Description
		Misc: 0 = differential clock Max Freq: 7 = 133 MHz Memory Size: Auto-detection

9.5.3 Program XSPI NOR using ISP

The boot ROM supports the booting of the image from an external XSPI NOR flash memory device. This section describes how to use the blhost tool to program the image into an external nonvolatile memory for booting. The blhost tool uses supported serial peripherals to communicate with the ROM code via ROM ISP mode. See In-System Programming.

See Boot interface pin allocation for XSPI pin assignments of NOR flash memory connections.

9.5.3.1 Configuring the XSPI NOR flash memory device

9.5.3.1.1 Configuration options, settings, and parameters

9.5.3.1.1.1 Options

Table 56. XSPI serial NOR flash memory configuration options

Offset	Field	Description
0	Option0	See Table 57 for details.
4	Option1	Optional; effective only if the "option_size" field in option0 configuration is a non-zero value. See Table 58 for more information.

9.5.3.1.1.2 Option 0 field descriptions

Table 57. Option 0 field descriptions

Field	Bits	Description
tag	31:28	Tag of the configuration option, fixed to Ch
option_size	27:24	Size in bytes = (option size + 1) × 4 Set to 0 if only option0 is used.
device_type	23:20	Device detection type: 0 – Read SFDP for SDR commands 1 – Read SFDP for DDR read commands 2 – Reserved 3 – HyperFlash 3V 4 – Macronix Octal DDR 6 – Micron Octal DDR

Table continues on the next page...

Table 57. Option 0 field descriptions (continued)

Field	Bits	Description
		8 – Adesto EcoXiP DDR
query_pad	19:16	Data pads during Query command: (read SFDP or read MID) 0 – 1 2 – 4 3 – 8
cmd_pad	15:12	Data pads during flash memory access command: 0 – 1 2 – 4 3 – 8
quad_mode_setting	11:8	Quad SPI mode enable setting: 0 – Not configured 1 – Set bit 6 in Status register 1 2 – Set bit 1 in Status register 2 3 – Set bit 7 in Status register 2 4 – Set bit 1 in Status register 2 by using the 0x31 command **NOTE** This field is effective only if the chip is compliant with JESD216 (9 longword SFDP table).
misc_mode	7:4	Miscellaneous mode setting: 0 – Disabled 1 – Enable 0-4-4 mode for high random read performance 3 – Data Order Swapped mode (for MXIC octal flash memory only)
max_freq	3:0	Maximum flash memory operation speed: For SDR: 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 80 MHz (default) 5 – 100 MHz 6 - 120 MHz 7 - 133 MHz

Table continues on the next page...

Table 57. Option 0 field descriptions (continued)

Field	Bits	Description
		8 – 166 MHz 9 – 200 MHz

9.5.3.1.1.3 Option 1 field descriptions

Table 58. Option 1 field descriptions

Field	Bits	Description
Reserved	31:8	Reserved for future use
dummy_cycles	7:0	Dummy cycles for the Read command: 0 – Use detected dummy cycle Others – Use dummy cycles provided in the flash memory data sheet

9.5.3.1.1.4 Option settings

The following table shows typical XSPI serial NOR flash memory configuration option settings.

Table 59. XSPI serial NOR flash memory configuration option settings

XSPI mode name	Access type	Setting
Quad SPI NOR	Quad SPI SDR read	Option0 = C000_0002h (50 MHz)
Quad SPI NOR	Quad SPI DDR read	Option0 = C010_0002h (50 MHz)
HyperFlash 3V0	—	Option0 = C033_3002h (50 MHz)
MXIC OPI DDR (OPI DDR enabled by default)	—	C043_3001h (30 MHz)
Micron Octal DDR	—	Option0 = C060_0002h (50 MHz)
Micron OPI DDR	—	Option0 = C060_3002h (50 MHz)
Micron OPI DDR (DDR read enabled by default)	—	Option0 = C063_3002h (50 MHz)
Adesto OPI DDR	—	Option0 = C080_3002h (50 MHz)

9.5.3.1.2 Configuring, erasing, and programming via blhost

9.5.3.1.2.1 Configure via blhost

To configure a XSPI NOR flash memory device, the configuration parameter must be stored in RAM first. Consider Macronix MX25L25645G, which connects to the XSPI interface, as an example. The following figure shows the configuration parameter for XSPI, which is C000_0001h. You select a configuration parameter according to the flash memory type.

```
C:\blhost>blhost.exe -p com4 fill-memory 0x2001f000 4 0xc0000001
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
```

```
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

Code Listing 1. Setting the XSPI configuration parameter in RAM

9.5.3.1.2.2 Read, erase, and program via blhost

Use the configuration parameter stored in RAM (see Configure via blhost) to configure XSPI, as shown in the following figure. After that, you can read, erase, and program the flash memory. For details related to blhost tool commands, use the help section of the blhost tool. The memory id of XSPI is Bh, The starting address of image in XSPI0 is 2800_4000h and the starting address of image in XSPI1 is 0800_4000h.

```
C:\blhost>blhost.exe -p com4 configure-memory 0xb 0x2001f000
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

Code Listing 2. Configuring XSPI using the parameter stored in RAM

```
C:\blhost\blhost>blhost.exe -p com3 read-memory 0x08004000 0x100
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
(1/1) 100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 256 (0x100)
Read 256 of 256 bytes.
```

Code Listing 3. Reading the NOR flash memory via the blhost tool

```
C:\blhost\blhost>blhost.exe -p com3 flash-erase-region 0x08004000 0x200
Ping responded in 1 attempt(s)
Inject command 'flash-erase-region'
```

```
Successful generic response to command 'flash-erase-region'
Response status = 0 (0x0) Success.
```

Code Listing 4. Erasing the NOR flash memory via the blhost tool

```
C:\blhost\blhost>blhost.exe -p com3 write-memory 0x08004000 boot_image.bin
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 5796 (0x16a4) bytes to the target.
Successful generic response to command 'write-memory'
(1/1) 100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 5796 of 5796 bytes.
```

Code Listing 5. Programming the boot image into NOR flash memory via the blhost tool

9.5.3.1.2.3 Generate and program FCB via blhost

Generate and program the XSPI NOR FCB into the flash memory for XSPI boot, with an FCB of 768 bytes at offset 0h. You use this FCB to configure the XSPI interface when booting the image from an external NOR flash memory via the XSPI interface. You can generate the FCB using the previous XSPI configuration parameter (C000_0002h). Store the configured FCB and program parameter into RAM (see Code Listing 6). The boot ROM uses these in the next step to automatically generate and program the FCB into flash memory at offset 0800_0000h (see Code Listing 7).

```
C:\blhost>blhost.exe -p com4 fill-memory 0x2000f000 4 0xf000000f
Ping responded in 1 attempt(s)
Inject command 'fill-memory'
Successful generic response to command 'fill-memory'
Response status = 0 (0x0) Success.
```

Code Listing 6. Storing the configured FCB parameter into RAM

```
C:\blhost>blhost.exe -p com4 configure-memory 0x9 0x2000f000
Ping responded in 1 attempt(s)
Inject command 'configure-memory'
Successful generic response to command 'configure-memory'
Response status = 0 (0x0) Success.
```

Code Listing 7. FCB generation and programming into the flash memory at offset 0800_0000h

```
C:\blhost\blhost>blhost.exe -p com3 read-memory 0x08000000 0x100
Ping responded in 1 attempt(s)
Inject command 'read-memory'
Successful response to command 'read-memory'
46 43 46 42 00 04 01 56 00 00 00 00 01 03 03 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 01 04 02 00 00 00 00 00 00 00 00 00
00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ec 04 20 0a 00 1e 04 32 04 26 00 00 00 00 00 00
```

```
05 04 04 24 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
21 04 20 08 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(1/1) 100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 256 (0x100)
Read 256 of 256 bytes.
```

Code Listing 8. Reading the FCB back via the blhost tool

9.5.3.1.3 Typical configuration parameters

You require different configuration parameters for different XSPI NOR flash memory types. The following table shows some typical flash memory configuration parameters.

Table 60. Typical NOR flash memory configuration parameters

Device vendor	Device type	Flash memory type	Configuration parameter	Flash memory mode	Comment
Adesto	Octal SPI flash memory	ATXP032	C080_3001h	DDR	ATXP032 has three versions, REV A, REV B, and REV C. For REV A, the boot ROM only supports auto probe boot. You must configure XSPI_CFG0[3:0] as 7h, with no need to configure FCB.
Cypress	HyperFlash memory	S26KS512S	C023_3001h	DDR	For HyperFlash boot, you must configure XSPI_CFG0[XSPI_FLASH_TYPE] as 010b before boot.
Macronix	Quad SPI flash memory	MX25L25645G	C000_0001h	SDR	—
Macronix	Quad SPI flash memory	MX25L25645G	C010_0001h	DDR	—
Macronix	Octal SPI flash memory	MX25UM51345G	C040_3001h	DDR	—
Micron	Octal SPI flash memory	MT35XL512ABA	C060_3001h	DDR	—

9.5.3.2 Programming the boot image into the flash memory

The following table lists the procedure and corresponding commands to program the boot image into the external flash memory with all data encrypted (see Configuring the XSPI NOR flash memory device for more information).

Table 61. XSPI NOR flash memory configuration

Step number	Step	Command to perform the step
1	Fill in the flash configuration parameter.	`blhost.exe -p com4 fill-memory 0x20020000 4 0xc0000001`
2	Enable the flash memory.	`blhost.exe -p com4 configure-memory 0xb 0x20020000`
3	Fill in the FCB generation parameters.	`blhost.exe -p com4 fill-memory 0x20020004 4 0xf000000f`
4	Program FCB at flash memory offset 0h.	`blhost.exe -p com4 configure-memory 0xb 0x20020004`
5	Erase the flash memory region from 0800_4000h to 0801_4000h.	`blhost.exe -p com4 flash-erase-region 0x08004000 0x10000`
6	Program the boot image into the flash memory from offset 0800_4000h.	`blhost.exe -p com4 write-memory 0x08004000 image.bin`

9.5.4 Dual image boot

XSPI supports dual image boot. For dual image boot, Boot ROM first tries to boot the image with higher image version. If boot fails, the boot ROM tries to boot the other image. If the boot still fails, it will go to recovery boot if REC_BOOT_EN is enabled. Before using this feature, you need to configure the XSPI_FLASH_SEC_IMAGE_OFFSET. The value of XSPI_FLASH_SEC_IMAGE_OFFSET is the offset of the second image (converted to KB) divided by 128.

In another word,

- Write image0 to the offset address of 4000h (default offset address) of XSPI flash:
  - For XSPI0 NOR, write image0 to 2800_4000h
  - For XSPI1 NOR, write image0 to 0800_4000h
- write image1 to the offset address of XSPI_FLASH_SEC_IMAGE_OFFSET × 128 KB of XSPI flash:
  - For XSPI0 NOR, write image1 to 2800_0000h + XSPI_FLASH_SEC_IMAGE_OFFSET × 128 KB
  - For XSPI1 NOR, write image1 to 0800_0000h + XSPI_FLASH_SEC_IMAGE_OFFSET × 128 KB.

Take booting from XSPI0 as an example. Set the XSPI_FLASH_SEC_IMAGE_OFFSET to 1, then image1 is at 2802_0000h. Commands below show the steps to program those two images.

```
// Initialize XSPI
C:\blhost>blhost.exe -p com4 -- fill-memory 0x20070000 0x4 0xc0000007
C:\blhost>blhost.exe -p com4 -- fill-memory 0x20070004 0x4 0x00000000
C:\blhost>blhost.exe -p com4 -- configure-memory 0xb 0x20070000
// Erase a piece of memory to write two images
C:\blhost>blhost.exe -p com4 -- flash-erase-region 0x28000000 0x4000 0xb
C:\blhost>blhost.exe -p com4 -- flash-erase-region 0x28004000 0x4000 0xb
C:\blhost>blhost.exe -p com4 -- flash-erase-region 0x28020000 0x4000 0xb
// Write image0
C:\blhost>blhost.exe -p com4 -- write-memory 0x28004000 boot_image0.bin 0xb
// Write image1
C:\blhost>blhost.exe -p com4 -- write-memory 0x28020000 boot_image1.bin 0xb
```

9.5.4.1 Basic remapping function

Boot ROM supports dual image boot by using the MMUs with XSPI. It allows user to use same linker file for the two images and keep both images on the XSPI NOR with different versions. Boot ROM boots the image with higher version. If the version of image0 located at offset 4000h (default offset) is lower than image1 located at higher offset, Boot ROM will remap the whole image1 (size specified by fuse field of XSPI_FLASH_IMAGE_SIZE in fuse word 153) to offset 4000h and boot the image. Figure below shows the XSPI NOR layout before remapping (left) and after remapping (right).

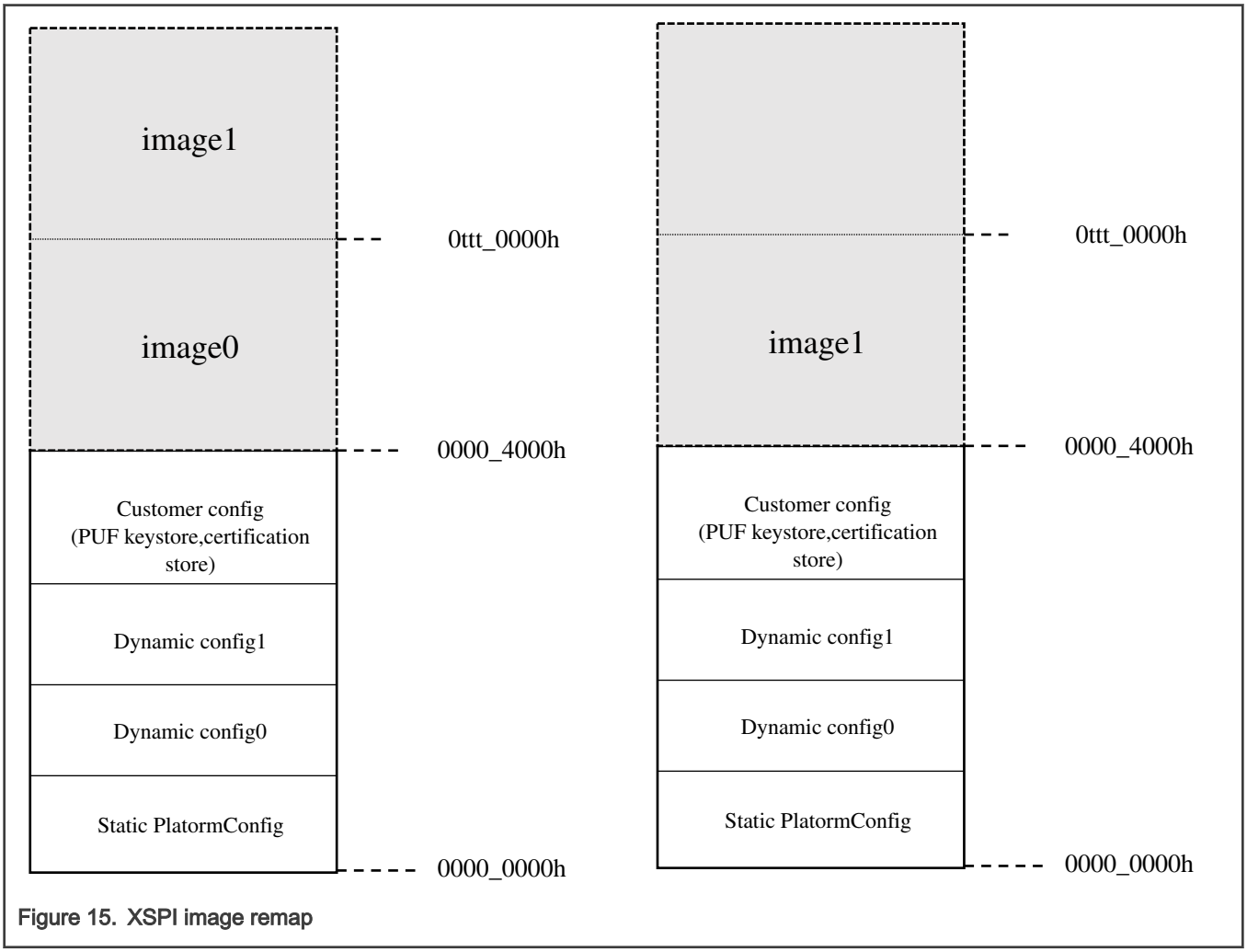


Figure 15. XSPI image remap

9.5.4.2 Image remap offset and size configuration

The configuration related to XSPI dual-image boot is in XSPI_CFG[1]. See the following table for more information.

Table 62. Boot image 1 remap offset (XSPI_CFG[1])

Name that is filled	Bit offset	Width	Value	Description
XSPI_FLASH_SEC_IMAGE_OFFSET (second image offset based on XSPI start address of 2800_0000h or 0800_0000h)	7	10	*x*	*x* multiplied by 128 KB

Table 63. Boot Image size (XSPI_CFG[1])

Name that is filled	Bit offset	Width	Value	Description
XSPI_FLASH_IMAGE_SIZE	17	7	0	Same size as the image offset (XSPI_IMAGE_OFFSET)
			1	128 KB
			2	256 KB
			m	m × 128 KB (m=3-7)
			8	1 MB
			n	(n-7) × 1 MB (n=8-71)

NOTE

XSPI remap size cannot exceed the start address of boot image1.

9.5.4.3 Dual image boot flow

Figure below shows the dual image boot flow done by Boot ROM.

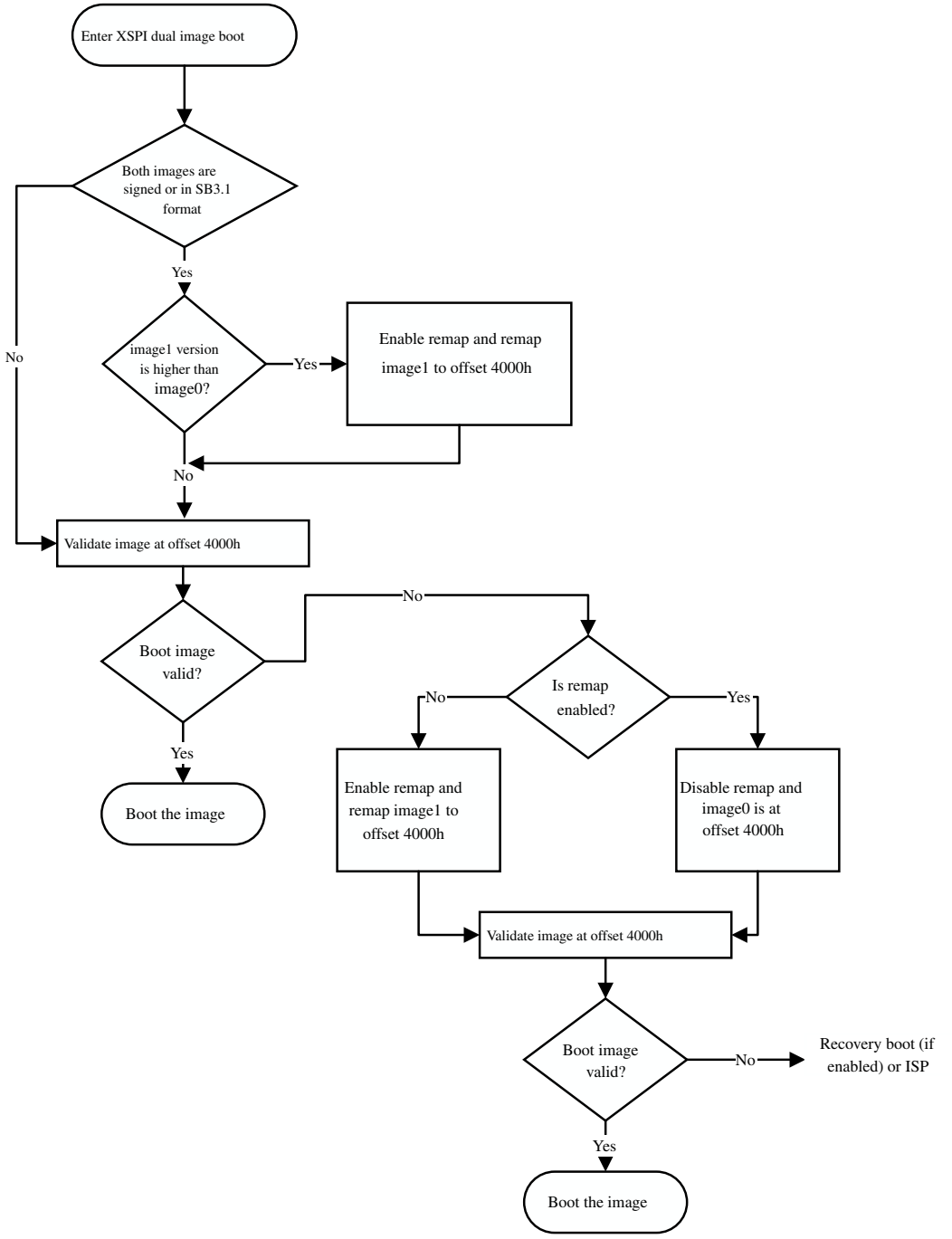


Figure 16. Dual image boot flow

9.6 Prime boot from eMMC

Boot ROM supports prime boot from eMMC connected to uSDHC instance 0 or 1. Load-to-RAM boot image is supported for eMMC. Boot ROM supports load the boot image from eMMC to on-chip SRAM (starts from offset 15000h) and XSPI pSRAM to boot.

At Develop (3h) lifecycle, load-to-RAM image can be any image with valid image header. When secure boot is enforced or lifecycle is after Develop (3h), the load-to-RAM image must be in SB3 format and the image to generate SB3 should either be CRC image or signed image.

You can start eMMC boot by:

- BOOT_SRC[1:0] = 00b and BOOT_PIN_MODE = 00b, uSDHC0 will be selected as boot source
- BOOT_SRC[1:0] = 10b, Boot ROM will directly boot from uSDHC flash, and uSDHC instance is selected via BOOT_IF_INST

Once eMMC boot is selected, Boot ROM tries to find the boot image from offset 1000h of the eMMC. XMCD is also supported by eMMC primary boot. XMCD for eMMC boot should be programed at offset 300h.

9.6.1 eMMC boot configuration

The eMMC boot can be performed using the uSDHC ports. Selected uSDHC port can be configured using fuses from fuse word 158 and 159 for different conditions. See table below for details:

Table 64. eMMC boot configuration

FUSE_WORD	Bits	Field name	Shipped value	Settings
158 SDHC_CFG0 for uSDHC0 159 SDHC_CFG1 for uSDHC1	1:0	SDHC_BUS_WIDTH	00	eMMC bus width selection • 00 - 4 bits • 01 – 8 bits • 10 – 4 bits DDR • 11 – 8 bits DDR
	2	SDHC_PWR_CYCLE_EN	0	Power cycle enable • 0 – No power cycle • 1 – Power cycle enabled via RST pad
	4:3	SDHC_PWR_CYCLE_WAIT	0	Power down wait selection • 00 – 20 ms • 01 – 10 ms • 10 – 5 ms • 11 – 2.5 ms
	6:5	SDHC_PWR_STABLE_WAIT	0	Power up wait selection • 00 – 5 ms • 01 – 2.5 ms • 10 – Reserved • 11 - Reserved
	7	SDHC_PWR_POL	0	uSDHC reset polarity selection • 0 – Reset active low • 1 – Reset active high
	8	Reserved	0	—

Table continues on the next page...

Table 64. eMMC boot configuration (continued)

FUSE_WORD	Bits	Field name	Shipped value	Settings
	9	SDHC_RESET_PRE_IDLE	0	eMMC disable pre-idle selection • 0 – Enable • 1 – Disable
	10	SDHC_MMC_SPEED	0	eMMC speed selection • 0 – Normal speed • 1 – High speed
	31:11	Reserved	0	—

The boot code support standards: eMMCv4.4 or higher.

9.6.2 eMMC boot flow

During the initialization, the eMMC frequency is set to 400 kHz. When the eMMC card enters the identification portion of the initialization, the voltage validation is performed, and Boot ROM checks the high-voltage settings and the card capacity. Boot ROM supports both the high-capacity and low-capacity eMMC cards. The boot partition can be selected for an MMC4.x card after the card initialization is complete.

To boot from eMMC card, it is mandatory to take below actions for a successful eMMC boot:

1. Configure eMMC card with configuration words at ISP mode
2. Program image to eMMC card at offset 1000h
3. Program XMCD at offset 300h if target load address is pSRAM connected to XSPI. For detailed XMCD configuration, see External Memory Configuration Data (XMCD)
4. Program eMMC boot fuse configuration and reset to enter eMMC boot via related configurations. See Prime boot from eMMC for methods to enter eMMC boot.

9.6.3 Program eMMC card using ISP

Boot ROM has implemented In-System-Programming (ISP) to enable programming eMMC flash via available peripherals and thus can be utilized to configure and program eMMC card.

For configuration of eMMC card, fill-memory with configuration words and configure-memory command can be used. Boot ROM will always store the configuration words to RAM with fill-memory command, and then apply configuration to configure the memory device via configure-memory. For eMMC configuration words when ISP mode, detailed definition is described as below.

Table 65. eMMC configuration block definition

Word index	Bits	Name	Description
Word0	[31:28]	TAG	eMMC configuration block tag used to mark if the block is valid or not • Ch: Valid block • Others: Invalid

Table continues on the next page...

Table 65. eMMC configuration block definition (continued)

Word index	Bits	Name	Description
	[27]	Reserved	—
	[26:24]	PARTITION_ACCESS	Select eMMC partition where the image or data write to • 0h: User data area • 1h: Boot partition 1 • 2h: Boot partition 2 • 3h: RPMB • 4h: General Purpose partition 1 • 5h: General Purpose partition 2 • 6h: General Purpose partition 3 • 7h: General Purpose partition 4
	[23:16]	Reserved	—
	[15:12]	TIMING_INTERFACE	Select the bus timing when accesses eMMC memory • 0h: Normal • 1h: HS • 2h-Fh: Reserved
	[11:8]	BUS_WIDTH	Select the bus width when accesses eMMC memory • 0h: x1 SDR • 1h: x4 SDR • 2h: x8 SDR • 3h-4: Reserved • 5h: x4 DDR • 6h: x8 DDR • 7h-Fh: Reserved
	[7:0]	Reserved	—
Word1	[31:26]	Reserved	—
	[25:24]	PWR_DOWN_TIME	eMMC power down delay time before power up the eMMC card Only valid when PWR_ CYCLE_ENABLE is enabled • 0h: 20 ms • 1h: 10 ms • 2h: 5 ms

Table continues on the next page...

Table 65. eMMC configuration block definition (continued)

Word index	Bits	Name	Description
			• 3h: 2.5 ms
	[23]	PWR_POLARITY	eMMC power control polarity. Only valid when PWR_CYCLE_ENABLE is enabled • 0h: Power down when uSDHC.RST set low • 1h: Power down when uSDHC.RST set high
	[22:21]	Reserved	—
	[20]	PWR_UP_TIME	eMMC power up delay time to wait voltage regulator output stable Only valid when PWR_CYCLE_ENABLE is enabled • 0h: 5 ms • 1h: 2.5 ms
	[19]	PWR_CYCLE_ ENABLE	Execute a power cycle before starting the SD initialization progress • 0h: Disable • 1h: Enable
	[18:4]	Reserved	—
	[3:0]	USHDC_INSTANCE	Select uSDHC instance as interface • 0h: uSDHC0 • 1h: uSDHC1 • Others: Reserved

Once eMMC card is correctly configured, Write-memory can be used to program boot image to eMMC card. Memory id of eMMC is 121h, and the offset of image is 1000h.

Below is an example of configuring eMMC boot image:

```
//Initialize EMMC
blhost -p com4 -- fill-memory 0x20070000 0x4 0xc0000000
blhost -p com4 -- fill-memory 0x20070004 0x4 0x00000000
blhost -p com4 -- configure-memory 0x121 0x20070000
//Erase memory
blhost -p com4 -- flash-erase-region 0x0 0x4000 0x121
//Write image to EMMC
blhost -p com4 -- write-memory 0x1000 boot_image.bin 0x121
```

9.7 Boot from serial NAND (XSPI)

BootROM supports load-to-RAM (load to SRAM and XSPI pSRAM are both supported) boot from serial NAND connected on XSPI0 or XSPI1. The boot ROM supports several Serial NAND Flash devices from different vendors.

You can start XSPI NAND boot by:

- BOOT_SRC[1:0] = 11b, Boot ROM will directly boot from XSPI NAND flash, and XSPI instance is selected via BOOT_IF_INST.
- ROM API bootloader entry path by setting the valid tag, prime boot mode, and the XSPI NAND interface and with the specific instance, Boot ROM will directly boot from XSPI NAND flash.

9.7.1 Serial NAND eFUSE configuration

The boot ROM determines the configuration of external serial NAND flash by parameters, which are provided by eFUSE during boot. See below table for parameters details.

Table 66. eFuse definition for serial NAND over XSPI

Fuse	Definition	Settings	Shipped Value
XSPI_CFG0[18:16]	XSPI_RESET_PIN_PORT	GPIO port. This field is valid only if XSPI_RESET_PIN_EN is set.	0
XSPI_CFG0[23:19]	XSPI_RESET_PIN_GPIO	GPIO pin number. This field is valid only if XSPI_RESET_PIN_EN is set.	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND)[0]	XSPI_NAND_SEARCH_CNT	0b - Search at most 1 block (for BCB) 1b - Search at most 2 blocks	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [1]	XSPI_NAND_COL_ADDR_WIDTH	0b - 11 bits (for 2 KB-page devices) 1b - 12 bits (for 4 KB-page devices)	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [3:2]	XSPI_NAND_CS_INTERVAL	CS de-asserted interval between two commands 00b - 100 ns 01b - 200 ns 10b - 400 ns 11b - 50 ns	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [5:4]	XSPI_NAND_SEARCH_STRIDE	BCB search step/stride, aligned with block size 00b - 64 pages 01b - 128 pages 10b - 256 pages 11b - 32 pages	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [7:6]	XSPI_NAND_HOLD_TIME	Hold time before interaction with NAND 00b - Wait until NAND is idle (by polling NAND's status register) 01b - 500 µs 10b - 1ms 11b - 3 ms	0

Table continues on the next page...

Table 66. eFuse definition for serial NAND over XSPI (continued)

Fuse	Definition	Settings	Shipped Value
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [9:8]	Reserved	—	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [11:10]	XSPI_NAND_RESET_TYPE	00b - No reset before interaction with NAND 01b - Reset the NAND with FFh 10b - Reset the NAND with 66h, 99h 11b - Reset the NAND with dedicated Reset Pin	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [15:12]	Reserved	—	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [18:16]	XSPI_NAND_FREQ	000b - 80 MHz 001b - 100 MHz 010b - 120 MHz 011b - 133 MHz 100b - 166 MHz 101b - 200 MHz 110b - 50 MHz 111b - 30 MHz	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [20:19]	Reserved	—	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [22:21]	XSPI_PAGES_PER_BLOCK	00b - 64 pages 01b - 128 pages 10b - 256 pages 11b - 32 pages	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [23]	Reserved	—	0
OTP_XSPI_NAND_CFG9 (XSPI_NAND) [31:24]	XSPI_FEATURE_CONFIG	Used for loading the arguments/parameters for device configuration commands. bit 0 - For Micron memory, it must be configured as 0b to disable continue read mode; for Macronix or Winbond memory, it must be configured as 1b to enable quad mode. bit 3 - For Winbond memory, it must be configured as 1b to enable buffer mode. bit 4 - Must be configured as 1b to enable error code correction.	0

Table continues on the next page...

Table 66. eFuse definition for serial NAND over XSPI (continued)

Fuse	Definition	Settings	Shipped Value
		Others - Reserved	

NOTE

All the fuse fields in BOOT_CFG9 (XSPI_NAND) are there for FCB derivation stage. After successful FCB derivation, XSPI is re-configured as per FCB.

9.7.2 XSPI NAND flash boot flow and boot control blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As part of the NAND media initialization, the ROM driver uses safe NAND timings to search for a Firmware Configuration Block (FCB) that contains the optimum NAND timings, page address of Discovered Bad Block Table (DBBT) Search Area and start page address of primary and secondary firmware.

The built-in hardware ECC in serial NAND device is used during data read, the FCB data structure is protected using CRC checksum. The driver reads raw 1024 bytes of the first sector and runs through CRC check that determines whether FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments page number to Search Stride number of pages to read for the next BCB until SearchCount pages have been read.

If search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM falls back to the main boot flow (e.g., turn to recovery boot).

The FCB contains the page address of DBBT Search Area, and the page address for primary and secondary boot images. ROM searches DBBT Search Area for DBBT, resembling the search for FCB. After the FCB is derived, the DBBT is loaded, and the primary or secondary boot image is loaded using the starting page address from FCB.

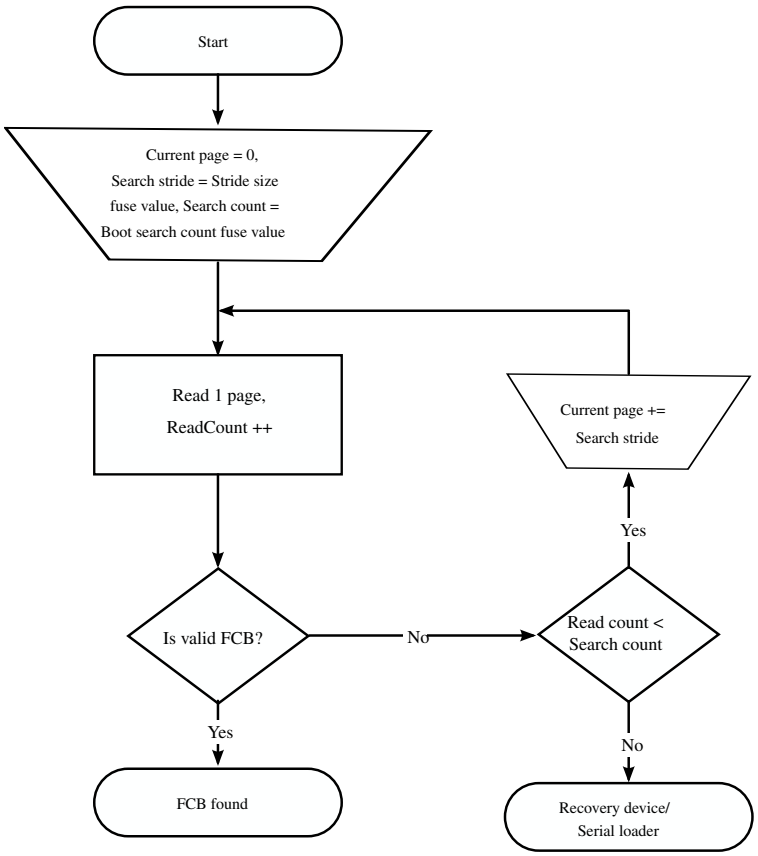


Figure 17. FCB search flow

After FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If DBBTSearchStartPage is 0 in the FCB, then ROM assumes that there are no bad blocks in the NAND device boot area. See figure below for the DBBT search flow.

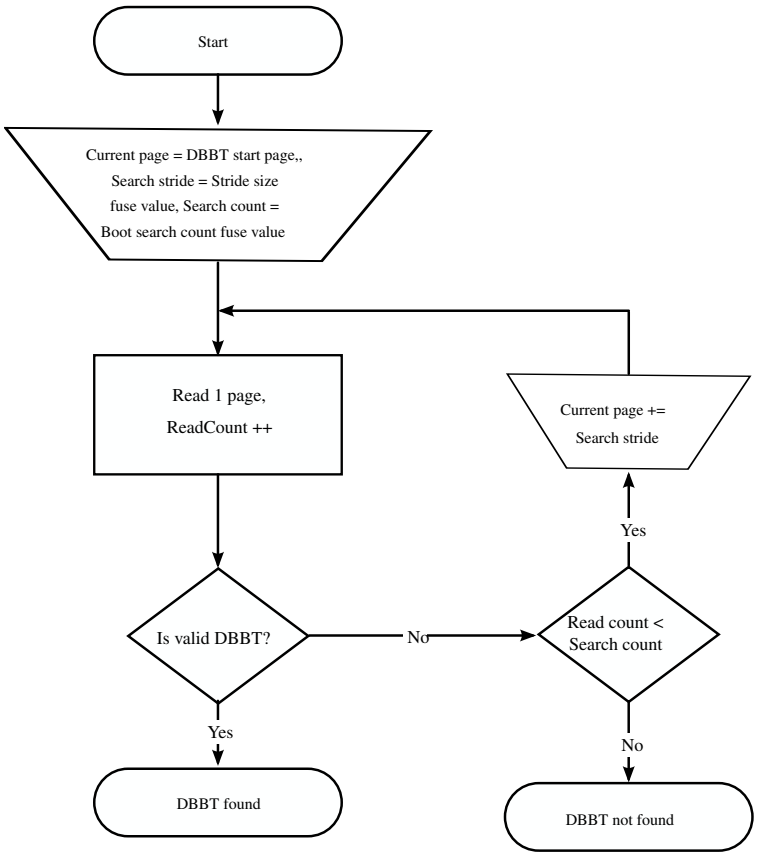


Figure 18. DBBT search flow

After finding the valid FCB data, if the FCB has the valid XMCD data, the device will configure the external pSRAM memory based on the XMCD configuration, and then load the NAND image into the pSRAM region and boot the NAND image.

If during primary image read a page has more errors than ECC can correct or other boot fails, the boot ROM will try the redundant boot hander flow and set the API Entry register SYSCON_S0->ROM_SCRATCH1_POR with increment boot_image_index and try a redundant image boot. The redundant boot number depends on the firmwareCopies value in the FCB, call the redundant boot hander flow until the NAND image boot pass. If all the redundant boots fail, the boot ROM goes back to main boot flow (Recovery Device boot or Serial Loader Flow).

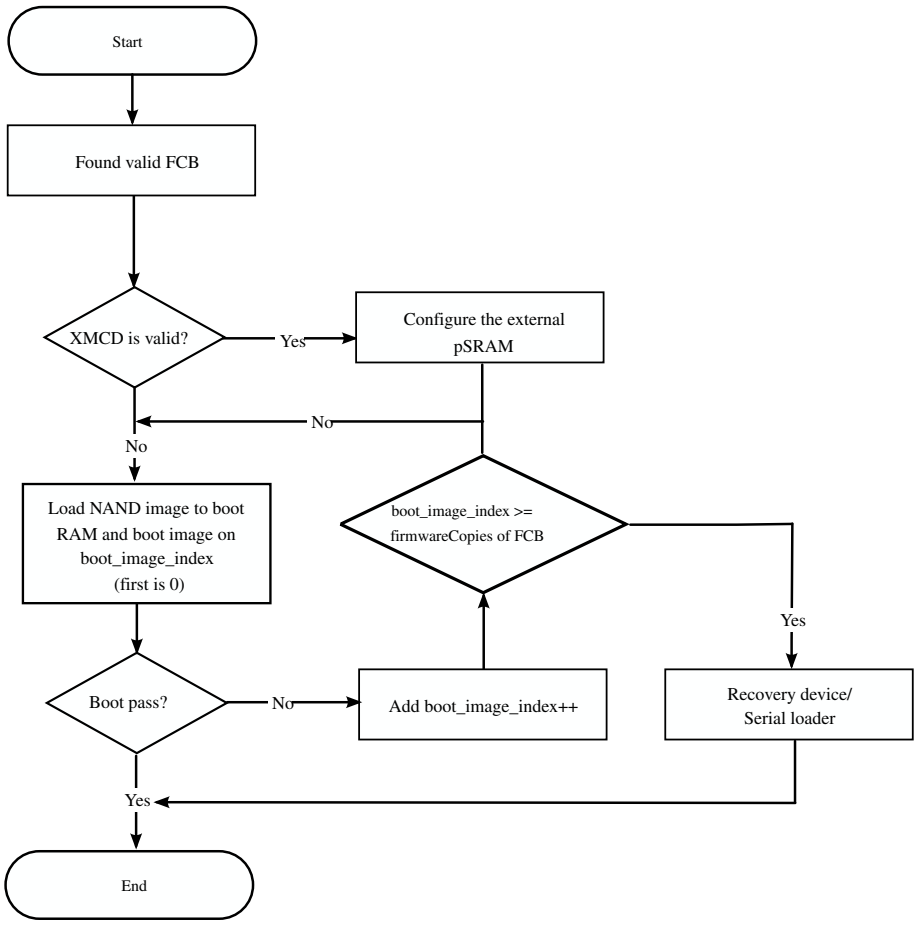


Figure 19. NAND image boot flow

9.7.3 Firmware Configuration Block (FCB)

The Firmware Configuration Block (FCB) is at the first page in the good block which is searched by searchStride (XSPI_NAND_SEARCH_STRIDE in the efuse region) step. The FCB should be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance. When the first serial NAND block becomes corrupted, ROM will find its copy in the next serial NAND block. The search area should span over at least two serial NAND blocks. The location information for DBBT search area and images are all specified in the FCB. The following table shows the flash control block structure.

Table 67. Flash control block structure

Name	Offset	Size bytes	Description
crcChecksum	000h	4	Checksum. This is calculated with an MPEG2 variant of CRC-32. See Table 68 for more details. it calculates from fingerprint to the end of FCB, 1020 bytes in total.
fingerprint	004h	4	4E46_4342h ASCII: “NFCB”

Table continues on the next page...

Table 67. Flash control block structure (continued)

<table>
<tr><th>Name</th><th>Offset</th><th>Size bytes</th><th>Description</th></tr>
<tr><td>version</td><td>008h</td><td>4</td><td>0000_0001h</td></tr>
<tr><td>DBBTSearchStartPage</td><td>00Ch</td><td>4</td><td>Start page address for bad block table search area</td></tr>
<tr><td>searchStride</td><td>010h</td><td>2</td><td>Search stride for DBBT and FCB search.</td></tr>
<tr><td>searchCount</td><td>012h</td><td>2</td><td>Copies of DBBT and FCB.</td></tr>
<tr><td>firmwareCopies</td><td>014h</td><td>4</td><td>Firmware copies
Valid range 1-8.</td></tr>
<tr><td>Reserved</td><td>018h</td><td>40</td><td>Reserved
Must be set to 0.</td></tr>
<tr><td>firmwareInfoTable</td><td>040h</td><td>64</td><td>This table consists of (up to 8 entries):
<table><tr><th>Field</th><th>Size</th><th>Description</th></tr><tr><td>StartPage</td><td>4</td><td>Start page of this firmware</td></tr><tr><td>pageCount</td><td>4</td><td>Pages in this firmware</td></tr></table></td></tr>
<tr><td>Reserved</td><td>080h</td><td>128</td><td>Reserved
Must be set to 0.</td></tr>
<tr><td>xspi_nand_config_t</td><td>100h</td><td>616</td><td>Serial NAND configuration block over XSPI. It consists of common XSPI memory configuration block and Serial NAND specified configuration parameters. see Serial NAND configuration block (616 bytes) for more details.</td></tr>
<tr><td>xmc_config</td><td>368h</td><td>24</td><td>XMCD configuration for pSRAM. See External Memory Configuration Data (XMCD).</td></tr>
<tr><td>Reserved</td><td>380h</td><td>128</td><td>Must be set to 0.</td></tr>
</table>

Table 68. CRC-32 variant algorithm

Property	Description
Width	32 bits
Polynomial	04C11DB7h
Initial value	FFFFFFFFh
Reflect in	False
Reflect out	False
XOR out	00000000h

9.7.3.1 Serial NAND configuration block (616 bytes)

Table 69. Serial NAND configuration block

Name	Offset	Size (bytes)	Description
memCfg	0h	552	The common memory configuration block, see Table 76 for more details
pageDataSize	228h	4	Page size in terms of bytes, usually, it is 2048 or 4096
pageTotalSize	22ch	4	It equals to $2^{\text{width of column adddress +1}}$
pagesPerBlock	230h	4	Pages in one block
bypassReadStatus	234h	1	• 0b – Read status Register • 1b – Bypass read status register
bypassEccRead	235h	1	• 0b – Perform ECC read • 1b – Bypass ECC read
hasMultiPlanes	236h	1	• 0b – Only 1 plane • 1b – Has two planes
skippOddBlocks	237h	1	• 0b – Read odd blocks • 1b – Skip odd blocks
eccCheckCustomEnable	238h	1	• 0b - Use common ECC check command and ECC related masks • 1b - Use ECC check related masks provided in this configuration block
ipCmdSerialClkFreq	239h	1	• 00b – No change, keep current serial clock unchanged • 0001b – 30 MHz • 0010b – 50 MHz • 0011b – 60 MHz • 0100b – 80 MHz • 0101b – 100 MHz • 0110b – 120 MHz • 0111b – 133 MHz • 1000b – 166 MHz
readPageTimeUs	23Ah	2	Wait time during page read, this field takes effect only if the bypassReadStatus is set to 1. **NOTE** Only applicable to ROM.
eccStatusMask	23Ch	4	ECC status mask

Table continues on the next page...

Table 69. Serial NAND configuration block (continued)

Name	Offset	Size (bytes)	Description
eccFailureMask	240h	4	ECC check failure mask
blocksPerDevice	244h	4	Blocks in a serial NAND
Reserved	248h	32	—

9.7.3.2 External Memory Configuration Data (XMCD)

On-chip RAM may not be enough for comprehensive or feature-rich applications. An application requires a large capacity of RAM and needs to run on the external RAM for high execution performance. However, enabling the external RAM on the chip is complicated, and the settings are chip specific. The Boot ROM on the chip simplifies the external RAM configuration by introducing the XMCD. The XMCD data structure resides at offset 368h starting from NAND FCB.

The XMCD consists of an XMCD header and a memory specific configuration block. On this chip, the following types of external RAM devices are supported. Each type of RAM device has a specific configuration block.

9.7.3.2.1 Simplified XSPI RAM configuration option structure

The simplified XSPI RAM configuration option structure is shown in the following tables.

Table 70. XMCD header definition

[31:28] Tag	[27:24] Version	[23:20] Memory interface	[19:16] Instance	[15:12] Configuration block type	[11:0] Configuration block size
Valid value = Ch	Fixed value of 0	• 0000b - XSPI • 0001b - SEMC (not used for XSPI NAND) • Others - Reserved	Instance number	• 0000b - Simplified configuration option block • 0001b - Full configuration block (not used for XSPI NAND) • Others - Reserved	Configuration block size (including the XMC header itself).

Table 71. Simplified XSPI RAM configuration option 0

[31:28] Tag	[27:24] Option size	[23:20] device type	[19:16] Reserved	[15:12] Misc.	[11:8] Maximum frequency	[7:0] Size in MB
Fixed to Ch	option words： • 0b – only has option0	• 0b - HyperRAM • 1b - APMemory	Reserved	For HyperRAM • 0b - 1.8 V • 1b - 3 V • Others - Reserved	Chip specific definitions • 0001b - 30 MHz	• 0b - Auto detection • Others - Size in MB

Table continues on the next page...

Table 71. Simplified XSPI RAM configuration option 0

[31:28] Tag	[27:24] Option size	[23:20] device type	[19:16] Reserved	[15:12] Misc.	[11:8] Maximum frequency	[7:0] Size in MB
	• 1b – Have option0 and option1 • Others - Reserved	• Others - Reserved			• 0010b - 50 MHz • 0011b - 60 MHz • 0100b - 80 MHz • 0101b - 100 MHz • 0110b - 120 MHz • 0111b - 133 MHz • 1000b - 166 MHz • 1001b - 200 MHz • Others - Reserved	

Table 72. Simplified XSPI RAM configuration option 1

[31:28] RAM_CONNECTION	[27:24] Reserved	[23:20] DQS_PINMUX_GROUP	[19:16] PINMUX ROUP	[15] x16_enable	[14:8] max_cs_low_time	[7:4] write dummy cycles	[3:0] read dummy cycles
• 0b - PORTA • 1b - PORTB • Others - Reserved	Reserved	• 0b - Default Group • Others - Secondary group	• 0b - Primary group • Others - Secondary group	• 0b – x16 mode not enabled • 1b - x16 mode enabled • Others - Reserved	Maximum CS low time in 0.1 µs	• 0b - Auto detection • Others - Specified dummy cycles	• 0b - Auto detection • Others - Specified dummy cycles

9.7.3.2.2 Example of XMCD for HyperRAM Support

The following table shows an example of XMCD for HyperRAM and APMemory support.

Table 73. Example XMCD block for HyperRAM

Offset	Field	Description
0h	C001_0008h	Tag = Ch Version = 0 Memory Interface=0: XSPI Instance= 1: XSPI1 Configuration block type=0: Simplified configuration option block Configuration block size= 8: (4-byte header + 4-byte option block)
1h	C000_0700h	Tag = Ch Option_Size = 0: only has option0 DeviceType=0: HyperRAM Reserved: 0 Misc=0: HyperRAM 1.8 V Max Freq= 7: 133 MHz Memory Size=00: Auto-detection

Table 74. Example XMCD block for APMemory

Offset	Field	Description
0	C001_0008h	Tag = Ch Version =0 Memory Interface=0: XSPI Instance= 1: XSPI1 Configuration block type=0: Simplified configuration option block Configuration block size= 8: 4-byte header + 4-byte option block
1	C110_0900h	Tag = Ch Option_Size = 1: Have option0 and option1 DeviceType=1: APMemory Reserved:00 Max Freq=9: 200 MHz Memory Size=00: Auto-detection
2	00000500h	RAM_CONNECTION=0: PORTA DQS_PINMUX_ GROUP=0: Default group PINMUX ROUP=0: Primary group x16_enable=0: x16 mode not enabled max_cs_low_time=5: 0.5 µs

Table continues on the next page...

Table 74. Example XMCD block for APMemory (continued)

Offset	Field	Description
		write dummy cycles=0: Auto detection read dummy cycles=0: Auto detection

9.7.4 Discovered Bad Blocks Table (DBBT)

Table 75. DBBT structure

Name	Offset	Size (bytes)	Description
crcChecksum	000h	4	Checksum. Calculated from Fingerprint to the end of DBBT, 1052 bytes in total
Fingerprint	004h	4	32-bit word with a value of 4442_4254h in ASCII “DBBT”
Version	008h	4	32-bit version number, this version of DBBT is 0000_0001h
Reserved	00Ch	4	Reserved
badBlockNumber	010h	4	Number of bad blocks. The maximum value is 0000_0100h.
Reserved	014h	12	Must be filled with all 0s
Bad block entries	020h	1024 (256 x 4)	Each bad block entry is a 32-bit value specifying the number of a found bad block. The number of valid bad block entries is specified by the badBlockNumber field, where valid bad block entries are stored sequentially starting at the beginning of the bad block entries segment. Unused bad block entries (those beyond the badBlockNumber) should be filled with all Fs.

9.7.4.1 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block. If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block will be checked.

9.7.5 Serial NOR and NAND flash configuration based on XSPI interface

The ROM software supports serial NOR flash and serial NAND flash based on XSPI module, using a 448-bytes common XSPI configuration block and several specified parameters for serial NOR flash and serial NAND flash respectively. See the following sections for more details.

9.7.5.1 XSPI configuration block

XSPI configuration block consists of parameters regarding specific flash devices including read command sequence, quad mode enablement sequence (optional), etc.

Table 76. XSPI configuration block

Name	Offset	Size(bytes)	Description
Tag	000h	4	4246_4346h, ascii: 'FCFB'
Version	004h	4	[07:00] bugfix = 0 [15:08] minor [23:16] major = 1

Table continues on the next page...

Table 76. XSPI configuration block (continued)

Name	Offset	Size(bytes)	Description
			[31:24] ASCII 'V'
Reserved	008h	4	—
readSampleClkSrc	00Ch	1	• 00h – internal loopback • 01h – loopback from DQS pad • 03h – Flash provided DQS
csHoldTime	00Dh	1	Serial flash CS Hold Time Recommend default value is 03h
csSetupTime	00Eh	1	Serial flash CS setup time Recommended default value is 03h
columnAdressWidth	00Fh	1	03h – For HyperFlash 0Bh/0Ch – For Serial NAND, see data sheet of your NAND flash for the value 00h – Other devices
deviceModeCfgEnable	010h	1	Device mode configuration enable feature • 00h – Disabled • 01h –Enabled
deviceModeType	011h	1	• 00h - Generic • 01h - Quad Enable • 02h - SPI-to-XSPI mode • 03h - XSPI-to-SPI mode • 04h - SPI-to-NoCmd mode
waitTimeCfgCommands	012h	2	Wait time for all configuration commands, unit 100 µs. Available for device that support XSPI configuration block. If it is greater than 0, ROM will wait waitTimeCfgCommands x 100 µs for all device memory configuration commands instead of using read status to wait until these commands complete.
deviceModeSeq	014h	4	Sequence parameter for device mode configuration Bit[7:0] - number of LUT sequences for Device mode configuration command
			Bit[15:8] - starting LUT index of Device mode configuration command Bit[31:16] - must be 0

Table continues on the next page...

Table 76. XSPI configuration block (continued)

Name	Offset	Size(bytes)	Description
deviceModeArg	018h	4	Device mode argument, effective only when deviceModeCfgEnable = 01h
configCmdEnable	01Ch	1	Configure command enable feature • 00h – Disabled • 01h – Enabled
configModeType	01Dh	3	Configure mode type • 00h - Generic • 01h - Quad Enable • 02h - SPI-to-XSPI mode • 03h - XSPI-to-SPI mode • 04h - SPI-to-NoCmd mode
configCmdSeqs	020h	12	Sequences for configure command, allow 3 separate configuration command sequences.
ipedMode	02Ch	1	IPED mode • 00h - kxSpiIpedModeNone • 01h - kxSpiIpedModeCtr • 02h - kxSpiIpedModeGcm • 03h - kxSpiIpedModeXex • Others - Reserved
x16DllaSlvFineOffset	02Dh	1	Fine offset delay elements in incoming DQS1
maxCsLowInterval	02Eh	1	Configure pSRAM tcsm, unit 0.1 µs
ahbAlignment	02Fh	1	AHB alignment, • 00b - No limit • 01b - 256 B • 10b - 512 B • 11b - 1 KB • Others - Reserved
cfgCmdArgs	030h	12	Arguments for each separate configuration command sequence.
ahbSplitEn	03Ch	1	Enable AHB transaction split for pSRAM
Reserved	03Dh	3	—
controllerMiscOption	040h	4	Bit0 – differential clock enable Bit1 – CK2 enable, must set to 0 Bit2 – ParallelModeEnable

Table continues on the next page...

Table 76. XSPI configuration block (continued)

Name	Offset	Size(bytes)	Description
			Bit3 – wordAddressableEnable Bit4 – Safe Configuration Frequency enable set to 1 for the devices that support DDR Read instructions Bit5 – Pad Setting Override Enable Bit6 – DDR Mode Enable, set to 1 for device supports DDR read command Bit 7 - Pad Setting Override Enable Bit 8 - Second Pin group Bit 9 - Second DQS pin group Bit 10 - Write Mask Enable Bit 11 - Write Opt1 Clear Other bits - Reserved
deviceType	044h	1	• 01h – Serial NOR • 02h – SerialNAND • 03h - Serial RAM (HyperRAM/APMemory)
sflashPadType	045h	1	• 01h – Single pad • 02h – Dual pads • 04h – Quadpads • 08h – Octalpads
serialClkFreq	046h	1	• 01h – 30 MHz • 02h – 50 MHz • 03h – 60 MHz • 04h – 80 MHz • 05h – 100 MHz • 06h – 120 MHz • 07h – 133 MHz • 08h – 166 MHz • Others: 30 MHz
lutCustomSeqEnable	047h	1	• 00h– Use pre-defined LUT sequence index and number • 01h- Use LUT sequence parameters provided in this block
Reserved	048h	8	—
sflashA1Size	050h	4	For XSPI NOR, need to fill with actual size

Table continues on the next page...

Table 76. XSPI configuration block (continued)

Name	Offset	Size(bytes)	Description
			For XSPI NAND, need to fill with actual size x 2
sflashA2Size	054h	4	For XSPI NOR, need to fill with actual size For XSPI NAND, need to fill with actual size x 2
sflashB1Size	058h	4	For XSPI NOR, need to fill with actual size For XSPI NAND, need to fill with actual size x 2
sflashB2Size	05Ch	4	For XSPI NOR, need to fill with actual size For XSPI NAND, need to fill with actual size x 2
csPadSettingOverride	060h	4	Overriding pad setting of CS with this value if it is not 0h
sclkPadSettingOverride	064h	4	Overriding pad setting of SCLK with this value if it is not 0h
dataPadSettingOverride	068h	4	Overriding pad setting of data signals with this value if it is not 0h
dqsPadSettingOverride	06Ch	4	Overriding pad setting of DQS with this value if it is not 0h
timeoutInMs	070h	4	Maximum wait time with the unit of ms during read busy status • 0000_0000h – Disabled timeout checking feature • Other value – Timeout if the wait time exceeds this value
commandInterval	074h	4	Unit: ns Currently, it is used for XSPI NAND only at high frequency
dataValidTime	078h	4	Time from clock edge to data valid edge, unit ns. This field is used when the XSPI1 Root clock is less than 100 MHz and the read sample clock source is device provided DQS signal without CK2 support. • [31:16] data valid time for DLLB in terms of 0.1 ns • [15:0] data valid time for DLLA in terms of 0.1 ns
busyOffset	07Ch	2	busy bit offset, valid range :0000h-001Fh
busyBitPolarity	07Eh	2	• 0000h – busy bit is 1 if device is busy • 0001h – busy bit is 0 if device is busy
lookupTable	080h	360	Lookup table
lutCustomSeq	1E8h	48	Customized LUT sequence, see Table 77 for details.

Table continues on the next page...

Table 76. XSPI configuration block (continued)

Name	Offset	Size(bytes)	Description
dllCraSdrVal	218h	4	Customizable DLLCRA for SDR setting
smprSdrVal	21Ch	4	Customizable SMPR SDR setting
dllCraDdrVal	220h	4	Customizable DLLCRA for DDR setting
smprDdrVal	224h	4	Customizable SMPR DDR setting

9.7.5.2 Serial (XSPI) NAND, the pre-defined LUT

Table 77. LUT sequence definition for serial NAND

Command index	Name	Index in lookup table	Description
0	ReadFromCache	0	Read from cache
1	ReadStatus	1	Read status
2	WriteEnable	3	Write Enable
3	BlockErase	5	Erase block
4	ProgramLoad	9	Program load
5	ReadPage	11	Read page to cache
6	ReadEccStatus	13	Read ECC status
7	ProgramExecute	14	Program execute
8	ReadFromCacheOdd	4	Read from cache while page in odd plane
9	ProgramLoadOdd	10	Program load for pages within odd blocks
Others	Reserved	2,6,7,8,12,15	All reserved indexes can be freely used for other purposes

NOTE

All the pre-defined LUT indexes are only applicable to boot stage. User application can use the whole 16 LUT entries freely based on their requirement.

9.7.6 Program XSPI NAND using ISP

Boot ROM has implemented In-System-Programming (ISP) to enable programming XSPI NAND flash via available peripherals and thus can be utilized to configure and program XSPI NAND.

For configuration of XSPI NAND, fill-memory with configuration words and configure-memory command can be used. Boot ROM will always store the configuration words to RAM with fill-memory command, and then apply configuration to configure the memory device via configure-memory. For XSPI NAND configuration words in ISP mode, detailed definition is described as below.

```
kSpiNandMemory_MaxFirmware = 8U,
typedef struct _spinand_img_option
{
    union
    {
        struct
```

```
        {
            uint32_t size : 4;          //!< Size in longword
            uint32_t reserved0 : 4;     //!< Reserved for future use
            uint32_t address_type : 4;  //!< NAND Adress type: 0 - byte address, 1 - block address
            uint32_t reserved1 : 8;     //!< Reserved for future use
            uint32_t search_stride : 4; //!< Search Stride: 0-64 pages, 1-128 pages, 2-256 pages,
3-32 pages.
            uint32_t search_count : 4;  //!< 1 - 1, 2 -2 ,3 - 3, 4 - 4
            uint32_t tag : 4;           //!< tag, fixed to 0x0C
        } B;
        uint32_t U;
        struct
        {
            uint32_t cf9_field : 20; //!< CFG data field, Choose the XSPI instance
            uint32_t magic : 12;     //!< Magic, must be 0xcf9 (standing for "CFG")
        } P;                         // Preconfig
    } option0;
    uint32_t nand_option_addr;
    struct
    {
        uint16_t block_count; //!!< Maximum block count for an application image
        uint16_t block_id;    //!< Image start block ID
    } image_info[kSpiNandMemory_MaxFirmware];
} spinand_img_option_t;
```

Table 78. XSPI NAND configuration block definition

Option index	>Bits	Name	Description
Option0	Select XSPI instance if [31:20] is CF9h		
	[31:20]	magic	Magic number, must be CF9h for CFG
	[19:0]	cf9_field	XSPI instance
	Option0 for configuring XSPI NAND if [31:20] is not CF9h		
	[31:28]	tag	fixed to 0Ch
	[27:24]	search_count	Search count for FCB and DBBT • 1h - 1 • 2h - 2 • 3h - 3, • 4h - 4
	[23:20]	search_stride	Search stride for FCB and DBBT • 0h - 64 pages • 1h - 128 pages • 2h - 256 pages • 3h - 32 pages
	[19:12]	Reserved	—

Table continues on the next page...

Table 78. XSPI NAND configuration block definition (continued)

Option index	>Bits	Name	Description
	[11:8]	address_type	Use the NAND block offset address or the physical offset address, • 0h - byte address • 1h - block address
	[7:4]	Reserved	—
	[3:0]	size	All option bytes size used for the configuration
Option1	[31:0]	nand_option_addr	See Table 79 for configuration options.
Option2	[31:16]	block_id	XSPI NAND image0 load start block ID
	[15:0]	block_count	Maximum block count for XSPI NAND image0
Option*[n]* (*n*=3-9)	[31:16]	block_id	XSPI NAND image*[n-2]* load start block ID
	[15:0]	block_count	Maximum block count for XSPI NAND image*[n-2]*

```
typedef struct _spinand_option
{
    serial_nand_config_option_t nand_config;
    spinand_xmc_info_t xmc_config;
} spinand_option_t;

typedef struct
{
    union
    {
        struct
        {
            uint32_t max_freq : 4;        //!< Device specific definitions, see System Boot Chapter
for more details
            uint32_t page_size_in_kb : 4; //!< 2 - 2048bytes, 4 - 4096 Bytes
            uint32_t pages_per_block : 4; //!< 0 - 64, 1 - 128, 2 - 256, 3 - 32
            uint32_t has_multiplanes : 4; //!< 0 - no , 1 - yes
            uint32_t flash_size : 4;      //!< 0 - 512Mbit, 1 - 1Gbit, 2 - 2Gbit, 4 - 4Gbit, 8 - 8Gbit
            uint32_t device_type : 4;     //!< Device type, 0 - Quad, 1 - Octal
            uint32_t option_size : 4;     //!< Option size : (option_size + 1) * 4 bytes
            uint32_t tag : 4;             //!< Fixed to 0x0C
        } B;
        uint32_t U;
    } option0;

    union
    {
        struct
        {
            uint32_t manufacturer_id : 8; //!< Manufacturer ID
            uint32_t eccFailureMask : 8;  //!< Ecc Failure Mask
            uint32_t eccCheckMask : 8;    //!< Ecc Check Mask
            uint32_t reserved : 8;        //!< Reserved for future use.
        } B;
        uint32_t U;
```

```
    } option1;
} serial_nand_config_option_t;
```

Table 79. nand_option_addr configuration options

Option index	Bits	Name	Description
Option0	[31:28]	tag	fixed to Ch
	[27:24]	option_size	Options size : (option_size + 1) x 4 bytes
	[23:20]	device_type	Device type • 0h - Quad • 1h - Octal
	[19:16]	flash_size	• 0h - 512 Mbit • 1h - 1 Gbit • 2h - 2 Gbit • 4h - 4 Gbit • 8h - 8 Gbit
	[15:12]	has_multiplanes	Configure the NAND model, which has the multiplanes • 0h - no • 1h - yes
	[11:8]	pages_per_block	Page numbers per block in the NAND flash • 0h - 64 • 1h - 128 • 2h - 256 • 3h - 32
	[7:4]	page_size_in_kb	Byte size per page in the NAND flash • 2h - 2048 bytes • 4h - 4096 bytes
	[3:0]	max_freq	Maximum frequency • 0h - 80 MHz • 1h - 100 MHz • 2h - 120 MHz • 3h - 133 MHz • 4h - 166 MHz • 5h - 200 MHz • 6h - 50 MHz • 7h - 30 MHz

Table continues on the next page...

Table 79. nand_option_addr configuration options (continued)

Option index	Bits	Name	Description
Option1	[31:24]	Reserved	—
	[23:16]	eccCheckMask	ECC check mask
	[15:8]	eccFailureMask	ECC failure mask
	[7:0]	manufacturer_id	Manufacturer ID

NOTE

See External Memory Configuration Data (XMCD) for spinand_xmc_info_t structure.

Below is an example of configuring XSPI NAND(W25N02JW) boot image with block address access:

```
//Initialize XSPI NAND
// Choose the XSPI instance 0 for NAND interface
blhost -p com4 -- fill-memory 0x20020000 0x4 0xcf900000
blhost -p com4 -- configure-memory 0x102 0x20020000
// Configure the FCB generated arguments, search_count:1, search_stride: 64 pages, address_type:
block address, size: 3 words in the options
blhost -p com4 -- fill-memory 0x20020000 0x4 0xc1000103
// The spinand_option_t options address
blhost -p com4 -- fill-memory 0x20020004 0x4 0x2002000c
// nand image block_id set as 2 means the image loaded in the block2 region, the block_count set as 1
means the image payload max block number is 1 block.
blhost -p com4 -- fill-memory 0x20020008 0x4 0x00020001
// The spinand_option_t options, Quad, 2Gbit, no multiplanes, 64 pages per block, 2048B per page,
166Mhz running clock
blhost -p com4 -- fill-memory 0x2002000c 0x4 0xc0020024
blhost -p com4 -- configure-memory 0x102 0x2002000c
//Erase NAND memory, start block id is 2, and erase block size is 1 block
blhost -p com4 -- flash-erase-region 0x2 0x1 0x102
//Write image to NAND
blhost -p com4 -- write-memory 0x00000002 boot_image.bin 0x102
```

Below is an example of configuring XSPI NAND(W25N02JW) boot image with physical address access:

```
//Initialize XSPI NAND
// Choose the XSPI instance 0 for NAND interface
blhost -p com4 -- fill-memory 0x20020000 0x4 0xcf900000
blhost -p com4 -- configure-memory 0x102 0x20020000
// Configure the FCB generated arguments, search_count:1, search_stride: 64 pages, address_type:
physical address, size: 3 words in the options
blhost -p com4 -- fill-memory 0x20020000 0x4 0xc1000003
// The spinand_option_t options address
blhost -p com4 -- fill-memory 0x20020004 0x4 0x2002000c
// nand image block_id set as 2 means the image loaded in the block2 region, the block_count set as 1
means the image payload max block number is 1 block.
blhost -p com4 -- fill-memory 0x20020008 0x4 0x00020001
// The spinand_option_t options, Quad, 2Gbit, no multiplanes, 64 pages per block, 2048B per page,
166Mhz running clock
blhost -p com4 -- fill-memory 0x2002000c 0x4 0xc0020024
blhost -p com4 -- configure-memory 0x102 0x2002000c
//Erase NAND memory, start block address is 0x40000, which is the block2 start offset address, and
erase block size is 0x2000
blhost -p com4 -- flash-erase-region 0x40000 0x2000 0x102
```

```
//Write image to NAND
blhost -p com4 -- write-memory 0x40000 boot_image.bin 0x102
```

Below is an example of configuring XSPI NAND(W25N02JW) boot image with block address access and configure the PSRAM by XMCD :

```
//Initialize XSPI NAND
// Choose the XSPI instance 0 for NAND interface
blhost -p com4 -- fill-memory 0x20020000 0x4 0xcf900000
blhost -p com4 -- configure-memory 0x102 0x20020000
// Configure the FCB generated arguments, search_count:1, search_stride: 64 pages, address_type:
block address, size: 3 words in the options
blhost -p com4 -- fill-memory 0x20020000 0x4 0xc1000103
// The spinand_option_t options address
blhost -p com4 -- fill-memory 0x20020004 0x4 0x2002000c
// nand image block_id set as 2 means the image loaded in the block2 region, the block_count set as 1
means the image payload max block number is 1 block.
blhost -p com4 -- fill-memory 0x20020008 0x4 0x00020001
// The spinand_option_t options, Quad, 2Gbit, no multiplanes, 64 pages per block, 2048B per page,
166Mhz running clock
blhost -p com4 -- fill-memory 0x2002000c 0x4 0xc2020024
blhost -p com4 -- fill-memory 0x20020010 0x4 0xc0010008
blhost -p com4 -- fill-memory 0x20020014 0x4 0xc0000700
blhost -p com4 -- configure-memory 0x102 0x2002000c
//Erase NAND memory, start block id is 2, and erase block size is 1 block
blhost -p com4 -- flash-erase-region 0x2 0x1 0x102
//Write image to NAND
blhost -p com4 -- write-memory 0x00000002 boot_image.bin 0x102
```

9.8 Recovery boot

If prime boot fails, Boot ROM can start recovery boot routine if recovery boot is enabled. Recovery boot can be enabled by configuring fuse (BOOT_CFG1[REC_BOOT_EN] = 1). And recovery boot is booting from serial nor flash connected to selected LP_FLEXCOMM. Boot ROM always read the first 64 bytes image header from offset 1000h of serial memory device for recovery boot, copy the image to the destination and then validate the image. If everything is fine, a successful recovery boot is done and jump to the application, else ROM will enter ISP mode. See Figure 13 for the high level boot flow for recovery boot.

9.8.1 Recovery boot configuration

The following parameters need to be configured for recovery boot:

- LP_FLEXCOMM instance, which can be selected by fuse feild BOOT_CFG5[REC_BOOT_FC].
- Chip-Select (CS) pin, which can be chosen by fuse feild BOOT_CFG5[REC_BOOT_FC_PCS].
- Baud rate of LPSPI, which can be configured by fuse field BOOT_CFG5[REC_BOOT_FC_BAUDRATE]

See Boot pin allocation for detailed port and pin allocations.

To make recovery boot work, recovery boot configuration fuses and recovery NOR memory should be correctly configured before using. And image should be programed to device at offset1000h.

Fuse	Definition	Shipped value	Settings
REC_BOOT_EN (BOOT_CFG1[4])	Recovery boot enable	0	Enable recovery boot through LP_FLEXCOMM. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Fuse	Definition	Shipped value	Settings
			1 - Enable
REC_BOOT_FC (BOOT_CFG5[19:16])	Port Selection. Specifies the LP_FLEXCOMM instance connected to 1-bit SPI flash used for recovery boot.	0000	This field is valid only if REC_BOOT_EN is set. Valid value: 0-13 *x*d - LP_FLEXCOMM instance *x* is selected as interface.
REC_BOOT_FC_PCS (BOOT_CFG5[21:20])	Specifies LP_FLEXCOMM PCS connected to 1-bit SPI flash used for recovery boot. This field is valid only if REC_BOOT_EN is set.	00	00 - PCS0 01 - PCS1 10 - PCS2 11 - PCS3
REC_BOOT_FC_BAUDRATE (BOOT_CFG5[23:22])	Specifies LP_FLEXCOMM baudrate with 1-bit SPI flash used for recovery boot. This field is valid only if REC_BOOT_EN is set.	00	00 - 20 MHz 01 - 10 MHz 10 - 5 MHz 11 - 2 MHz

9.8.2 Program LPSPI NOR using ISP

In general, the following actions need to be taken to program memory device:

- Configure the memory
- Erase the memory
- Program the memory

Below table shows the configurable parameters when programing boot image for recovery boot at ISP mode

Table 80. LPSPI configuration

Offset	Field	Size	Description		
0	option0	4	EEPROM/NOR configuration options		
			Offset	**Field**	**Description**
			31:28	tag	Fixed Ch
			27:24	Reserved	—
			23:20	spi_index	LPSPI instance 0 - LPSPI0 x - LPSPIx 13 - LPSPI13
			19:16	pcs_index	CS index of the LPSPI module 0 - PCS0

Table continues on the next page...

Table 80. LPSPI configuration (continued)

Offset	Field	Size	Description		
					1 - PCS1 2 - PCS2 3 - PCS3
			15:12	memory_type	Memory device type 0 - NOR, parameters specified 2 - NOR, parameters auto detected
			11:8	memory_size	Memory size
					(0≤n≤11): 512 KB x 2^n
					(12≤n≤15): 32 KB x 2^{n-12}
			7:4	sector_size	Sector size n(0≤n≤1): 4 KB x 2^n n(2≤n≤5): 32 KB x 2^{n-2}
			3:0	page_size	Page size n(0≤n≤2): 256 B x 2^n n(3≤n≤5): 32 B x 2^{n-3}
4	Option1	4	**Offset**	**Field**	**Description**
			31:4	Reserved	Reserved
			3:0	speed	LPSPI clock speed 0 - 20 MHz (default) 1 - 10 MHz 2 - 5 MHz 3 - 2 MHz

Example for programing the serial memory for recovery boot.

```
    //Take below actions under ISP mode
    //Step1. Configure the device according to the device used
    blhost -p com4 -- fill-memory 0x30080000 0x4 0xc0312000; //Configuration word0: LPSPI3 used,
PCS1 as CS port, auto probe for nor device
    blhost -p com4 -- fill-memory 0x30080004 0x4 0x00000000; //Configuration word1: LPSPI speed is
20 mbps
    blhost -p com4 -- configure-memory 0x110 0x30080000; //Configure serial memory device.
    //Step2. Erase the region
    blhost -p com4 -- flash-erase-region 0x0 0x40000 0x110
    //Program boot image to destination
    blhost -p com4 -- write-memory 0x1000 C:\recovery_boot.bin 0x110
```

Chapter 10
ROM API

10.1 Overview

This section provides an introduction to the ROM API in this device.

NOTE

This chapter is also available in the i.MX RT700 Security Reference Manual, with differences related to security functionality. The chapter in the Reference Manual does not show the security relevant information in it.

10.1.1 Features

- The ROM API supports OTP-eFuse read and programming.
- The ROM API provides native support for Nboot API in the user application.
- The ROM API supports In-application-programming functionality by the IAP API, via the unified memory interface or the Sbloader API.

10.2 Functional description

This section introduces the ROM API structure first, then describes the OTP API, IAP API and NBOOT APIs. This section also demonstrates the typical usage of each API.

- OTP API provides the functionality to advance lifecycle; program/read critical one-time programmable parameters and some general-purpose one-time programmable FUSE field.
- IAP API enables the flexibility of doing in-application-programming by SBloader API.

NOTE

All ROM APIs are callable from secure world only. Therefore, all function pointers use secure address aliases. Applications should switch to secure mode and then call ROM API or create NSC veneers for ROM APIs.

10.2.1 ROM API structure

The ROM API table locates at address 0x1303fc00. See Figure 20.

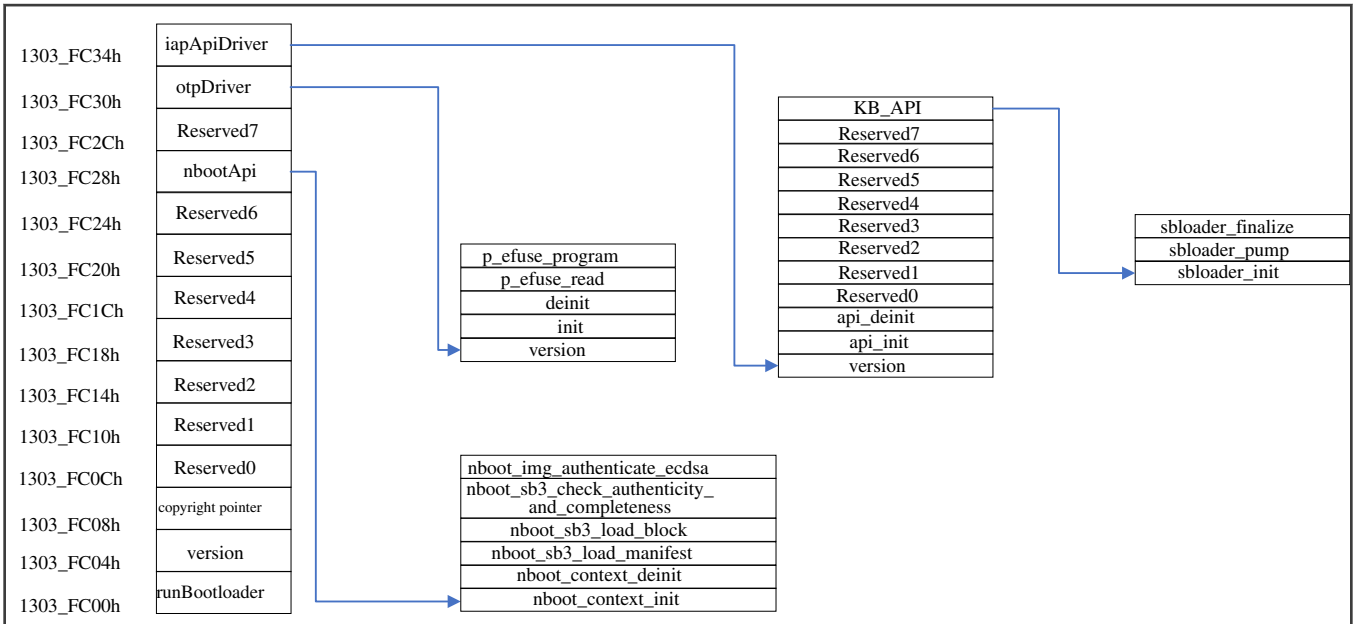


Figure 20. ROM API structure

10.2.2 runBootloader API

The ROM bootloader provides an API for the user application to enter boot mode or ISP mode based on the designated ISP interface mode.

Prototype

```
void (*runBootloader)(void *arg);
```

Table 81. runBootloader API arg fields

Field	Offset	Description	
Tag	[31:24]	Fixed value: EBh (enter boot)	
Boot mode	[23:20]	0000b - Prime boot mode	
		0001b - ISP mode	
		0010b - ProvFw mode. It requests ROM bootloader to handle provisioning firmware resides in SRAM P4 and P5. Provisioning firmware should be in SB3.1 format and it should be preloaded to start of SRAM P4 before calling runBootloader API to enter ProvFw mode. For this mode, bit 0-19 of this arg don't need to be set.	
Boot interface	[19:16]	**ISP boot mode**	**Prime boot mode**
		0h - UART	4h – XSPI NOR
		1h - I2C	5h – 1-bit LPSPI NOR
		2h - LPSPI	6h – Reserved
		3h - USB-HID (USB0)	7h – eMMC

Table continues on the next page...

Table 81. runBootloader API arg fields (continued)

Field	Offset	Description	
		8h - eUSB (USB1)	9h – XSPI NAND
		Fh - Auto detection	Other values – Reserved
Instance	[15:12]	Instance number when Boot Mode is configured as Prime boot mode and the Boot interface is configured as XSPI NOR (4h), LPSPI NOR (5h) or eMMC (7h): If Boot interface = 4h • 0000b - XSPI0 • 0001b - XSPI1 • Others - Reserved If Boot interface = 5h • 0000b - LPSPI0 • …. • 1101b - LPSPI13 • Others - Reserved If Boot interface = 7h • 0000b - eMMC0 • 0001b - eMMC1 • Others - Reserved	
Reserved	[11:8]	—	
recovery_boot_cfg1	[7:6]	Recovery boot interface configuration 1	
recovery_boot_cfg0	[5:4]	Recovery boot interface configuration 0	
Reserved	[3:0]	—	

Example:

In the application image boot process, use runbootloader api to reset rom boot from XSPI0.

```
#define ROM_API_TREE ((uint32_t*)0x1303fc00)
#define RUN_BOOTLOADER_API_TREE ((void (*)(void *)) ROM_API_TREE[0])
uint32_t arg = 0xEB040000U;
RUN_BOOTLOADER_API_TREE->runBootloader(&arg);
```

After the application image, which calls the above runBootloader API, has booted successfully, the device will boot image from XSPI0.

10.2.3 OTP-eFuse APIs

10.2.3.1 Version

This fields indicates the version number of the OTP API in this device.

Prototype :

```
standard_version_t version;
```

Table 82. Definition of the version Field in OTP API tree

Parameter	Description
standard_version_t	Pointer to version structure to store current eFuse driver version information uint8_t bugfix; /* bugfix version [7:0] */ uint8_t minor; /* minor version [15:8] */ uint8_t major; /* major version [23:16] */ char name; /* character name is "E", where "E" stands for eFuse. Character name and version are fixed */

Example:

```
#define ROM_API_TREE ((uint32_t*)0x1303fc00)
#define OTP_API_TREE ((efuse_driver_t *)ROM_API_TREE[12])
uint32_t otpDriverVersion = OTP_API_TREE-> version;
```

In this device, the OTP driver version is "0x45010100", it means the OTP driver version is 1.1.0.

Following is the efuse_driver_t definition:

```
//!@brief EFUSE driver API Interface
typedef struct
{
    standard_version_t version; //!< efuse driver API version number.
    void (*init)(uint32_t src_clk_freq);
    status_t (*deinit)(void);
    status_t (*p_efuse_read)(uint32_t addr,      uint32_t *data);
    status_t (*p_efuse_program)(uint32_t addr, uint32_t data);
} efuse_driver_t;
```

10.2.3.2 efuse_init

The API is used for initializing the OTP clock. The *efuse_init* API should be called before all the other OTP APIs. The parameter src_clk_freq is the frequency of OTP module source clock. On this chip, it is the frequency of COMPUTE_MAIN_CLK.

Prototype :

```
void (*init)(uint32_t src_clk_freq);
```

Example :

```
//efuse init
 debug_printf("Start efuse init! \r\n");
 OTP_API_TREE->init (192000000);
```

See the possible status codes in Table 85. If the status is kStatus_Success return value, it means OTP driver initialization completed successfully.

10.2.3.3 efuse_deinit

This API is used to disable eFuse controller clock.

Prototype :

```
status_t (*deinit)(void);
```

Example :

```
uint32_t status = OTP_API_TREE->deinit ();
```

See the possible status codes in Table 85. If the status is kStatus_Success return value, it means OTP driver deinitialization completed successfully.

10.2.3.4 efuse_read

This API is used to read data from OTP-eFuse controller.

Prototype :

```
status_t (*p_efuse_read)(uint32_t addr, uint32_t *data);
```

Table 83. Parameters used in otp_read API

Parameter	Description
addr	eFuse index.
data	Pointer to the buffer used for storing the read 32 bit data.

Example :

```
uint32_t efuseIndex = 472;
uint32_t fuseReadVal = 0;

//efuse read
uint32_t status = OTP_API_TREE-> p_efuse_read (efuseIndex, &fuseReadVal);
```

See the possible status codes in Table 85. If the status is kStatus_Success return value, it means the OTP data read completed successfully.

10.2.3.5 efuse_program

This API is used to program data to specified OTP Word.

Prototype :

```
status_t (*p_efuse_program)(uint32_t addr, uint32_t data);
```

Table 84. Parameters used in the otp_program API

Parameter	Description
addr	Pointer to the eFuse index that specifies the EFUSE word to be programmed
data	Pointer to the 32-bit source data used for programming into the eFuse region.

Example:

```
#define EFUSE_INDEX (472)
uint32_t eFuseData = 0x01010101;
uint32_t status = OTP_API_TREE-> p_efuse_program (EFUSE_INDEX, & eFuseData);
```

See the possible status codes in Table 85. If the status is kStatus_Success return value, it means no error has been reported for this program operation. Normally, it means the data 0x01010101 has been programmed into the eFuse word 472. However, if it is already programmed before, then its value will not be changed to 0x01010101 without error reported from this API. For a programmed ECC OTP word, calling this API will return kStatus_Success but the word value will not be changed.

10.2.3.6 Possible return codes of OTP API

Table 85. Return codes for OTP APIs

Error Code	Value (h)	Description
kStatus_Success	5AC3C35A	Operation succeeded without error.
kStatus_Fail	C35AC35A	The operation failed with a generic error.
kStatus_OTP_ProgramFail	CE42	Fail to program the OTP field

10.2.3.7 OTP-eFuse APIs Example

This section provides an example of OTP API usage.

NOTE

The OTP-eFuse program is only-program-once, user need to make sure the program data is valid value.

Example:

```
#define ROM_API_TREE ((uint32_t*)0x1303fc00)
#define OTP_API ((efuse_driver_t *)ROM_API_TREE[12])

OTP_API->init(192000000);
        debug_printf("efuse init succeess! \r\n");

        uint32_t fuseProgramVal = 0x12345678u;
        uint32_t fuseReadVal = 0;
        uint32_t efuseIndex1 = 472;
      uint32_t efuseIndex2 = 473;
        status_t status = kStatus_Success;

        //efuse read
        status = OTP_API->p_efuse_read(efuseIndex1,&fuseReadVal);
        if(status != kStatus_Success)
```

```
        {
            debug_printf("efuse_read failed ! efuseIndex = %d \r\n",efuseIndex1);
            break;
        }
        debug_printf("efuse_read efuseIndex = %d, fuseReadVal = %x \r\n",efuseIndex1,fuseReadVal);

        //efuse program
        status = OTP_API->p_efuse_program(efuseIndex2,fuseProgramVal);
        if(status != kStatus_Success)
        {
            debug_printf("efuse_program failed ! efuseIndex = %d \r\n",efuseIndex2);
            break;
        }
        debug_printf("efuse_program success! efuseIndex = %d, fuseProgramVal = %x
\r\n",efuseIndex2,fuseProgramVal);
        //efuse read and verify
        status = OTP_API->p_efuse_read(efuseIndex2,&fuseReadVal);
        if(status != kStatus_Success)
        {
            debug_printf("efuse_read failed ! efuseIndex = %d \r\n",efuseIndex2);
            break;
        }
        if(fuseProgramVal != fuseReadVal)
        {
            debug_printf("efuse program verifying FAILED, read value = %x, efuseIndex = %d
\r\n",fuseReadVal,efuseIndex2);
            status = kStatus_Fail;
            break;
        }
        debug_printf("efuse verify success! efuseIndex = %d \r\n",efuseIndex);

        status = OTP_API->deinit();
        if(status != kStatus_Success)
        {
            debug_printf("efuse deinit failed! \r\n");
            break;
        }
```

10.2.4 IAP APIs

10.2.4.1 General description

IAP is in–application–programming.

The bootloader IAP APIs are organized in the iap_api_interface_t structure, see the details from the below data structure. See the details of each API in the subsequent sections.

```
typedef struct iap_api_interface_struct
{
    standard_version_t version; //!< IAP API version number.
    status_t (*api_init)(api_core_context_t *coreCtx, const kp_api_init_param_t *param);
    status_t (*api_deinit)(api_core_context_t *coreCtx);
    const uint32_t reserved0 ;
    const uint32_t reserved1 ;
    const uint32_t reserved2 ;
    const uint32_t reserved3 ;
    const uint32_t reserved4 ;
    const uint32_t reserved5 ;
```

```
    const uint32_t reserved6 ;
    const uint32_t reserved7 ;
    status_t (*sbloader_init)(api_core_context_t *ctx);
    status_t (*sbloader_pump)(api_core_context_t *ctx, uint8_t *data, uint32_t length);
    status_t (*sbloader_finalize)(api_core_context_t *ctx);
} iap_api_interface_t;
```

Each IAP API depends on the common context named api_core_context_t. The detail of the context is defined as bellow:

```
/* @brief The API context structure */


typedef struct api_core_context
{
    uint32_t reserved0[166];
    ldr_Context_v3_t *sbloaderCtx;
    nboot_context_t *nbootCtx;
    uint32_t reserved1[5];
} api_core_context_t;

struct _ldr_Context_v3
{
    pLdrFnc_v3_t Action;        //!< pointer to loader action function
    uint32_t block_size;        //!< size of each block in bytes
    uint32_t block_data_size;   //!< data size in bytes (NBOOT_SB3_CHUNK_SIZE_IN_BYTES)
    uint32_t block_data_total;  //!< data max size in bytes (block_size * data_size
    uint32_t block_buffer_size; //!< block0 and block size
    uint32_t block_buffer_position;
    uint8_t block_buffer[MAX(NBOOT_SB3_MANIFEST_MAX_SIZE_IN_BYTES,
       NBOOT_SB3_BLOCK_MAX_SIZE_IN_BYTES)]; //! will be used for both block0 and blockx
    uint32_t processedBlocks;

    uint8_t data_block_offset; //! data block offset in a block.
    bool in_data_block;        //!< in progress of handling a data block within a block
    uint8_t *data_block;
    uint32_t data_block_position;

    bool in_data_section; //!< in progress of handling a data section within a data block
    uint32_t data_section_handled;
    sb3_section_header_t data_section_header;

    bool in_data_range; //!< in progress of handling a data range within a data section
    uint32_t data_range_handled;
    uint32_t data_range_gap;
    sb3_data_range_header_t data_range_header;
    bool has_data_range_expansion;
    sb3_data_range_expansion_t data_range_expansion;

    uint32_t commandSet; //!< support command set during sb file handling

    uint32_t data_position;
    uint8_t data_buffer[SB3_DATA_BUFFER_SIZE_IN_BYTE]; //!< temporary data buffer

    // options from ROM API
    kb_options_t fromAPI;
};
```

The kp_api_init_param_t is defined as below:

```
typedef struct kb_api_parameter_struct
{
    uint32_t allocStart;
    uint32_t allocSize;
} kp_api_init_param_t;
```

The IAP APIs require a dedicated RAM region as the buffer for each operation. The user application needs to prepare an unused RAM region organized as the *kp_api_init_param_t* structure, and pass this structure to the iap_api_init API. The allocSize in the iap_api_init API call is used to store the different flash context which is used in the api_core_context_t. The minimum size can be set to at least the size of (api_core_context_t).

These fields are managed by api_core_context_t parameter, the user application doesn't need to touch it.

NOTE

Before IAP APIs are used, caller needs to make sure that ELS clocks are enabled. This is required for correct operation of kb API on devices with security subsystem.

10.2.4.2 version

The version field indicates the IAP API version number in the device.

Prototype :

```
standard_version_t version;
```

Table 86. Definition of the version in IAP API tree

Parameter	Description

standard_version_t	Pointer to version structure to store current FLASH driver version information
	uint8_t bugfix; //!< bugfix version [7:0]
	uint8_t minor; //!< minor version [15:8]
	uint8_t major; //!< major version [23:16]
	char name; //!< character name is 0x49 = "I".

Example:

```
#define ROM_API_TREE ((uint32_t*)0x1303fc00)
#define IAP_API_TREE ((iap_api_interface_t*)ROM_API_TREE[13])
uint32_t IAPVersion = IAP_API_TREE-> version;
```

In this SoC, the version of IAP Driver is "0x49010100", it means the API version is 1.1.0.

10.2.4.3 iap_api_init

The API is used for initializing the IAP API context for the other IAP APIs, this API must be called prior to other IAP APIs.

Prototype :

```
status_t iap_api_init(api_core_context_t *coreCtx, const kp_api_init_param_t *param);
```

Table 87. Parameters used in iap_api_init API

Parameter	Description
coreCtx	Pointer to IAP API core context structure.
param	Pointer to IAP API initialization data structure, used for storing the data during execute the IAP APIs.

Example:

```
api_core_context_t api_ctx = {0};
kp_api_init_param_t params = {0};
params.allocStart = 0x20350000u;
params.allocSize = 0x6000u;
uint32_t status = IAP_API_TREE->api_init(&api_ctx,&params);
```

See the possible status codes in Table 89. If the status is kStatus_Success return value, it means the IAP API operating environment has been init successfully.

10.2.4.4 iap_api_deinit

The API is used for deinitializing the IAP API context.

Prototype :

```
status_t iap_api_deinit(api_core_context_t *coreCtx);
```

Table 88. Parameters used in iap_api_deinit API

Parameter	Description
coreCtx	Pointer to IAP API core context structure.

Example:

```
uint32_t status = IAP_API_TREE->api_deinit(&api_ctx);
```

See the possible status codes in Table 89. If the status is kStatus_Success return value, it means the IAP API operating environment has been freed and cannot call the other IAP APIs.

10.2.4.5 Possible return codes for IAP driver APIs

Table 89. Return and Error codes for IAP APIs

Error Code	Value	Description
kStatus_IAP_Success	0	IAP API execution succeeded
kStatus_IAP_Fail	1	IAP API execution failed
kStatus_Success	0	Operation succeeded without error.

Table continues on the next page...

Table 89. Return and Error codes for IAP APIs (continued)

Error Code	Value	Description
kStatus_Fail	1	The operation failed with a generic error.
kStatus_Invalidargument	4	The requested command's argument is undefined.
kStatusRomLdrSectionOverrun	10100	means reached the end of the sb file processing
kStatusRomLdrSignature	10101	the signature or version are incorrect
kStatusRomLdrSectionLength	10102	the bootOffset/ new section count is out of range
kStatusRomLdrEOFReached	10104	the end of the image file is reached
kStatusRomLdrChecksum	10105	The checksum of a command tag block is invalid.
kStatusRomLdrCrc32Error	10106	The CRC-32 of the data for a load command is incorrect.
kStatusRomLdrUnknownCommand	10107	An unknown command was found in the SB file.
kStatusRomLdrIdNotFound	10108	There was no bootable section found in the SB file.
kStatusRomLdrDataUnderrun	10109	The SB state machine is waiting for more data.
kStatusRomLdrJumpReturned	10110	The function that was jumped to by the SB file has returned.
kStatusRomLdrCallFailed	10111	The call command in the SB file failed.
kStatusRomLdrKeyNotFound	10112	A matching key was not found in the SB file's key dictionary to unencrypt the section.
kStatusRomLdrSecureOnly	10113	The SB file sent is unencrypted, and security on the target is enabled.
kStatusRomLdrResetReturned	10114	The SB file reset operation has unexpectedly returned.
kStatusRomLdrRollbackBlocked	10115	An Image version rollback event detected
kStatusRomLdrInvalidSectionMacCount	10116	Invalid Section MAC count detected in the SB file
kStatusRomLdrUnexpectedCommand	10117	The command tag in the sb-file is unexpected
kStatusRomLdrBadSBKEK	10118	Bad SBKEK detected

10.2.4.6 KB APIs

KB API is used in the application to update the firmware of device to process Secure Binary (SB) file. It include a set of SBloader APIs. The flow diagram of the SBloader APIs call demo is as below.

NOTE

KB APIs can't be called from SRAM alias address.

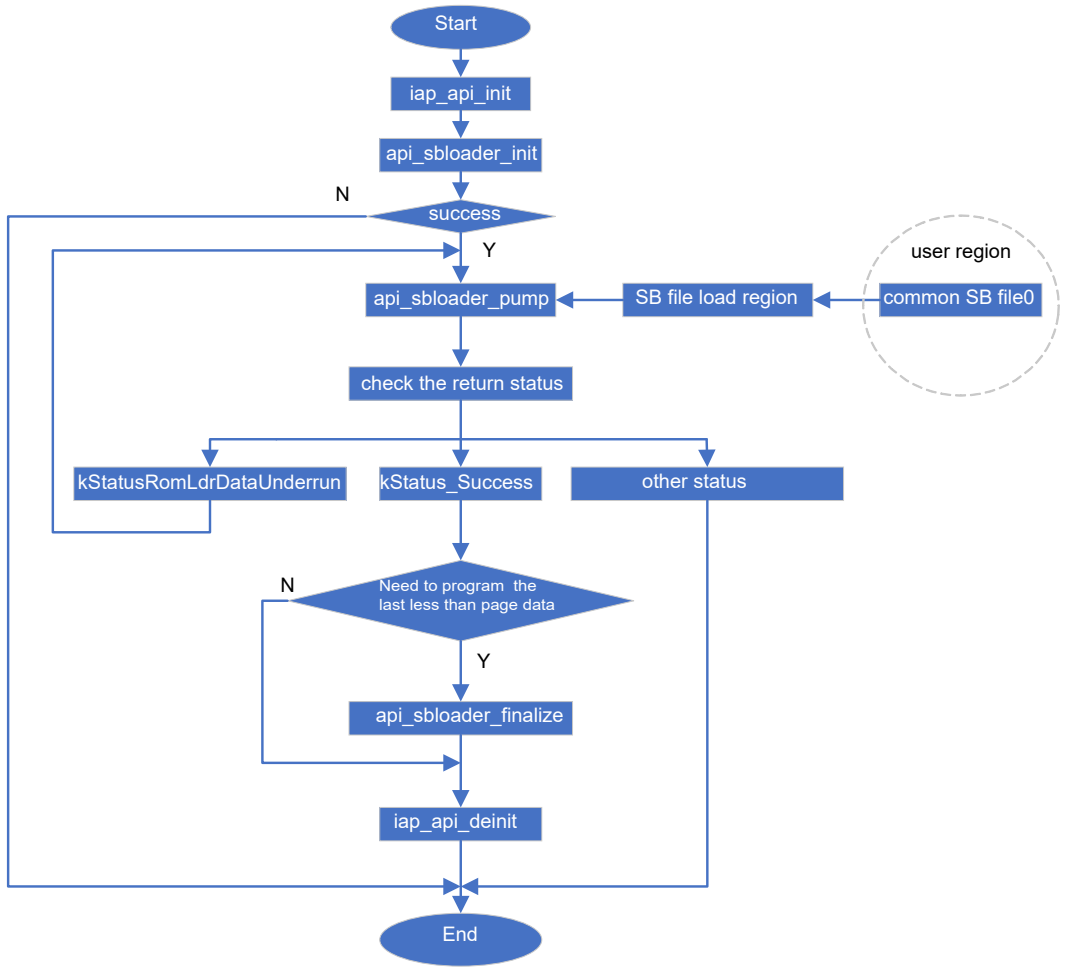


Figure 21. Flow diagram of the SBloader APIs call demo

The above flow diagram mainly introduces the SBloader APIs call flow. When the valid SB3.1 file has been generated and the necessary running environment(ROKTH and KeyBlob) has been setup up, the mainly details can be referred to the SB file chapter. From the flow diagram, the iap_api_init will be called first to initialize and configure the IAP API core context. And then call the api_sbloader_init to initialize the loader state machine. If the api_sbloader_init return status is success, then the spi_sbloader_pump will be called to process the SB file data stream, which is transferred from the user region.

In the running process of the api_sbloader_pump function, the return status includes three kinds of the results, which can be referred to in Table 89.Return and Error codes for IAP APIs. If the return status is kStatusRomLdrDataUnderrun, it indicates that more data in the common SB_file0 should be transferred to the api_sbloader_pump to process, and the process has not reached the end of the SB file data. When the kStatus_Success has been received, it indicates that common SB_file0 has been completely received and processed to update the firmware in the SB_file0 successfully. The other status will be thought that api_sbloader_pump API failed to execute. And the iap_api_deinit should be executed to clear the IAP APIs context parameters.

In addition, in the process of executing the SB file in the api_sbloader_pump, only the load address of the SB file data will be transferred to the api_sbloader_pump, the data of the SB file is executed with the packet chunk. Users should get and load the packet chunk of the SB file into the ram region like the following function get_data_from_sbfile(), and then call the api_sbloader_pump to process the ram data repeatedly until to the end of the SB file. In transferring the SB file chunk to the device, the user can use different channels to receive the chunk data like UART, SPI, I2C, USB, OTP interface.

10.2.4.6.1 api_sbloader_init

This API is used for initializing the sbloader state machine before calling the api_sbloader_pump. This API should be called after the iap_api_init API.

Prototype :

```
status_t api_sbloader_init(api_core_context_t *ctx);
```

Table 90. Parameters used in api_sbloader_init API

Parameter	Description
ctx	Pointer to IAP API core context structure.

Example:

```
uint32_t status = IAP_API_TREE-> sbloader_init (&apiCoreCtx);
```

See the possible status codes in Table 89. If the status is kStatus_Success return value, it means the sbloader runtime environment has been initialized successfully.

10.2.4.6.2 api_sbloader_pump

This API is used for handling the secure binary (SB3.1 format) data stream, which is used for image update, lifecycle advancing, etc. This API should be called after the iap_api_init and api_sbloader_init APIs.

Prototype :

```
status_t api_sbloader_pump(api_core_context_t *ctx, uint8_t *data, uint32_t length);
```

Table 91. Parameters used in api_sbloader_pump API

Parameter	Description
ctx	Pointer to IAP API core context structure.
data	Pointer to source data that is the sb file buffer data.
length	Points to the size of the process buffer data.

Example:

```
/* load the SB file chunk-by-chunk into chunk_processing_buffer */

get_chunk_from_SB_file(uint8_t * chunk_processing_buffer, uint8_t * SBfile_incoming_chunk, uint32_t
chunk_size);
status = IAP_API_TREE-> sbloader_pump(context, & chunk_processing_buffer, chunk_size);
```

See the possible status codes in Table 89. If the status is kStatus_Success return value, it means the sb data has been processed successfully.

10.2.4.6.3 api_sbloader_finalize

The API is used for finalizing the sbloader operations.

Prototype :

```
status_t api_sbloader_finalize(api_core_context_t *ctx)
```

Table 92. Parameters used in api_sbloader_finalize API

Parameter	Description
ctx	Pointer to IAP API core context structure.

Example:

```
status = IAP_API_TREE-> sbloader_finalize (context);
```

See the possible status codes in Table 89. If the status is kStatus_Success return value, it means the last sb data has programed successfully.

10.2.4.7 IAP API examples

The section will provide the example to introduce All the IAP APIs usage and help users understand and use more deeply.

```
#define ROM_API_TREE ((uint32_t*)0x1303fc00)
#define IAP_API ((iap_api_interface_t*)ROM_API_TREE[13])
api_core_context_t api_ctx = {0};
kp_api_init_param_t params = {0};
params.allocStart = 0x20350000u;
params.allocSize = 0x6000u;

status = IAP_API->api_init(&api_ctx,&params);
if (status != kStatus_Success)
{
return status;
}
status = IAP_API->sbloader_init(&api_ctx);
if (status != kStatus_Success)
{
return status;
}
uint8_t user_buf[1024]; uint32_t size;
while ((status = get_data_from_sbfile(&buffer, &size)) == kStatus_Success)
{
status = IAP_API ->sbloader_pump(&api_ctx, &buffer, size); if (status == kStatusRomLdrDataUnderrun)
{
/* continue to receive the data from SB file for processing */ debug_printf("Warning: Success to
execute ROM API erase, but need more data \n");
}
else if (status == kStatus_Success)
{
debug_printf("Completed: Success to execute ROM API erase\n"); break;    /* finish the SB file
firewarre update and break */
}
else if (status != kStatus_Success)
```

```
{
debug_printf("Error: Failed to run API kb_execute, status is hex(%x) \n", (uint32_t)status); return
status;    /* return the fail status or other actions */
}
}
status = IAP_API_TREE->sbloader_finalize (&api_ctx); if (status != kStatus_Success)
{
debug_printf("Error: Failed to sbloader_finalize\n"); return status;
}
status = IAP_API->api_deinit(&api_ctx);
```

Chapter 11
In-System Programming (ISP)

11.1 Overview

This chip includes in-system programming (ISP) functions to support image programming from the serial interface (UART, I2C, SPI), USB1 and USB0.

In-System Programming supports:

- UART, I2C, SPI, USB1 and USB0 peripheral interfaces
- Automatic detection of the active peripheral
- UART peripheral implements auto-baud detection
- Common packet-based protocol for all peripherals
- Protection of RAM used by the bootloader while it is running
- Retrieval the properties of the device, such as Flash and RAM size
- Support for external memory(XSPI Nor Flash/XSPI NAND Flash/eMMC/Serial Nor Flash/fuse) access
- Support for encrypted image download

Each ISP interface can be enabled or disabled by fuse. The following table lists fuse bits to control ISP entry methods.

Table 93. BOOT_CFG0 fuse bit configuration for ISP

Bit name	Bits	Default value	Definition	Description
ISP_UART_DIS	3	0	0 - Enabled 1 - Disabled	Disable LPUART interface during ISP/serial boot mode.
ISP_SPI_DIS	4	0	0 - Enabled 1 - Disabled	Disable LPSPI interface during ISP/serial boot mode.
ISP_I2C_DIS	5	0	0 - Enabled 1 - Disabled	Disable LPI2C interface during ISP/serial boot mode.
ISP_USB_DIS	6	0	0 - Enabled 1 - Disabled	Disable USB0 interface during ISP/serial boot mode.
ISP_eUSB_DIS	7	0	0 - Enabled 1 - Disabled	Disable USB1 (eUSB) interface during ISP/ serial boot mode.
ISP_MODE_PIN_DIS	8	0	0 - Enabled 1 - Disabled	Disable boot mode selection using ISP mode pins.
ISP_DM_DIS	9	0	0 - Enabled 1 - Disabled	Disable ISP mode entry through debug mailbox command.

Table continues on the next page...

Table 93. BOOT_CFG0 fuse bit configuration for ISP (continued)

Bit name	Bits	Default value	Definition	Description
ISP_FT_DIS	10	0	0 - Enabled 1 - Disabled	Disable ISP mode entry as a fall through if valid image is not found in boot media.
ISP_API_DIS	11	0	0 - Enabled 1 - Disabled	Disable ISP mode entry using ROM API.

11.1.1 Features

- Peripheral interfaces:
  - UART
  - I2C
  - SPI
  - USB0
  - USB1
- Automatic detection of the active peripheral
- UART peripheral implements auto-baud detection
- A common packet-based protocol for all peripherals
- Packet error detection and retransmission
- Protection of RAM used by the bootloader while it is running
- Retrieval of the properties of the device, such as flash memory and RAM size
- Multiple options for executing the bootloader, either at system startup or under application control at runtime
- Support for encrypted image downloading
- External flash memory access

11.2 Functional description

11.2.1 Bootloader

ISP stores the boot code in internal ROM. After a reset, the Arm processor starts its code execution from this memory. ISP executes the bootloader code every time the part is powered on, is reset, or is woken up from a deep power-down, low-power mode.

The bootloader provides a flash memory and OTP-eFuse programming utility that operates over a serial connection on the MCUs. It enables quick and easy programming of MCUs through the entire product lifecycle, including application development, final product manufacturing, and beyond. The host-side command line and GUI tools are available to communicate with the bootloader. Users can use host tools to upload and download application code and to perform manufacturing via the bootloader.

For the bootloader operation and boot pin, see Overview.

11.2.2 ISP (In-system programming)

Serial booting and other related functions are supported in several different ways:

- For details of the ISP protocol, see In-System programming protocol.
- For details of the ISP packet, see Bootloader packet types.

- For details of the ISP commands, see Bootloader command set.
- For details of UART In-System Programming, see UART ISP.
- For details of I^2C In-System Programming, see I2C In-System Programming.
- For details of SPI In-System Programming, see SPI In-System programming.
- For details of USB0 and USB1 In-System programming, see USB In-System Programming.

11.2.3 Memory map after reset

The boot ROM is located in the memory region beginning with the address 1300 0000h. Both the ISP and IAP software use parts of the on-chip RAM. Use of the RAM is described in ISP interrupt and SRAM use.

Based on the ISP fuse configuration or boot pin settings, the ROM enters ISP mode and can auto-detect activity on these interfaces:

- I2C
- SPI
- UART
- USB0/USB1

The auto-detect looks for activity on these interfaces and selects the appropriate interface when a properly formed frame is received. If an invalid frame is received, the data is discarded and scanning resumes. Communications for these interfaces are described in In-System programming protocol and Bootloader packet types.

11.2.4 ISP interrupt and SRAM use

11.2.4.1 Interrupts during IAP

When the user application code starts executing, the interrupt vectors from the SRAM are active. Before making any IAP call, you must disable the interrupts. The IAP code does not use or disable interrupts.

11.2.4.2 RAM used by the ISP command handler

ISP reserves the regions designated below for bootloader use when the bootloader is running.

Table 94. RAM used by ROM during ROM execution

RAM region	Purpose
2000_0000h–2001_4FFFh	• BSS • RW • Stack • Heap

11.2.5 ISP flow

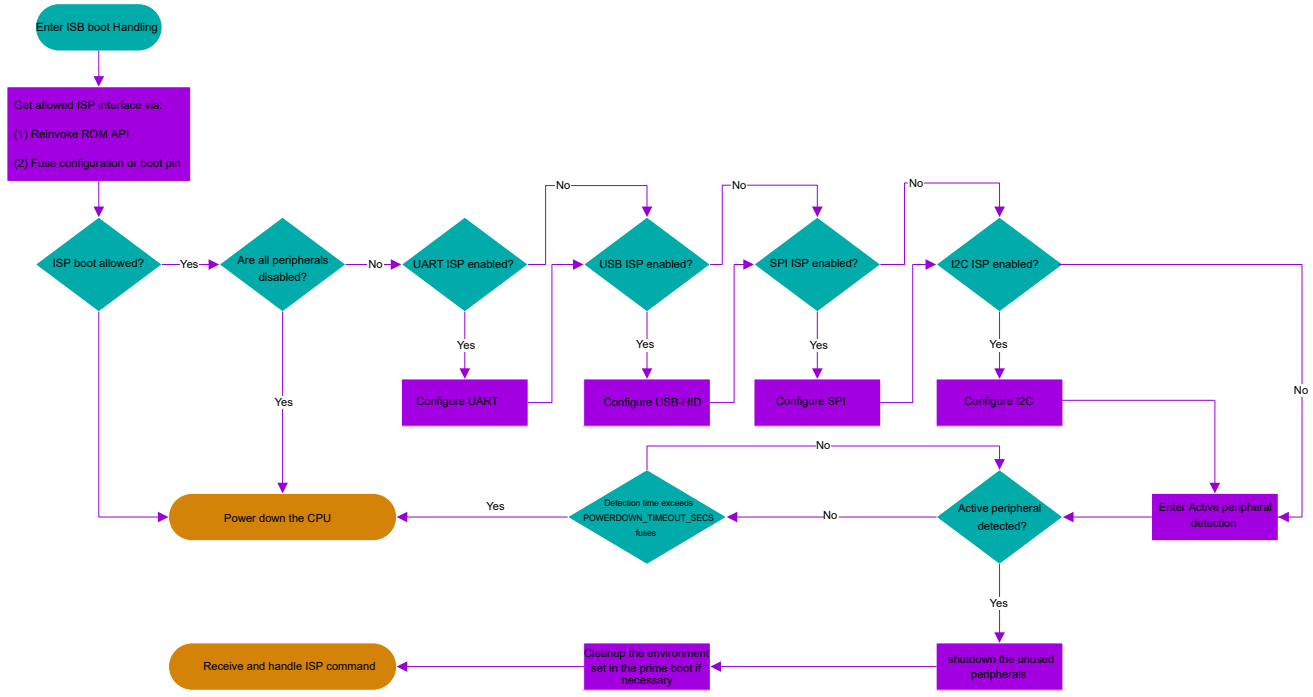


Figure 22. ISP flow

11.3 In-System programming protocol

This section explains the general protocol for the packet transfers between the host and the bootloader. The description includes the transfer of packets for different transactions, such as commands with no data phase and commands with incoming or outgoing data phase. The next section describes various packet types used in a transaction.

Each command sent from the host is replied to with a response command.

Commands may include an optional data phase.

- If the data phase is incoming (from the host to the bootloader), it is part of the original command.
- If the data phase is outgoing (from the bootloader to host), it is part of the response command.

11.3.1 Command with no data phase

The protocol for a command with no data phase contains:

1. Command packet (from the host).
2. Generic response command packet (to host).

The ACK sent in response to a command or a data packet can arrive at any time before, during, or after the command or data packet has been processed. Refer the diagrams in the Bootloader command set section.

11.3.2 Command with the incoming data phase

The protocol for a command with incoming data phase contains:

1. Command packet (from host) (kCommandFlag_HasDataPhase set).
2. Generic response command packet (to host).

3. Incoming data packets (from the host).
4. Generic response command packet to host.

Note:

- The host may not send any further packets while it is waiting for the response to a command.
- The data phase is aborted if the Generic Response packet prior to the start of the Data phase does not have a status of kStatus_Success.
- Data phases may be aborted by the receiving side by sending the final GenericResponse early with a status of kStatus_AbortDataPhase. The host may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet sent after the data phase includes the status of the entire operation.

11.3.3 Command with outgoing data phase

The protocol for a command with an outgoing data phase contains:

1. Command packet (from the host).
2. ReadMemory Response command packet (to host) (kCommandFlag_HasDataPhase set).
3. Outgoing data packets (to host).
4. Generic response command packet (to host).

- The data phase is considered part of the response command for the outgoing data phase sequence.
- The host may not send any further packets while the host is waiting for the response to a command.
- The data phase is aborted if the ReadMemory Response command packet, prior to the start of the data phase, does not contain the kCommandFlag_HasDataPhase flag.
- Data phases may be aborted by the device sending the final Generic Response early with a status of kStatus_AbortDataPhase. The sending side may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet sent after the data phase includes the status of the entire operation.

11.4 Bootloader packet types

11.4.1 Introduction

The bootloader device works in target mode. All data communications are initiated by a host, which is either a PC or an embedded host. The bootloader device is the target, which receives a command or data packet. All data communications between host and target are packetized.

There are six types of packets used:

- Ping packet.
- Ping Response packet.
- Framing packet.
- Command packet.
- Data packet.
- Response packet.

All fields in the packets are in little-endian byte order.

11.4.2 Ping packet

The Ping packet is the first packet sent from a host to the target to establish a connection on the selected peripheral in order to run autobaud detection. The Ping packet can be sent from host to target at any time that the target is expecting a command packet.

If the selected peripheral is UART, a Ping packet must be sent before any other communications. For other serial peripherals, it is optional.

In response to a Ping packet, the target sends a Ping response packet, discussed in the later sections.

Table 95. Ping packet format

Byte #	Value	Name
0	0x5A	Start byte
1	0xA6	Ping

11.4.3 Ping response packet

The target sends a Ping response packet back to the host after receiving a Ping packet. If communication is over a UART peripheral, the target uses the incoming Ping packet to determine the baud rate before replying with the Ping response packet. After the Ping response packet is received by the host, the connection is established, and the host starts sending commands to the target.

Table 96. Ping response packet format

Byte	Value	Parameter
0	0x5A	Start byte
1	0xA7	Ping response code
2	0x00	Protocol bugfix
3	0x03	Protocol minor
4	0x01	Protocol major
5	0x50	Protocol name = 'P' (0x50)
6	0x00	Options low
7	0x00	Options high
8	0xfb	CRC16 low
9	0x40	CRC16 high

For the UART peripheral, it must be sent by the host when a connection is first established, in order to run autobaud. For other serial peripherals, it is optional but recommended to determine the serial protocol version. The version number is in the same format as the bootloader version number returned by the GetProperty command.

11.4.4 Framing packet

The framing packet is used for flow control and error detection for the communications links that do not have such features built in. The framing packet structure sits between the link layer and the command layer. It wraps command and data packets as well.

Every framing packet containing data sent in one direction results in a synchronizing response framing packet in the opposite direction.

The framing packet described in this section is used for serial peripherals including the UART, CAN, I^2C, and SPI. The USB HID peripheral does not use framing packets. Instead, the packetization inherent in the USB protocol itself is used.

Table 97. Framing packet format

Byte	Value	Parameter	Description
0	0x5A	Start byte	
1		PacketType	
2		Length_low	Length is a 16-bit field that specifies the entire command or data packet size in bytes.
3		Length_high	
4		Crc16_low	This is a 16-bit field. The CRC16 value covers entire framing packet, including the start byte and command or data packets, but does not include the CRC bytes. See CRC16 algorithm.
5		Crc16_high	
6....n		Command or Data packet payload	

A special framing packet that contains only a start byte and a packet type is used for synchronization between the host and target.

Table 98. Special framing packet format

Byte	Value	Parameter
0	0x5A	Start byte
1	0xA*n*	packetType

The Packet Type field specifies the type of the packet from one of the defined types (below):

Table 99. Packet type field

Packet type	Name	Description
0xA1	kFramingPacketType_Ack	The previous packet was received successfully; the sending of more packets is allowed.
0xA2	kFramingPacketType_Nak	The previous packet was corrupted and must be re-sent.
0xA3	kFramingPacketType_AckAbort	Data phase is being aborted.

Table 100. Packet type field

Packet type	Name	Description
0xA4	kFramingPacketType_Command	The framing packet contains a command packet payload.
0xA5	kFramingPacketType_Data	The framing packet contains a data packet payload.

Table continues on the next page...

Table 100. Packet type field (continued)

Packet type	Name	Description
0xA6	kFramingPacketType_Ping	Sent to verify the other side is alive. Also used for UART autobaud.
0xA7	kFramingPacketType_PingResponse	A response to Ping. It contains the framing protocol version number and options.

11.4.5 CRC16 algorithm

The CRC is computed over each byte in the framing packet header, excluding the CRC16 field itself, and all of the payload bytes. The CRC algorithm is the XMODEM variant of CRC16.

The characteristics of the XMODEM variants are:

Table 101. CRC16 algorithm

Width	16
Polynomial	0x1021
Init value	0x0000
Reflect in	False
Reflect out	False
Xor out	0x0000
Check result	0x31c3

The check result is computed by running the ASCII character sequence "123456789" through the algorithm.

```
uint16_t crc16_update(const uint8_t * src, uint32_t lengthInBytes)
{
    uint32_t crc = 0;
    uint32_t j;
    for (j=0; j < lengthInBytes; ++j)
        {
        uint32_t i;
        uint32_t byte = src[j];
        crc ^= byte << 8;
        for (i = 0; i < 8; ++i)
        {
                    uint32_t temp = crc << 1;
            if (crc & 0x8000)
            {
                                temp ^= 0x1021;
            }
                        crc = temp;
        }
    }
    return crc;
}
```

11.4.6 Command packet

The command packet carries a 32-bit command header and a list of 32-bit parameters.

Table 102. Command packet format

Command Packet Format (32 bytes)										
Command Header (4 bytes)				28 bytes for Parameters (Max 7 parameters)						
Tag	Flags	Rsvd	Param Count	Param 1 (32-bit)	Param 2 (32-bit)	Param 3 (32-bit)	Param 4 (32-bit)	Param 5 (32-bit)	Param 6 (32-bit)	Param 7 (32-bit)

Table 103. Command header format

Byte #	Command header field	Description
0	Command or Response tag	The command header is 4 bytes long with these fields.
1	Flags	
2	Reserved. Should be 0x00.	
3	ParameterCount	

The header is followed by 32-bit parameters up to the value of the ParameterCount field specified in the header. Because a command packet is 32 bytes long, only seven parameters can fit into the command packet.

- **Flags:** Each command packet contains a flag byte. Only bit 0 of the flag byte is used. If bit 0 of the flag byte is set to 1, then data packets follow the command sequence. The number of bytes that are transferred in the data phase is determined by a command specific parameter in the parameters array.
- **ParameterCount:** The number of parameters included in the command packet.
- **Parameters:** The parameters are word-length (32 bits). With the default maximum packet size of 32 bytes, a command packet can contain up to seven parameters.

Command packets are also used by the target to send responses back to the host. As mentioned earlier, command packets and data packets are embedded into framing packets for all of the transfers.

Table 104. Command tags

Command tag	Name	Description
0x01	FlashEraseAll	The command tag specifies one of the commands supported by the bootloader. The valid command tags for the bootloader are listed here.[1]
0x02	FlashEraseRegion	
0x03	ReadMemory	
0x04	WriteMemory	
0x05	FillMemory	
0x06	Reserved	

Table continues on the next page...

Table 104. Command tags (continued)

Command tag	Name	Description
0x07	GetProperty	
0x08	ReceiveSbFile	
0x09	Execute	
0x0A	Reserved	
0x0B	Reset	
0x0C	SetProperty	
0x0D	Reserved	
0x0E	eFuseProgram/FlashProgramOnce	
0x0F	eFuseRead/FlashReadOnce	
0x10	Reserved	
0x11	ConfigureMemory	
0x12	Reserved	
0x13	Reserved	
0x14	Reserved	
0x15	Reserved	
0x16	TrustProvisioning	

1. The GetProperty, Reset, TrustProvisioning, SetProperty, and ReceiveSbFile are allowed in limited ISP mode.

Table 105. Response tags

Response tag	Name	Description
0xA0	GenericResponse	The response tag specifies one of the responses the bootloader (target) returns to the host. The valid response tags are listed here.
0xA3	ReadMemoryResponse	
0xA7	GetPropertyResponse (used for sending responses to GetProperty command only)	
0xA3	ReadMemoryResponse (used for sending responses to ReadMemory command only)	

Table continues on the next page...

Table 105. Response tags (continued)

Response tag	Name	Description
0xAF	FlashReadOnceResponse (used for sending responses to FlashReadOnce command only)	
0xB6	TrustProvisioningResponse(used for sending response to TrustProvisioning command only)	

11.4.7 Response packet

The responses are carried using the same command packet format wrapped with framing packet data. Types of responses include:

- GenericResponse.
- GetPropertyResponse.
- ReadMemoryResponse.
- FlashReadOnceResponse.
- TrustProvisioningResponse

GenericResponse: After the bootloader has processed a command, the bootloader sends a generic response with status and command tag information to the host. The generic response is the last packet in the command protocol sequence. The generic response packet contains the framing packet data and the command packet data (with generic response tag = 0xA0) and a list of parameters (defined in the next section). The parameter count field in the header is always set to 2, for status code and command tag parameters.

Table 106. GenericResponse parameters

Byte #	Parameter	Description
0 - 3	Status code	Bootloader Status Error Codes are errors encountered during the execution of a command by the target. If a command succeeds, then a kStatus_Success code is returned.
4 - 7	Command tag	The Command tag parameter identifies the response to the command sent by the host.

GetPropertyResponse: The GetPropertyResponse packet is sent by the target in response to the host query that uses the GetProperty command. The GetPropertyResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a GetPropertyResponse tag value (0xA7).

The parameter count field in the header is set to greater than 1, to always include the status code and one or many property values.

Table 107. GetPropertyResponse parameters

Byte #	Value	Parameter
0 - 3		Status code
4 - 7		Property value

Table continues on the next page...

Table 107. GetPropertyResponse parameters (continued)

Byte #	Value	Parameter
. . .		. . .
		Can be up to maximum 6 property values, limited to the size of the 32-bit command packet and property type.

ReadMemoryResponse: The ReadMemoryResponse packet is sent by the target in response to the host sending a ReadMemory command. The ReadMemoryResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a ReadMemoryResponse tag value (0xA3), the flags field set to kCommandFlag_HasDataPhase (1).

The parameter count set to two for the status code and the data byte count parameters shown below.

Table 108. ReadMemoryResponse parameters

Byte #	Parameter	Description
0 - 3	Status code	The status of the associated Read Memory command.
4 - 7	Data byte count	The number of bytes sent in the data phase.

FlashReadOnceResponse: The FlashReadOnceResponse packet is sent by the target in response to the host sending a FlashReadOnce command. The FlashReadOnceResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a FlashReadOnceResponse tag value (0xAF), and the flags field set to 0. The parameter count is set to 2 plus *the number of words* requested to be read in the FlashReadOnceCommand.

Table 109. FlashReadOnceResponse parameters

Byte #	Value	Parameter
0 - 3		Status code
4 - 7		Byte count to read
. . .		. . .
		Can be up to 20 bytes of requested read data.

TrustProvisioningResponse: The TrustProvisioningResponse packet is sent by the target in response to the host sending a TrustProvisioning command. The TrustProvisioningResponse packet contains the framing packet data and command packet data, with the command/response tag set to a TrustProvisioningResponse tag value (0xB6), and the flags field set to 0. The parameter count various for different operations.

11.5 Bootloader command set

11.5.1 Introduction

All bootloader commands follow the command packet format wrapped by the framing packet as explained in previous sections.

For a list of status codes returned by bootloader see Bootloader Status Error Codes.

11.5.2 GetProperty command

The GetProperty command is used to query the bootloader about various properties and settings. Each supported property has a unique 32-bit tag associated with it. The tag occupies the first parameter of the command packet. The target returns a

GetPropertyResponse packet with the property values for the property identified with the tag in the GetProperty command.

Properties are the defined units of data that can be accessed with the GetProperty or SetProperty commands. Properties may be read-only or read-write. All read-write properties are 32-bit integers, so they can easily be carried in a command parameter.

The 32-bit property tag is the only parameter required for GetProperty command.

Table 110. Parameters for GetProperty Command

Byte #	Parameter
0 - 3	Property tag (which can be get from the get-property 0 command)
4 - 7	External Memory Identifier (only applies to get property for external memory, or status identifier if the property tag is equal to 8).

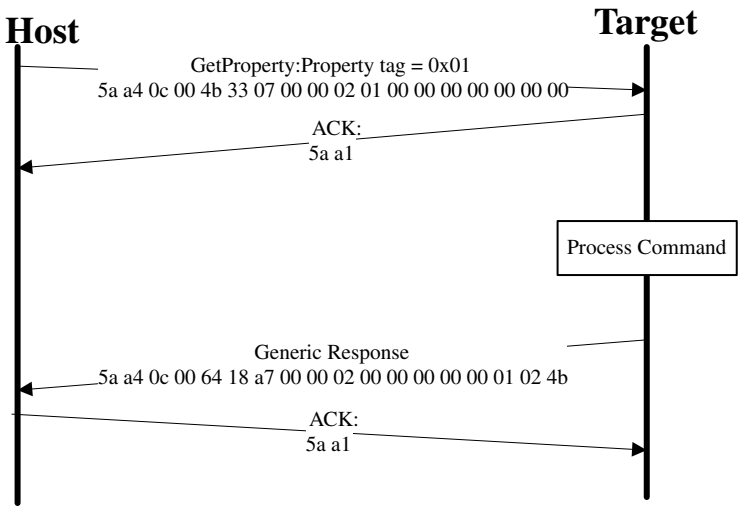


Figure 23. Protocol sequence for GetProperty command

Table 111. GetProperty command packet format (example)

GetProperty	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x0C 0x00
	Crc16	0x4B 0x33

Table 112. GetProperty command packet format (example)

GetProperty	Parameter	Value
Command packet	CommandTag	0x07 – GetProperty
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x02
	PropertyTag	0x00000001 - CurrentVersion
	Memory ID	0x00000000 - Internal Flash

The GetProperty command has no data phase.

Response: In response to a GetProperty command, the target sends a

GetPropertyResponse packet with the response tag set to 0xA7. The parameter count indicates the number of parameters sent for the property values, with the first parameter showing status code 0, followed by the property value(s). Table 113 shows an example of a GetProperty response packet.

Table 113. GetProperty response packet format (example)

GetPropertyResponse	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x0c 0x00 (12 bytes)
	Crc16	0x07 0x7a
Command packet	ResponseTag	0xA7
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x02
	Status	0x00000000
	PropertyValue	0x0000014b - CurrentVersion

11.5.3 SetProperty command

The SetProperty command is used to change or alter the values of the properties or options of the bootloader. The command accepts the same property tags used with the GetProperty command. However, only some properties are writable. If an attempt to write a read-only property is made, an error is returned indicating the property is read-only and cannot be changed.

The property tag and the new value to set are the two parameters required for the SetProperty command.

Table 114. Parameters for SetProperty command

Byte #	Command
0 - 3	Property tag
4 - 7	Property value

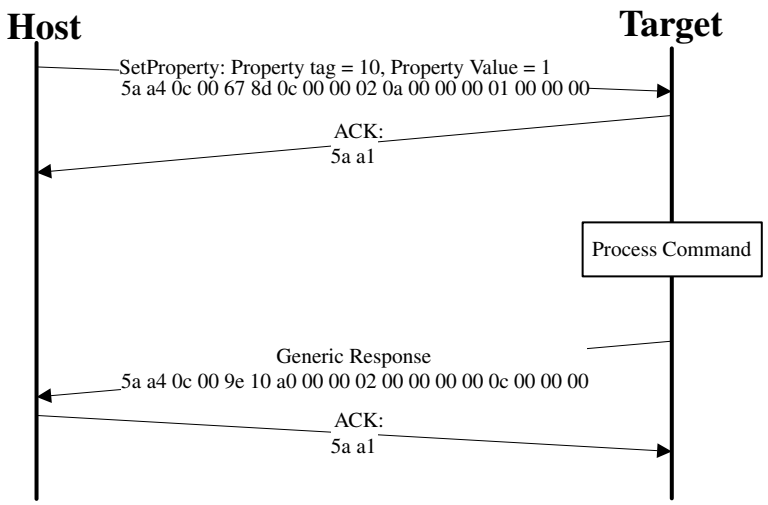


Figure 24. Protocol sequence for SetProperty Command

Table 115. SetProperty command packet format (example)

SetProperty	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x0C 0x00
	Crc16	0x67 0x8D
Command packet	CommandTag	0x0C – SetProperty with property tag 10
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x02
	PropertyTag	0x0000000A - VerifyWrites
	PropertyValue	0x00000001

The SetProperty command has no data phase.

Response: The target returns a GenericResponse packet with one of following status codes:

Table 116. SetProperty response status codes

Status code
kStatus_Success
kStatus_ReadOnly
kStatus_UnknownProperty
kStatus_InvalidArgument

11.5.4 FlashEraseAll command

The FlashEraseAll command performs an erase of the entire flash memory. If any flash regions are protected, then the FlashEraseAll command fails and returns an error status code. The Command tag for FlashEraseAll command is 0x01 set in the commandTag field of the command packet.

The FlashEraseAll command requires memory ID. If memory ID is not specified, the internal flash (memory ID =0) will be selected as default.

Table 117. Parameter for FlashEraseAll command

Byte #	Parameter	
0-3	Memory ID	
	0x000	Internal Flash
	0x09	XSPI NOR
	0x110	Serial NOR/EEPROM through SPI

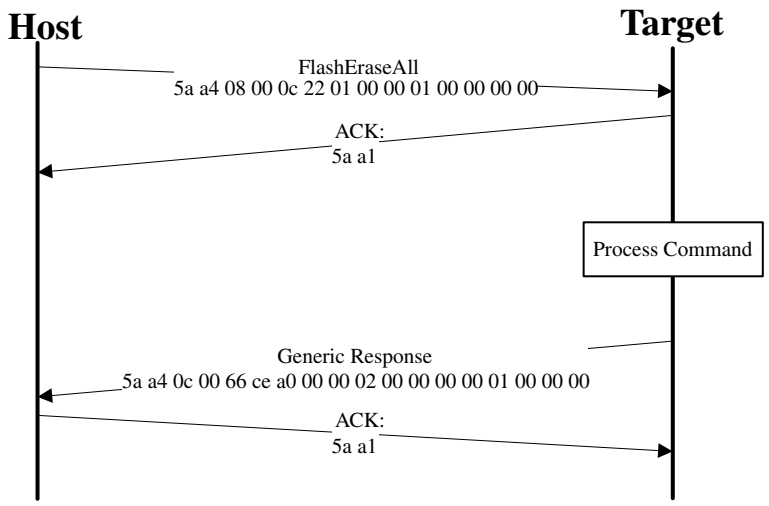


Figure 25. FlashEraseAll command

Table 118. FlashEraseAll command packet format (example)

FlashEraseAll	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x08 0x00
	Crc16	0x0C 0x22
Command packet	CommandTag	0x01 - FlashEraseAll
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x01
	Memory ID	Refer the above table

The FlashEraseAll command has no data phase.

Response: The target returns a GenericResponse packet with status code either set to kStatus_Success for successful execution of the command or set to an appropriate error status code.

11.5.5 FlashEraseRegion command

The FlashEraseRegion command performs an erase of one or more sectors of the flash memory.

The start address, and number of bytes are the two parameters required for the FlashEraseRegion command. The start and byte count parameters must be 4-byte aligned ([1:0] = 00), or the FlashEraseRegion command fails and returns kStatus_FlashAlignmentError (101). If the region specified does not fit in the flash memory space, the FlashEraseRegion command fails and returns kStatus_FlashAddressError (102). If any part of the region specified is protected, the FlashEraseRegion command fails and returns kStatus_MemoryRangeInvalid (10200).

Table 119. Parameter for FlashEraseRegion command

Byte #	Parameter
0-3	Start address
4 - 7	Byte count
8 - 11	Memory ID

The FlashEraseRegion command has no data phase.

Response: The target returns a GenericResponse packet with one of the following error status codes.

Table 120. FlashEraseRegion response status codes

Status code
kStatus_Success (0).

Table continues on the next page...

Table 120. FlashEraseRegion response status codes (continued)

Status code
kStatus_MemoryRangeInvalid (10200).
kStatus_FlashAlignmentError (101).
kStatus_IFlashAddressError (102).

Table 121. FlashEraseRegion response status codes

Status code
kStatus_FlashAccessError (103).
kStatus_FlashProtectionViolation (104).
kStatus_FlashCommandFailure (105).

11.5.6 ReadMemory command

The ReadMemory command returns the contents of memory at the given address, for a specified number of bytes. This command can read any region of memory accessible by the CPU and not protected by security.

The start address, and number of bytes are the two parameters required for ReadMemory command. The memory ID is optional. Internal memory will be selected as default if memory ID is not specified.

Table 122. Parameter for read memory command

Byte #	Parameter	Description
0-3	Start address	Start address of memory to read from.
4-7	Byte count	Number of bytes to read and return to caller.
8-11	Memory ID	Internal or external memory Identifier.

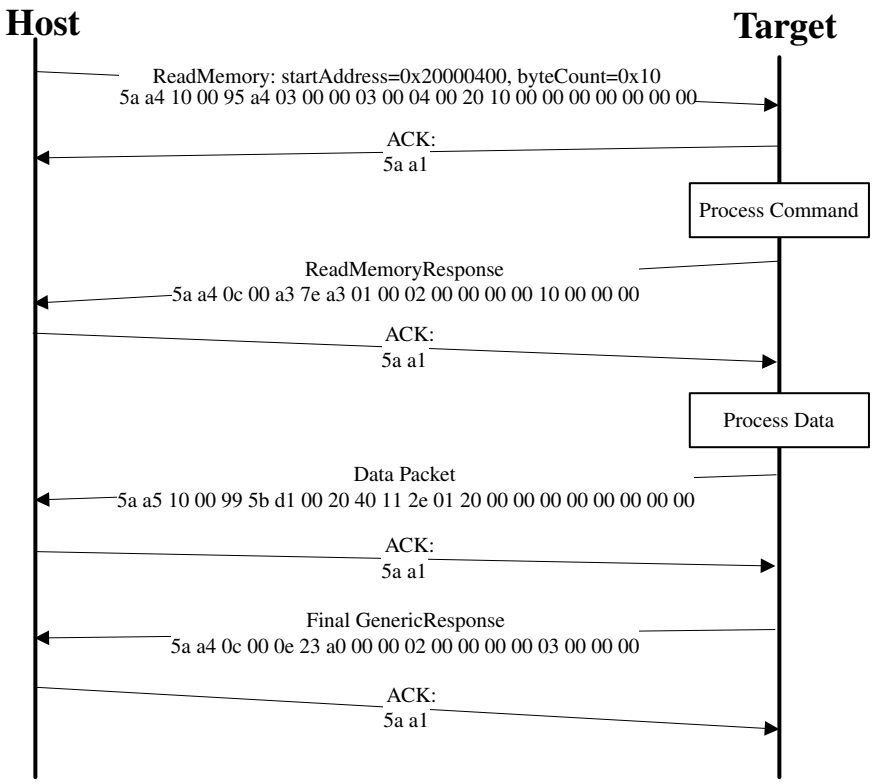


Figure 26. Command sequence for ReadMemory

Table 123. ReadMemory command packet format (example)

ReadMemory	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x10 0x00
	Crc16	0xF4 0x1B
Command packet	CommandTag	0x03 - ReadMemory
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x03
	StartAddress	0x20000400
	ByteCount	0x00000064
	Memory ID	0x0

Data Phase: The ReadMemory command has a data phase. The host needs to pull data packets until the number of bytes of data specified in the byteCount parameter of ReadMemory command are received by host.

Response: The target returns a GenericResponse packet with a status code either set to kStatus_Success upon successful execution of the command or set to an appropriate error status code

11.5.7 WriteMemory command

The WriteMemory command writes data provided in the data phase to a specified range of bytes in memory (flash or RAM). However, if flash protection is enabled, then writes to protected sectors fail.

Special care must be taken when writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion command.
- Writing to flash requires the start address to be page aligned.
- The byte count is rounded up to a page size, and trailing bytes are filled with the flash erase pattern (0xff).
- If the VerifyWrites property is set to true, then writes to flash also performs a flash verify program operation.

When writing to RAM, the start address does not need to be aligned, and the data is not padded.

The start address and number of bytes are the two parameters required for WriteMemory command. The memory ID is optional. Internal memory will be selected as default if memory ID is not specified.

Table 124. Parameters for WriteMemory command

Byte #	Command
0-3	Start address
4-7	Byte count
8-11	Memory ID

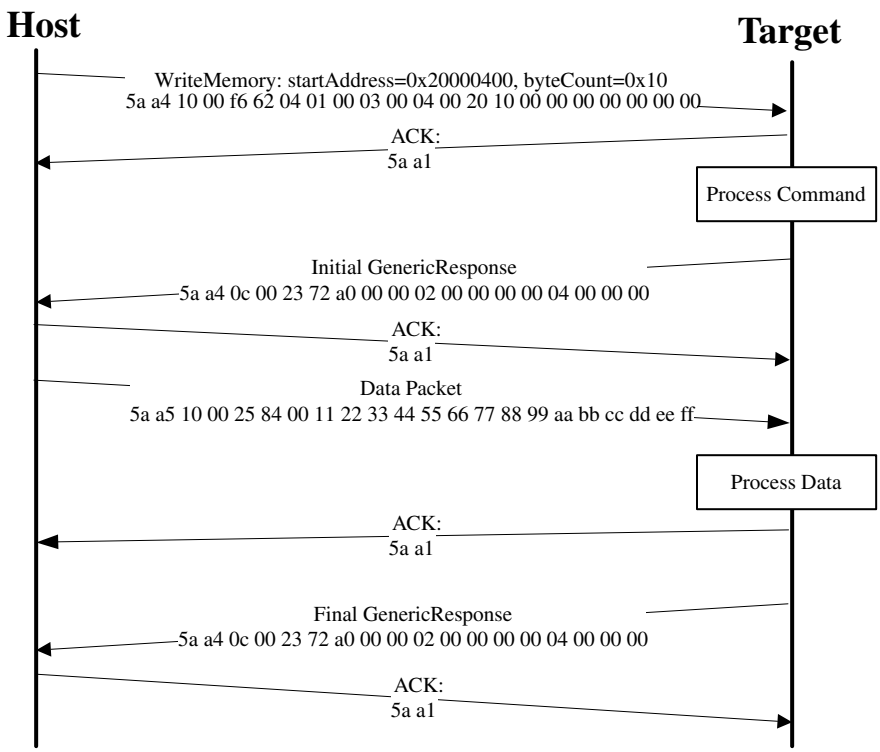


Figure 27. Protocol sequence for WriteMemory command

Table 125. WriteMemory command packet format (example)

WriteMemory	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x10 0x00
	Crc16	0x97 0xDD
Command packet	CommandTag	0x04 - WriteMemory
	Flags	0x01
	Reserved	0x00
	ParameterCount	0x03
	StartAddress	0x20000400
	ByteCount	0x00000064
	Memory ID	0x0

Data Phase: The WriteMemory command has a data phase; the host sends data packets until the number of bytes of data specified in the byteCount parameter of the WriteMemory command are received by the target.

Response: The target returns a GenericResponse packet with a status code set to kStatus_Success upon successful execution of the command, or to an appropriate error status code.

11.5.8 FillMemory command

The FillMemory command fills a range of bytes in memory with a data pattern. It follows the same rules as the WriteMemory command. The difference between FillMemory and WriteMemory is that a data pattern is included in FillMemory command parameter, and there is no data phase for the FillMemory command, while WriteMemory does have a data phase.

Table 126. Parameters for FillMemory command

Byte #	Command
0-3	Start address of memory to fill.
4-7	• Number of bytes to write with the pattern. • The start address should be 32-bit aligned. • The number of bytes must be evenly divisible by 4.
8-11	32-bit pattern.

- To fill with a byte pattern (8-bit), the byte must be replicated four times in the 32-bit pattern.
- To fill with a short pattern (16-bit), the short value must be replicated two times in the 32-bit pattern.

For example, to fill a byte value with 0xFE, the word pattern is 0xFEFEFEFE; to fill a short value 0x5AFE, the word pattern is 0x5AFE5AFE.

Special care must be taken while writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll, or FlashEraseRegion command.
- Writing to flash requires the start address to be 4-byte aligned ([1:0] = 00).
- If the VerifyWrites property is set to true, then writes to flash also performs a flash verify program operation.

When writing to RAM, the start address does not need to be aligned, and the data is not padded.

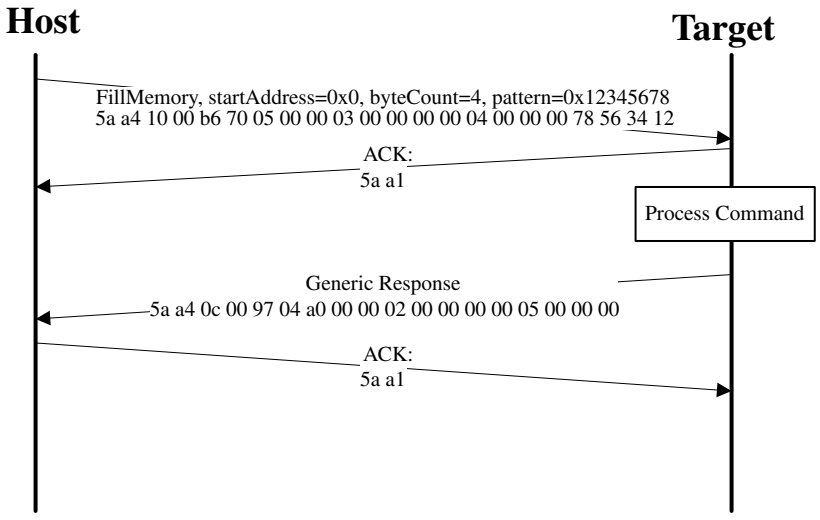


Figure 28. Protocol sequence for FillMemory command

Table 127. FillMemory command packet format (example)

FillMemory	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x10 0x00
	Crc16	0xE4 0x57
Command packet	CommandTag	0x05 – FillMemory
	Flags	0x01
	Reserved	0x00
	ParameterCount	0x03
	StartAddress	0x00007000
	ByteCount	0x00000800
	PatternWord	0x12345678

The FillMemory command has no data phase.

Response: upon successful execution of the command, the target (bootloader) returns a GenericResponse packet with a status code set to kStatus_Success, or to an appropriate error status code.

11.5.9 Execute command

The Execute command results in the bootloader setting the program counter to the code at the provided jump address, R0 to the provided argument, and a Stack pointer to the provided stack pointer address. Prior to the jump, the system is returned to the reset state.

The Jump address, function argument pointer, and stack pointer are the parameters required for the Execute command. If the stack pointer is set to zero, the called code is responsible for setting the processor stack pointer before using the stack.

Table 128. Parameters for Execute command

Byte #	Command
0-3	Jump address.
4-7	Argument word.
8-11	Stack pointer address.

The Execute command has no data phase.

Response: Before executing the Execute command, the target validates the parameters and return a GenericResponse packet with a status code either set to kStatus_Success or an appropriate error status code.

11.5.10 Reset command

The Reset command results in the bootloader resetting the chip.

The Reset command requires no parameters.

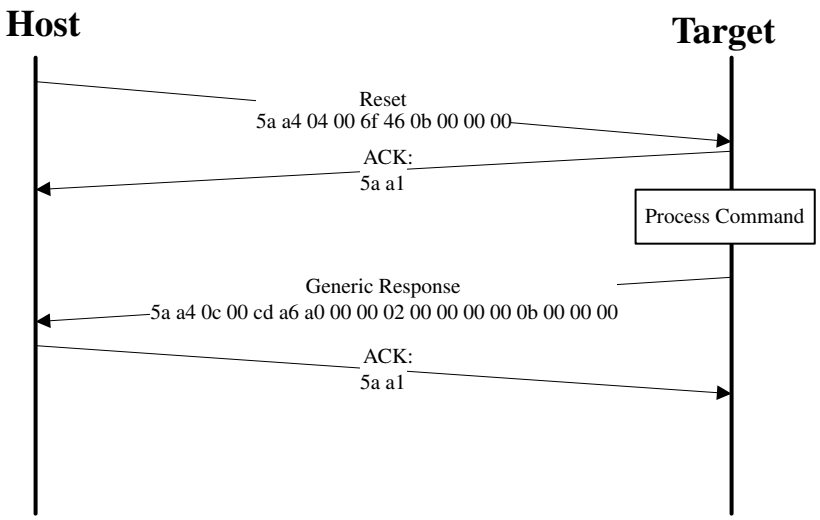


Figure 29. Protocol sequence for Reset command

Table 129. Reset command packet format (example)

Reset	Parameter	Value
Framing packet	Start byte	0x5A
	PacketType	0xA4, kFramingPacketType_Command
	Length	0x04 0x00
	Crc16	0x6F 0x46
Command packet	CommandTag	0x0B - reset
	Flags	0x00
	Reserved	0x00
	ParameterCount	0x03

The Reset command has no data phase.

Response: The target returns a GenericResponse packet with status code set to kStatus_Success, before resetting the chip.

The reset command can also be used to switch boot from flash after successful flash image provisioning via ROM bootloader. After issuing the reset command, allow five seconds for the user application to start running from Flash.

11.5.11 eFuseProgramOnce/FlashProgramOnce command

The FlashProgramOnce command writes data (that is provided in a command packet) to a specified range of bytes in the program once field. Special care must be taken when writing to the program once field.

• The program once field only supports programming once, so any attempted to reprogram a program-once field gets an error response.

• Writing to the program once field requires the byte count to be 4-byte aligned or 8-byte aligned. The FlashProgramOnce command uses three parameters: index 2, byteCount, data.

Table 130. Parameters for FlashProgramOnce Command

Byte #	Command
0 - 3	Index of program once field
4 - 7	Byte count (must be evenly divisible by 4)
8 - 11	Data

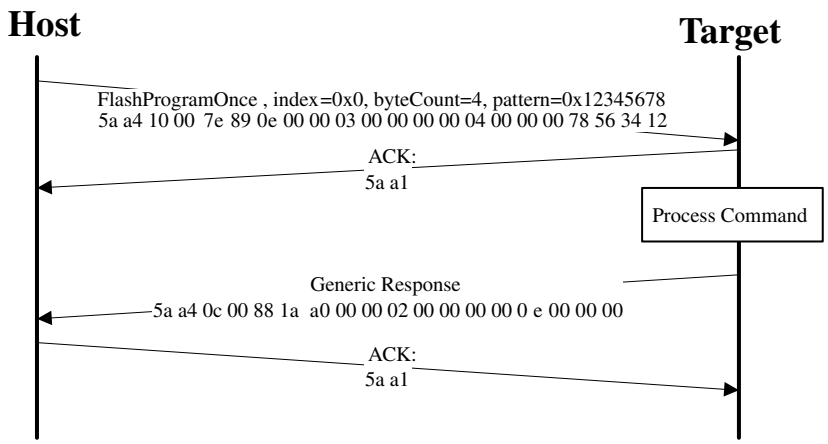


Figure 30. Protocol Sequence for FlashProgramOnce Command

Table 131. FlashProgramOnce Command Packet Format

FlashProgramOnce	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x10 0x00
	crc16	0x7E4 0x89
Command packet	commandTag	0x0E – FlashProgramOnce
	flags	0
	reserved	0
	parameterCount	3

Table continues on the next page...

Table 131. FlashProgramOnce Command Packet Format (continued)

FlashProgramOnce	Parameter	Value
	index	0x0000_0051 (the customer last efuse index)
	byteCount	0x0000_0004
	Data	0x1234_5678

Response: upon successful execution of the command, the target (Kinetis bootloader) returns a GenericResponse packet with a status code set to kStatus_Success, or to an appropriate error status code.

11.5.12 eFuseReadOnce / FlashReadOnce command

The FlashReadOnce command returns the contents of the program once field by given index and byte count. The FlashReadOnce command uses 2 parameters: index and byteCount. eFuseReadOnce command uses only one parameter index.

Table 132. Parameters for FlashReadOnce Command

Byte #	Parameter	Description
0 - 3	Index	Index of the program once field (to read from)
4 - 7	byteCount	Number of bytes to read and return to the caller

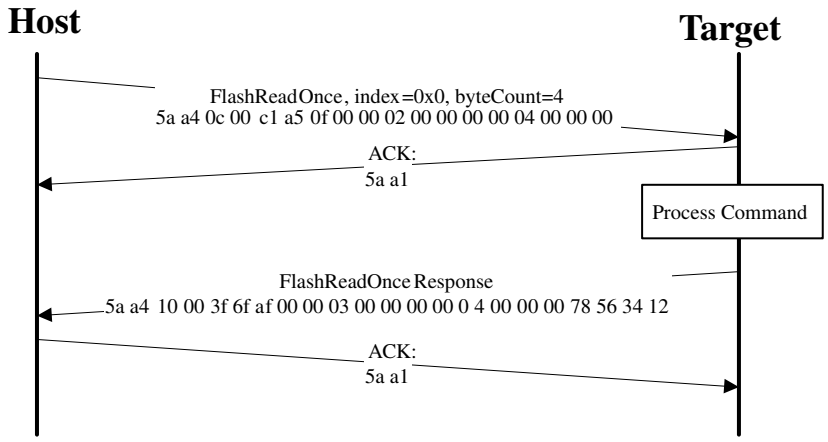


Figure 31. Protocol Sequence for FlashReadOnce Command

Table 133. FlashReadOnce Command Packet Format

FlashReadOnce	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	Length	0x0C 0x00

Table continues on the next page...

Table 133. FlashReadOnce Command Packet Format (continued)

FlashReadOnce	Parameter	Value
	Crc	0xC1 0xA5
Command packet	commandTag	0x0F – FlashReadOnce
	Flags	0x00
	Reserved	0x00
	parameterCount	0x02
	Index	0x0000_0000
	byteCount	0x0000_0004

Table 134. FlashReadOnce Response Format

FlashReadOnce Response	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	Length	0x10 0x00
	Crc	0x3F 0x6F
Command packet	commandTag	0xAF
	Flags	0x00
	Reserved	0x00
	parameterCount	0x03
	Status	0x0000_0000
	byteCount	0x0000_0004
	Data	0x1234_5678

Response: upon successful execution of the command, the target returns a FlashReadOnceResponse packet with a status code set to kStatus_Success, a byte count and corresponding data read from Program Once Field upon successful execution of the command or returns with a status code set to an appropriate error status code and a byte count set to 0.

11.5.13 ConfigureMemory command

The ConfigureMemory command configures the internal/external memory device using a pre-programmed configuration block. The parameters passed in the command are the memory ID, and then the memory address from which the configuration data can be loaded from. Options for loading the data can be a scenario where the configuration data is written to a RAM or flash location and then this command directs the bootloader to use the data at that location to configure the external memory devices.

Table 135. Parameters for ConfigureMemory command

Byte #	Command
0-3	Memory ID.
4-7	Configuration block address.

Response: The target (Bootloader) returns a GenericResponse packet with a status code either set to kStatus_Success upon successful execution of the command or set to an appropriate error code.

The following table shows the supported memory IDs.

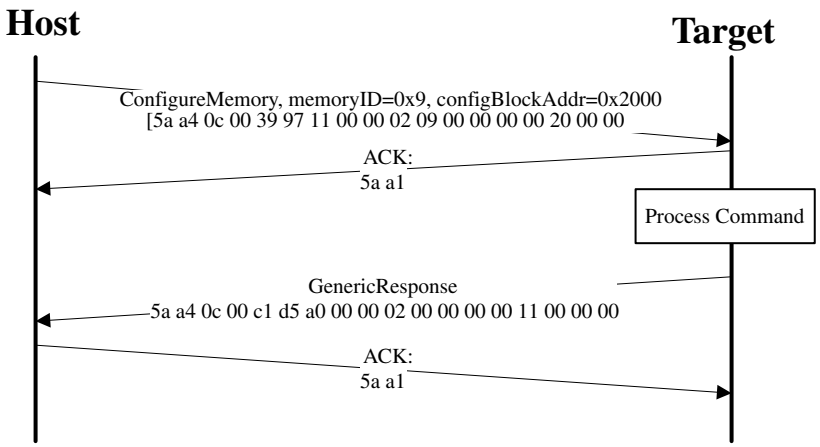


Figure 32. Protocol sequence for ConfigureMemory command

Table 136. Supported memory IDs

Memory ID	Description
0	Internal RAM/FLASH (Used for the PRINCE configuration).
0x9	External XSPI device. See Serial NOR Flash through XSPI.
0x110	External 1-bit SPI NOR FLASH device. See 1-bit Serial NOR flash through SPI.

11.5.14 ReceiveSBFile command

The Receive SB File command (ReceiveSbFile) starts the transfer of an SB file to the target. The command only specifies the size in bytes of the SB file that is sent in the data phase. The SB file is processed as it is received by the bootloader.

Table 137. Parameters for Receive SB File command

Byte #	Command
0-3	Byte count

Data Phase: The Receive SB file command has a data phase; the host sends data packets until the number of bytes of data specified in the byteCount parameter of the Receive SB File command are received by the target.

Response: The target returns a GenericResponse packet with a status code set to the kStatus_Success upon successful execution of the command or set to an appropriate error code.

NOTE

Only the byte address mode is supported in sbloder for NAND device.

11.5.15 Get/SetProperty command properties

This section lists the properties of the GetProperty and SetProperty commands.

Table 138. Properties used by Get/SetProperty commands, sorted by values

Property	Writable	Tag value	Size (byte)	Description
CurrentVersion	No	01h	4	Current bootloader version.
AvailablePeripherals	No	02h	4	The set of peripherals supported on this chip.
AvailableCommands	No	07h	4	The set of commands supported by the bootloader
Check Status	No	08h	4	32-bit return value for CRC Check 10401 - Application CRC check failed 10402 - Application CRC check is inactive 10403 - Application CRC check is invalid
MaxPacketSize	No	0Bh	4	Maximum supported packet size for the currently active peripheral interface.
ReservedRegions	No	0Ch	8*n	List of memory regions reserved by the bootloader. Returned as value pairs (<start-address-of-region>, <end-address-of-region>). If HasDataPhase flag is not set, then the Response packet parameter count indicates the number of pairs. If HasDataPhase flag is set, then the second parameter is the number of bytes in the data phase. "n" indicates number of memory region pairs
RAMStartAddress [<memoryID>]	No	0Eh	4	Start address of on-chip SRAM memoryID 0 (default) - Start address of SRAM partitions located in Common domain 1 - Start address of SRAM partitions located in Sense domain
RAMSizeInBytes [<memoryID>]	No	0Fh	4	On-chip SRAM size in bytes memoryID 0 (default) - Size of SRAM partitions located in Common domain 1 - Size of SRAM partitions located in Sense domain
SystemDeviceId	No	10h	4	Value of the DEVICD_ID and DIEID
LifeCycleState	No	11h	8	The life cycle of the device.

Table continues on the next page...

Table 138. Properties used by Get/SetProperty commands, sorted by values (continued)

Property	Writable	Tag value	Size (byte)	Description
				5aa5_5aa5h - Device is in development life cycle. c33c_c33ch - Device is in deployment life cycle.
UniqueDevice/UUID	No	12h	16	Unique device identification
Target ROM Version	No	18h	4	The target build version number
ExternalMemoryAttributes	No	19h	24	List of attributes supported by the specified memory Id (110h=SPI NOR FLASH).
IrqNotifierPin	Yes	1Ch	4	IRQ notifier pin setting bit[7:0] - GPIO pin, range: 0-31 bit[15:8] - GPIO port, range: 0-7 bit[16] - active polarity: 0 - active low, 1 - active high bit [30:17] - reserved bit [31] - enable flag: 0 - disable, 1 - enable **NOTE** You must not configure the IRQ notifier pin the same as that used as the boot pin, boot interface pin or ISP interface pin.

11.5.15.1 Property definitions

Get/Set property definitions are provided in this section.

11.5.15.1.1 CurrentVersion property

The value of this property is a 4-byte structure containing the current version of the bootloader.

Table 139. CurrentVersion property fields

Bit	[31:24]	[23:16]	[15:8]	[7:0]
Field	Name = 'K' (0x4B)	Major version	Major version	Bugfix version

11.5.15.1.2 AvailablePeripherals property

The value of this property is a bitfield that lists the peripherals supported by the bootloader and the hardware on which it is running.

Table 140. Peripheral bits

Bit	[31:7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Field	Reserved	Reserved	Reserved	USB HID	CAN	SPI target	I2C target	LPUART

If the peripheral is available, then the corresponding bit will be set in the property value. All reserved bits must be set to 0.

11.5.15.1.3 AvailableCommands property

This property value is a bitfield with set bits indicating the commands enabled in the bootloader. Only commands that can be sent from the host to the target are listed in the bitfield. Response commands such as GenericResponse are excluded.

The bit number that identifies whether a command is present is the command's tag value minus 1. 1 is subtracted from the command tag because the lowest command tag value is 0x01. To get the bit mask for a given command, use this expression: mask = 1 << (tag - 1).

Table 141. Command bits

Bit	Command
[Others]	Reserved
[21]	TrustProvisioning
[20:17]	Reserved
[16]	ConfigureMemory
[15]	Reserved
[14]	eFuseRead/FlashReadOnce
[13]	eFuseProgram/FlashProgramOnce
[12]	Reserved
[11]	SetProperty
[10]	Reset
[9]	Reserved
[8]	Execute
[7]	ReceiveSbFile
[6]	GetProperty
[5]	Reserved
[4]	FillMemory
[3]	WriteMemory
[2]	ReadMemory
[1]	FlashEraseRegion
[0]	FlashEraseAll

11.5.15.1.4 ExternalMemoryAttributes property

The value returned by this property is a 24-byte data structure containing available external memory attributes: start address, total size in KB, page size, sector size, and block size. Below is the breakdown of 24-byte structure.

Table 142. Fields of ExternalMemoryAttributes property

Field offset	Field Description
0 - 3	The value returned is a bitmap showing the supported attributes for the external memory, with the corresponding bitfield set. 0x00000001 - start address

Table continues on the next page...

Table 142. Fields of ExternalMemoryAttributes property (continued)

Field offset	Field Description
	0x00000002 - total size 0x00000004 - page size 0x00000008 - sector size 0x00000010 - block size
4 - 7	Start address of external memory.
8 - 11	Total size of external memory in kilobytes.
12 - 15	Page size of external memory in bytes.
16 - 19	Sector size of external memory in bytes.
20 - 23	Block size of external memory in bytes.

11.6 Bootloader Status Error Codes

This section describes the status error codes that the Bootloader returns to the host.

Table 143. Bootloader status error codes, sorted by value

Error Code	Value	Description
kStatus_Success	0	Operation succeeded without error.
kStatus_Fail	1	The operation failed with a generic error.
kStatus_ReadOnly	2	Requested value cannot be changed because it is read-only.
kStatus_OutOfRange	3	Requested value is out of range.
kStatus_InvalidArgument	4	The requested command's argument is undefined.
kStatus_Timeout	5	A timeout occurred.
kStatus_NoTransferInProgress	6	No send in progress
kStatus_UnknownCommand	10000	This command is not recognized.
kStatus_SecurityViolation	10001	Security violation happened when receiving disallowed commands.
kStatus_AbortDataPhase	10002	Sender requested data phase abort.
kStatus_Ping	10003	Ping command received from the host.
kStatus_NoResponse	10004	No Response from the host.

Table continues on the next page...

Table 143. Bootloader status error codes, sorted by value (continued)

Error Code	Value	Description
kStatus_NoResponseExpected	10005	Expected No Response from the host.
kStatus_CommandUnsupported	10006	Unsupported command was received.
kStatusRomLdrSectionOverrun	10100	Means reached the end of the sb file processing.
kStatusRomLdrSignature	10101	The signature or version are incorrect.
kStatusRomLdrSectionLength	10102	The bootOffset/ new section count is out of range.
kStatusRomLdrUnencryptedOnly	10103	The unencrypted image is disabled.
kStatusRomLdrEOFReached	10104	The end of the image file is reached.
kStatusRomLdrChecksum	10105	The checksum of a command tag block is invalid.
kStatusRomLdrCrc32Error	10106	The CRC-32 of the data for a load command is incorrect.
kStatusRomLdrUnknownCommand	10107	An unknown command was found in the SB file.
kStatusRomLdrIdNotFound	10108	There was no bootable section found in the SB file.
kStatusRomLdrDataUnderrun	10109	The SB state machine is waiting for more data.
kStatusRomLdrJumpReturned	10110	The function that was jumped to by the SB file has returned.
kStatusRomLdrCallFailed	10111	The call command in the SB file failed.
kStatusRomLdrKeyNotFound	10112	A matching key was not found in the SB file's key dictionary to unencrypt the section.
kStatusRomLdrSecureOnly	10113	The SB file sent is unencrypted, and security on the target is enabled.
kStatusRomLdrResetReturned	10114	The SB file reset operation has unexpectedly returned.
kStatusRomLdrRollbackBlocked	10115	An Image version rollback event detected.
kStatusRomLdrInvalidSectionMacCount	10116	Invalid Section MAC count detected in the SB file.
kStatusRomLdrUnexpectedCommand	10117	The command tag in the sb-file is unexpected.
kStatusRomLdrBadSBKEK	10118	Bad SBKEK detected.
kStatusRomLdrPendingJumpCommand	10119	Represents that the jump command is pending and the actual jump is implemented in sbloader_finalize().

Table continues on the next page...

Table 143. Bootloader status error codes, sorted by value (continued)

Error Code	Value	Description
kStatusMemoryRangeInvalid	10200	The requested address range does not match an entry, or the length extends past the matching entry's end address.
kStatusMemoryReadFailed	10201	Memory Read failed.
kStatusMemoryWriteFailed	10202	Memory Write failed.
kStatusMemoryCumulativeWrite	10203	Cumulative write occurred due to a write to an unerased flash memory region
kStatusMemoryNotConfigured	10205	Memory was not configured before access
kStatusMemoryAlignmentError	10206	Alignment error during memory access
kStatusMemoryVerifyFailed	10207	Verifying operation failed after erasing or programming flash memory
kStatusMemoryWriteProtected	10208	Memory to be written is protected
kStatusMemoryAddressError	10209	Memory address is invalid or wrong
kStatusMemoryUnsupportedCommand	10214	The memory command is not supported.
kStatus_UnknownProperty	10300	The requested property value is undefined.
kStatus_ReadOnlyProperty	10301	The requested property value cannot be written.
kStatus_InvalidPropertyValue	10302	The specified property value is invalid.
kStatus_AppCrcCheckPassed	10400	CRC check is valid and passed.
kStatus_AppCrcCheckFailed	10401	CRC check is valid but failed.
kStatus_AppCrcCheckInactive	10402	CRC check is inactive.
kStatus_AppCrcCheckInvalid	10403	CRC check is invalid because the BCA is invalid, or the CRC parameters are unset (all 0xFF bytes).
kStatus_AppCrcCheckOutOfRange	10404	CRC check is valid, but addresses are out of range.
kStatus_RomApiExecuteCompleted	0	ROM successfully process the whole sb file/ boot image.
kStatus_RomApiNeedMoreData	10801	ROM needs more data to continue processing the boot image.
kStatus_RomApiBufferSizeNotEnough	10802	The user buffer is not enough for use by Kboot during execution of the operation.
kStatus_RomApiInvalidBuffer	10803	The user buffer is not ok for sbloader or authentication.

Table continues on the next page...

Table 143. Bootloader status error codes, sorted by value (continued)

Error Code	Value	Description
kStatus_XSPI_SequenceExecutionTimeout	15300	The XSPI sequence execution timeout.
kStatus_XSPI_InvalidSequence	15301	The XSPI LUT sequence invalid.
kStatus_XSPI_DeviceTimeout	15302	The XSPI device timeout.
kStatus_xSpiNAND_ReadPageFail	20800	XSPI NAND read page failure
kStatus_xSpiNAND_ReadCacheFail	20801	XSPI NAND read cache failure
kStatus_xSpiNAND_EccCheckFail	20802	XSPI NAND ECC cehck failure
kStatus_xSpiNAND_PageLoadFail	20803	XSPI NAND page load failure
kStatus_xSpiNAND_PageExecuteFail	20804	XSPI NAND page execute failure
kStatus_xSpiNAND_EraseBlockFail	20805	XSPI NAND erase block failure
kStatus_xSpiNAND_FcbUpdateFail	20808	XSPI NAND FCB update failure
kStatus_xSpiNAND_DbbtUpdateFail	20809	XSPI NAND DBBT update failure
kStatus_xSpiNAND_WriteAlignmentError	20810	XSPI NAND write address is not aligned
kStatus_xSpiNAND_NotFound	20811	XSPI NAND is not found
kStatus_XSPINOR_ProgramFail	20900	Page programming failure.
kStatus_XSPINOR_EraseSectorFail	20901	Sector Erase failure.
kStatus_XSPINOR_EraseAllFail	20902	Chip Erase failure.
kStatus_XSPINOR_WaitTimeout	20903	The execution time out.
kStatus_XSPINOR_NotSupported	20904	Page size overflow.
kStatus_XSPINOR_WriteAlignmentError	20905	Address alignment error.
kStatus_XSPINOR_CommandFailure	20906	Erase/Program verify error.
kStatus_XSPINOR_SFDP_NotFound	20907	Timeout occurs during the API. call
kStatus_XSPINOR_Unsupported_SFDP_Version	20908	Unrecognized SFDP version.
kStatus_XSPINOR_FLASH_NotFound	20909	Flash detection failure.
kStatus_XSPINOR_DTRRead_DummyProbeFailed	20910	DDR Read dummy probe failure.
kStatus_IAP_Success	0	IAP API execution succeeded.
kStatus_IAP_Fail	1	IAP API execution failed.
kStatus_IAP_InvalidArgument	100001	Invalid argument detected during API execution.

Table continues on the next page...

Table 143. Bootloader status error codes, sorted by value (continued)

Error Code	Value	Description
kStatus_IAP_OutOfMemory	100002	The Heap size is not enough during API execution.
kStatus_IAP_ReadDisallowed	100003	The read memory operation is disallowed during API execution.
kStatus_IAP_CumulativeWrite	100004	The FLASH region to be programmed is not empty.
kStatus_IAP_EraseFailuire	100005	Erase operation failed.
kStatus_IAP_CommandNotSupported	100006	The specific command is not supported.
kStatus_IAP_MemoryAccessDisabled	100007	Memory access is disabled, typically occurred on the XSPI NOR if it is not configured properly.
kStatus_NBOOT_Success	0x5A5A5A5Au	Operation completed successfully in NBOOT functions.
kStatus_NBOOT_Fail	0x5A5AA5A5u	Operation failed in NBOOT functions.
kStatus_NBOOT_InvalidArgument	0x5A5AA5F0u	Invalid argument passed to the function in NBOOT functions.
kStatus_NBOOT_RequestTimeout	0x5A5AA5E1u	Operation timed out in NBOOT functions.
kStatus_NBOOT_KeyNotLoaded	0x5A5AA5E2u	The requested key is not loaded in NBOOT functions.
kStatus_NBOOT_AuthFail	0x5A5AA5E4u	Authentication failed in NBOOT functions.
kStatus_NBOOT_OperationNotAvaialable	0x5A5AA5E5u	Operation not available on this HW in NBOOT functions.
kStatus_NBOOT_KeyNotAvailable	0x5A5AA5E6u	Key is not available in NBOOT functions.
kStatus_NBOOT_IvCounterOverflow	0x5A5AA5E7u	Overflow of IV counter (PRINCE/IPED) in NBOOT functions.
kStatus_NBOOT_SelftestFail	0x5A5AA5E8u	FIPS self-test failure in NBOOT functions.
kStatus_NBOOT_InvalidDataFormat	0x5A5AA5E9u	Invalid data format for example antipole in NBOOT functions.
kStatus_NBOOT_IskCertUserDataTooBig	0x5A5AA5EAu	Size of User data in ISK certificate is greater than 96 bytes in NBOOT functions.
kStatus_NBOOT_IskCertSignatureOffsetTooSmall	0x5A5AA5EBu	Signature offset in ISK certificate is smaller than expected in NBOOT functions.
kStatus_NBOOT_MemcpyFail	0x5A5A845Au	Unexpected error detected during nboot_memcpy()in NBOOT functions.

Table continues on the next page...

Table 143. Bootloader status error codes, sorted by value (continued)

Error Code	Value	Description
kStatus_SKBOOT_Success	0x5ac3c35au	Operation completed successfully in SKBOOT functions.
kStatus_SKBOOT_Fail	0xc35ac35au	Operation failed in SKBOOT functions.
kStatus_SKBOOT_InvalidArgument	0xc35a5ac3u	Return Invalid argument status in SKBOOT functions.
kStatus_MKBOOT_Success	0x0000c35au	Operation completed successfully in Prime KBOOT.
kStatus_MKBOOT_Fail	0x0000c3c3u	Operation failed in Prime KBOOT.
kStatus_MKBOOT_InvalidArgument	0x00005ac3u	Return Invalid argument status in Prime KBOOT.

NOTE

In UART, I2C, CAN and SPI ISP modes, the ROM expects posted responses to be read by the host within 20ms*number of bytes, otherwise the ROM reports the abort command. Considering an ACK from the ROM is only two bytes, the host must read those bytes within 40ms of the ROM posting.

11.7 UART ISP

11.7.1 Introduction

The bootloader integrates an autobaud detection algorithm for the UART peripheral, thereby providing flexible baud rate choices.

Autobaud feature: If UART*n* is used to connect to the bootloader, then the UART*n*_RX pin must be kept high and not left floating during the detection phase in order to comply with the autobaud detection algorithm. After the bootloader detects the Ping packet (0x5A 0xA6) on UART*n*_RX, the bootloader firmware executes the autobaud sequence.

If the baudrate is successfully detected, then the bootloader sends a Ping packet response [(0x5A 0xA7), protocol version (4 bytes), protocol version options (2 bytes) and crc16 (2 bytes)] at the detected baudrate. The bootloader then enters a loop, waiting for bootloader commands via the UART peripheral.

NOTE: The data bytes of the ping packet must be sent continuously (with no more than 80 ms between bytes) in a fixed UART transmission mode (8-bit data, no parity bit and 1 stop bit). If the bytes of the ping packet are sent one-by-one with more than 80 ms delay between them, then the autobaud detection algorithm may calculate an incorrect baud rate. In this instance, the autobaud detection state machine should be reset.

Supported baud rates: The baud rate is closely related to the MCU core and system clock frequencies. Typical baud rates supported are 9600, 19200, 38400, 57600, 115200 ,230400, 460800 and 1000000.

Packet transfer: After autobaud detection succeeds, bootloader communications can take place over the UART peripheral. The following flow charts show:

- How the host detects an ACK from the target.
- How the host detects a ping response from the target.
- How the host detects a command response from the target.

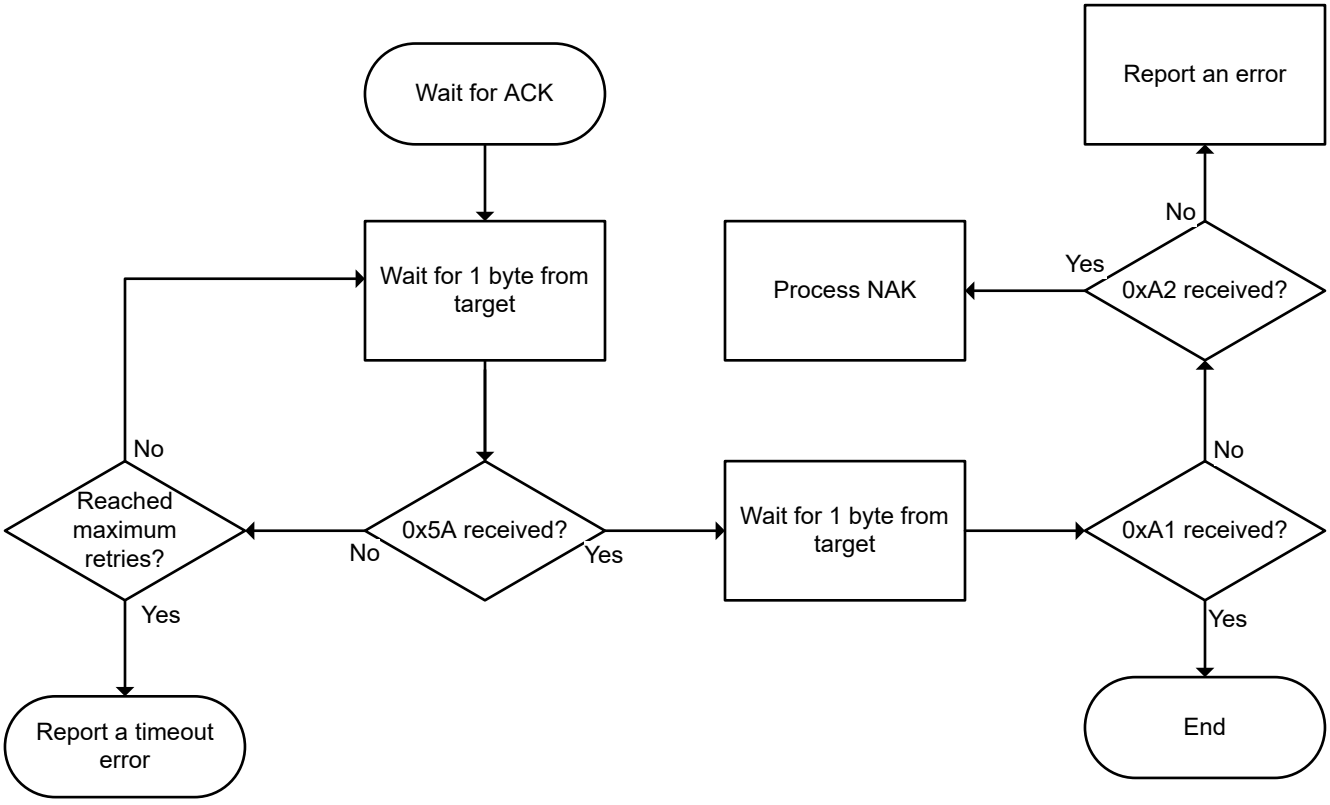


Figure 33. Host reads an ACK from target via UART

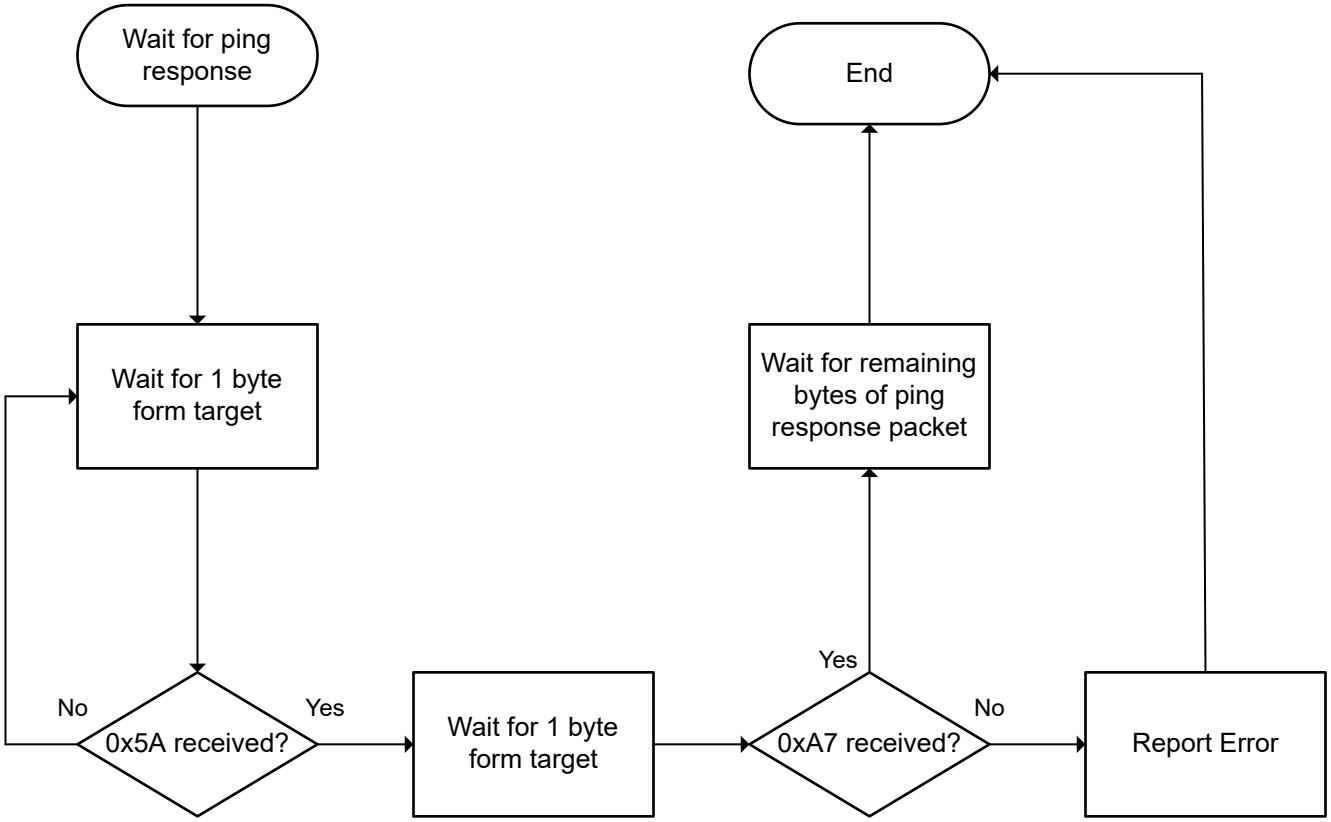


Figure 34. Host reads a ping response from target via UART

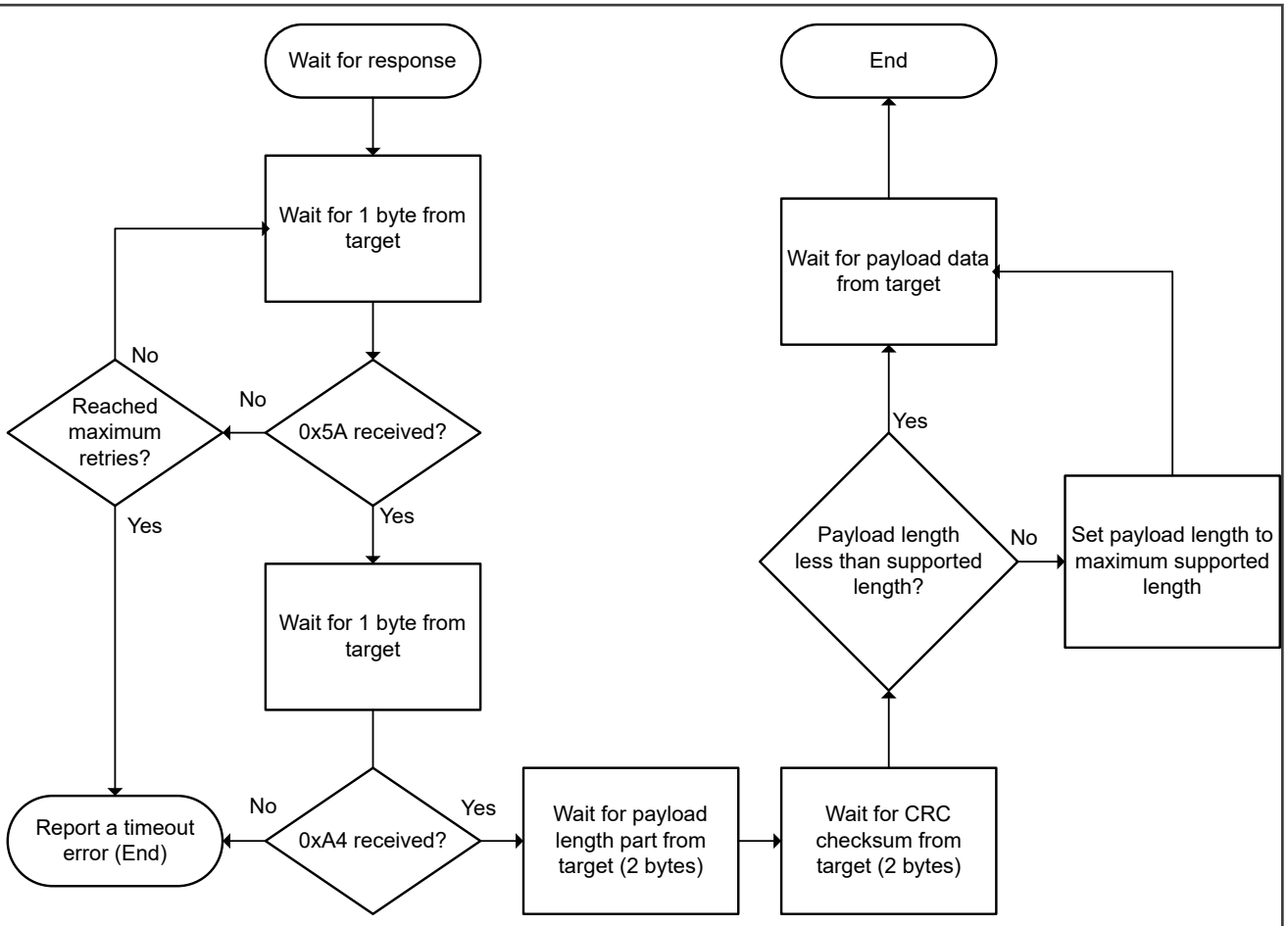


Figure 35. Host reads a command response from target via UART

See Bootloader packet types for more details on UART ISP command format, UART ISP response format, UART ISP data format, and UART ISP commands.

11.8 I2C In-System Programming

11.8.1 Introduction

The bootloader supports loading data into flash via the I2C peripheral, where the I2C peripheral serves as the I2C target. A 7-bit target address is used during the transfer. The bootloader uses 0x10 as the I2C target address and supports up to 400 kbit/s as the I2C baud rate.

The maximum supported I2C baud rate depends on the core clock frequency when the bootloader is running. The typical baud rate is 400 kbit/s with factory settings.

Because the I2C peripheral serves as an I2C target device, each transfer should be started by the host, and each outgoing packet should be fetched by the host.

- An incoming packet is sent by the host with a selected I2C target address and the direction bit is set to write.
- An outgoing packet is read by the host with a selected I2C target address and the direction bit is set as read.
- 0x00 is sent as the response to host if the target is busy with processing or preparing data.

The following charts show the communication flow of the host reading the ping and ACK packets, and the corresponding responses from the target.

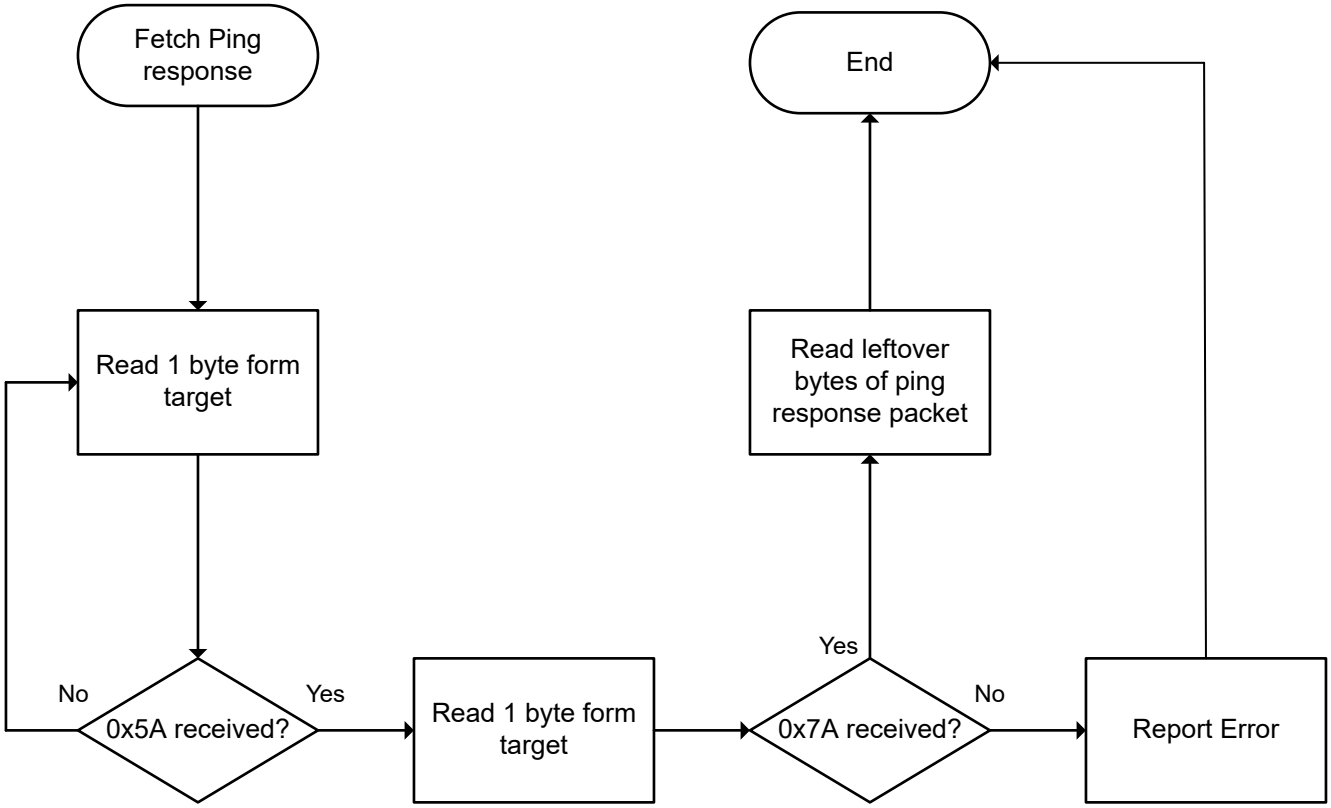


Figure 36. Host reads ping response from target via I2C

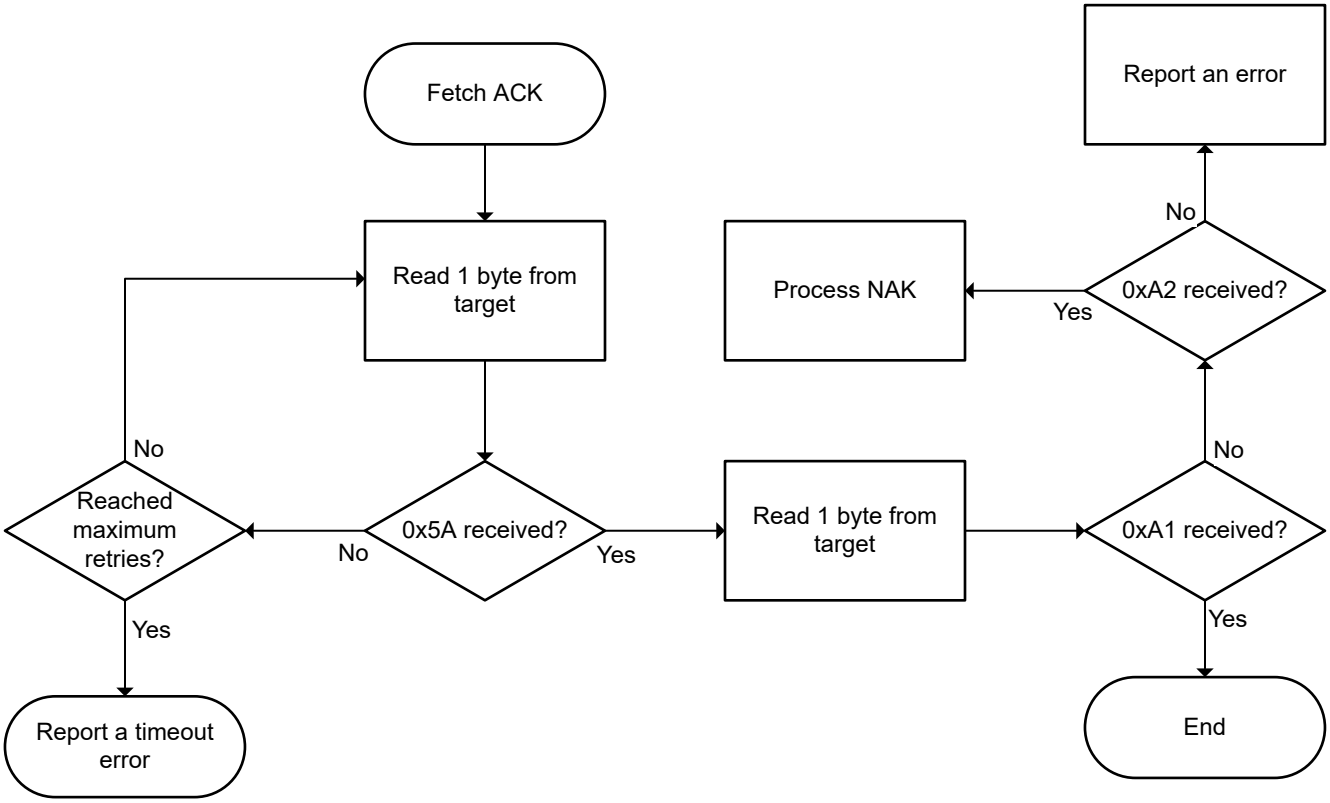


Figure 37. Host reads ACK packet from target via I2C

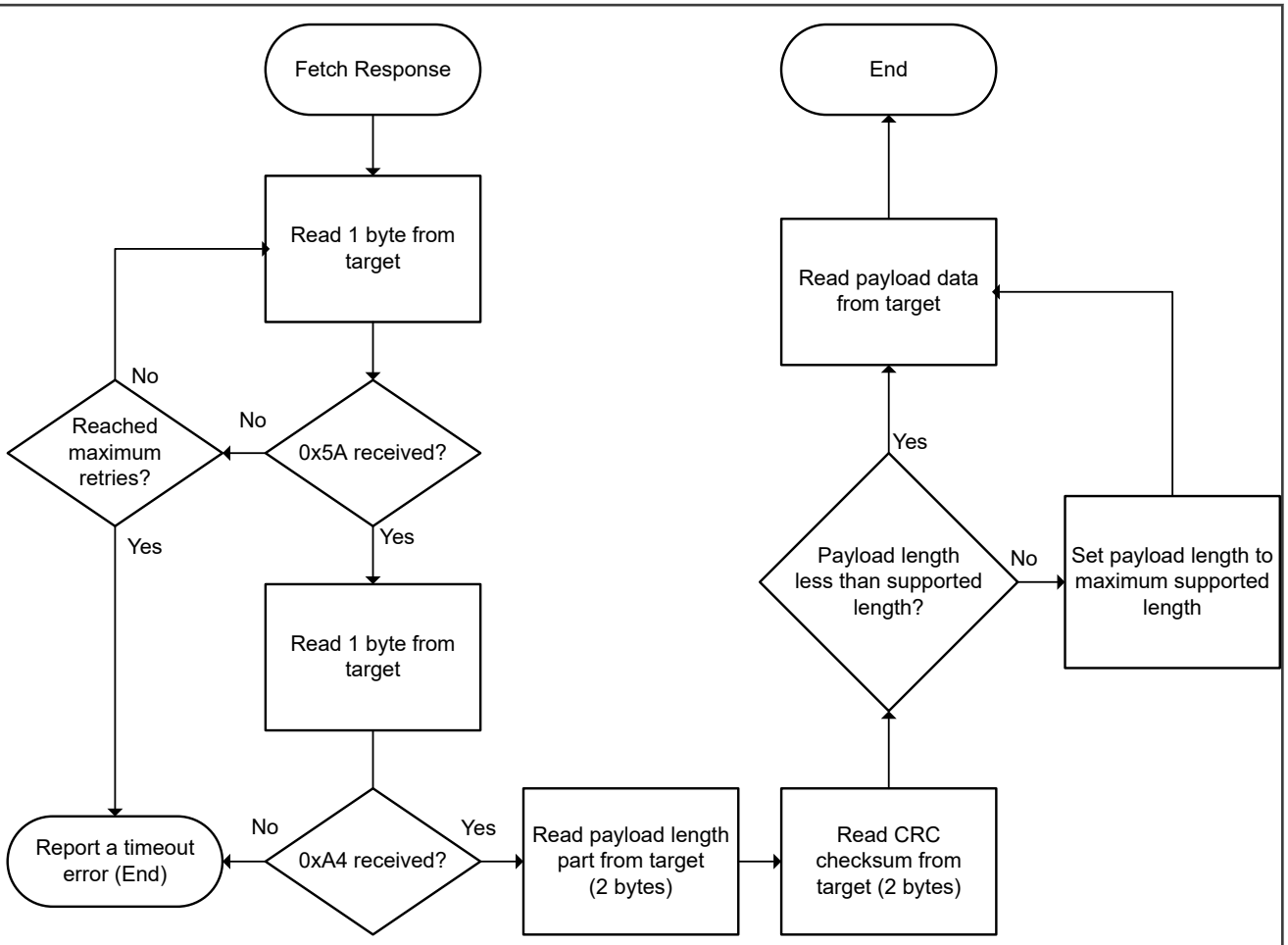


Figure 38. Host reads response from target via I2C

See Bootloader packet types for more details on I2C ISP command format, I2C ISP response format, and I2C ISP data format.

See Bootloader command set for more details on I2C ISP commands.

The bootloader also supports the active notification pin (nIRQ pin) to notify the host processor it is busy or ready for new commands/data.

To accelerate the I2C transfer between the host and the bootloader, the bootloader provides an active notification pin known as the nIRQ pin, it can be enabled by the SetProperty command. In SetProperty command, an active polarity field in defined.

- If the active polarity is set as default 0, then the host needs to wait until it sees a negative edge on the nIRQ pin before reading any data from the bootloader, and it needs to wait until the nIRQ pin is high before sending any data to the bootloader.
- If the active polarity is set as 1, then the host needs to wait until it sees a positive edge on the nIRQ pin before reading any data from the bootloader, and it needs to wait until the nIRQ pin is low before sending any data to the bootloader.

See Table 138 for more details.

11.9 SPI In-System programming

11.9.1 Introduction

The bootloader supports loading data into flash via the SPI peripheral, where the SPI peripheral serves as an SPI target. The SPI transfer should be SPI Mode 3 with 8 data bits.

The maximum supported baud rate of the SPI depends on the core clock frequency when the bootloader is running. The typical baud rate is 2000 kbit/s with the factory settings. The actual baud rate is lower or higher than 2000 kbit/s, depending on the actual value of the core clock.

Because the SPI peripheral serves as an SPI target device, each transfer should be started by the host, and each outgoing packet should be fetched by the host.

- The transfer on SPI is slightly different from I^2C:
- The host receives 1 byte after it sends out any byte.
- Received bytes should be ignored when the host is sending out bytes to the target
- The host starts reading bytes by sending 0x00s to target

The byte 0x00 is sent as a response to host if the target is under the following conditions:

- Processing incoming packet.
- Preparing outgoing data.
- Received invalid data.

The bootloader also supports the active notification pin (nIRQ pin) to notify the host processor it is busy or ready for new commands/data.

To accelerate the SPI transfer between the host and the bootloader, the bootloader provides an active notification pin known as the nIRQ pin, it can be enabled by the SetProperty command. In SetProperty command, an active polarity field in defined.

- If the active polarity is set as default 0, then the host needs to wait until it sees a negative edge on the nIRQ pin before reading any data from the bootloader, and it needs to wait until the nIRQ pin is high before sending any data to the bootloader.
- If the active polarity is set as 1, then the host needs to wait until it sees a positive edge on the nIRQ pin before reading any data from the bootloader, and it needs to wait until the nIRQ pin is low before sending any data to the bootloader.

See Table 138 for more details.

See below figure for the typical physical connection between the host and the bootloader device.

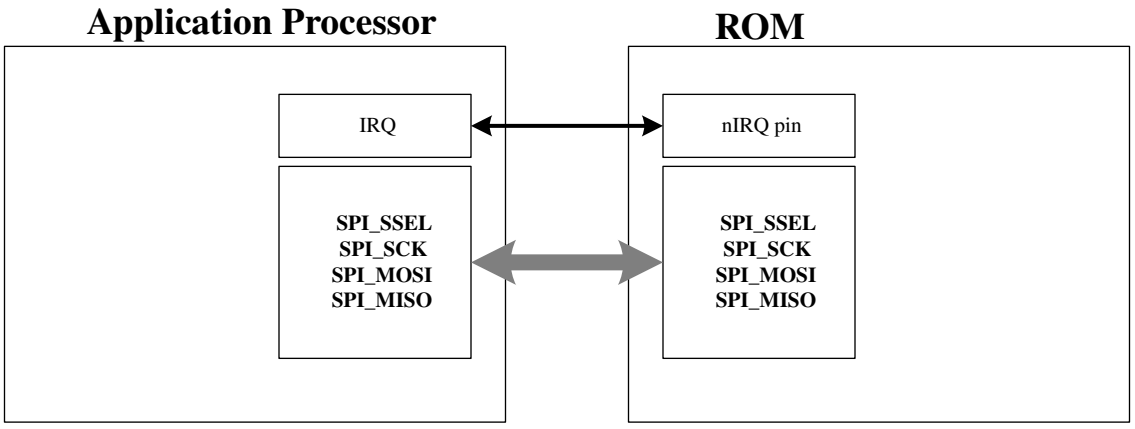


Figure 39. Physical interface for SPI ISP

The following flowcharts show how the host reads a ping response, an ACK and a command response from target via SPI without the nIRQ pin enabled.

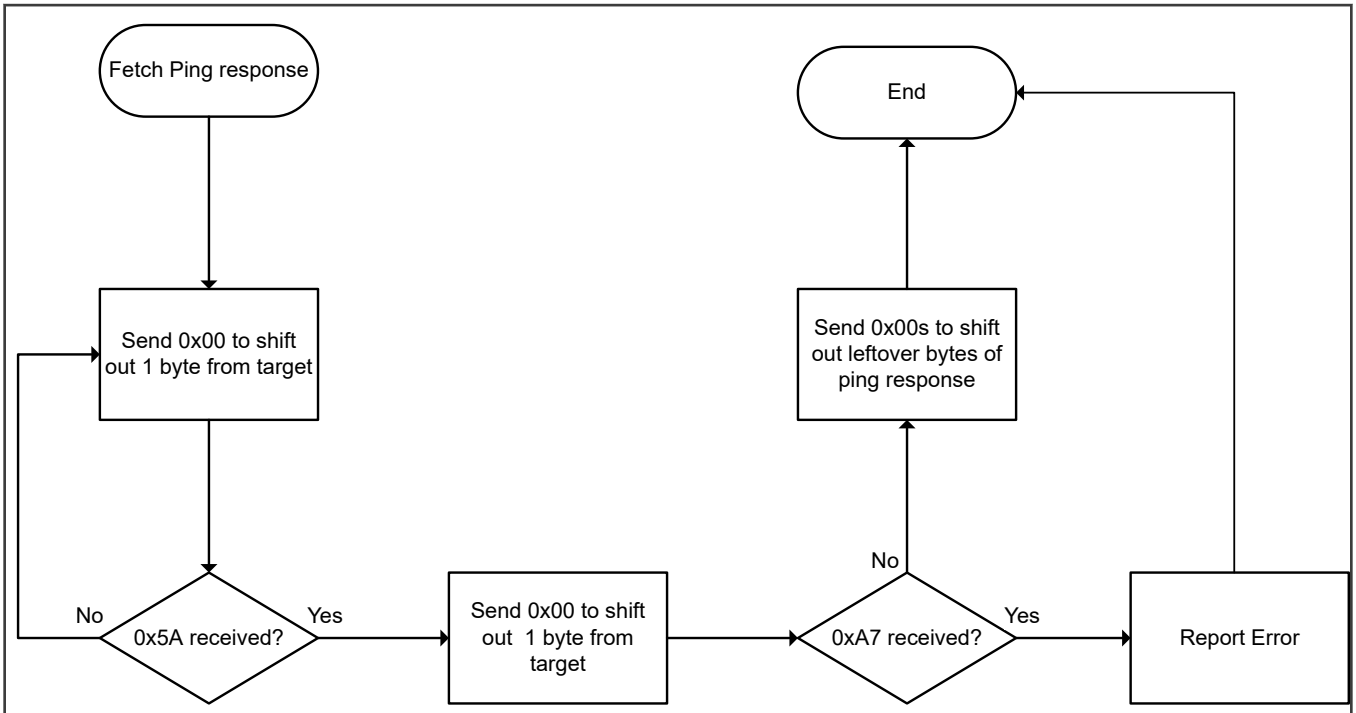


Figure 40. Host reads ping packet from target via SPI

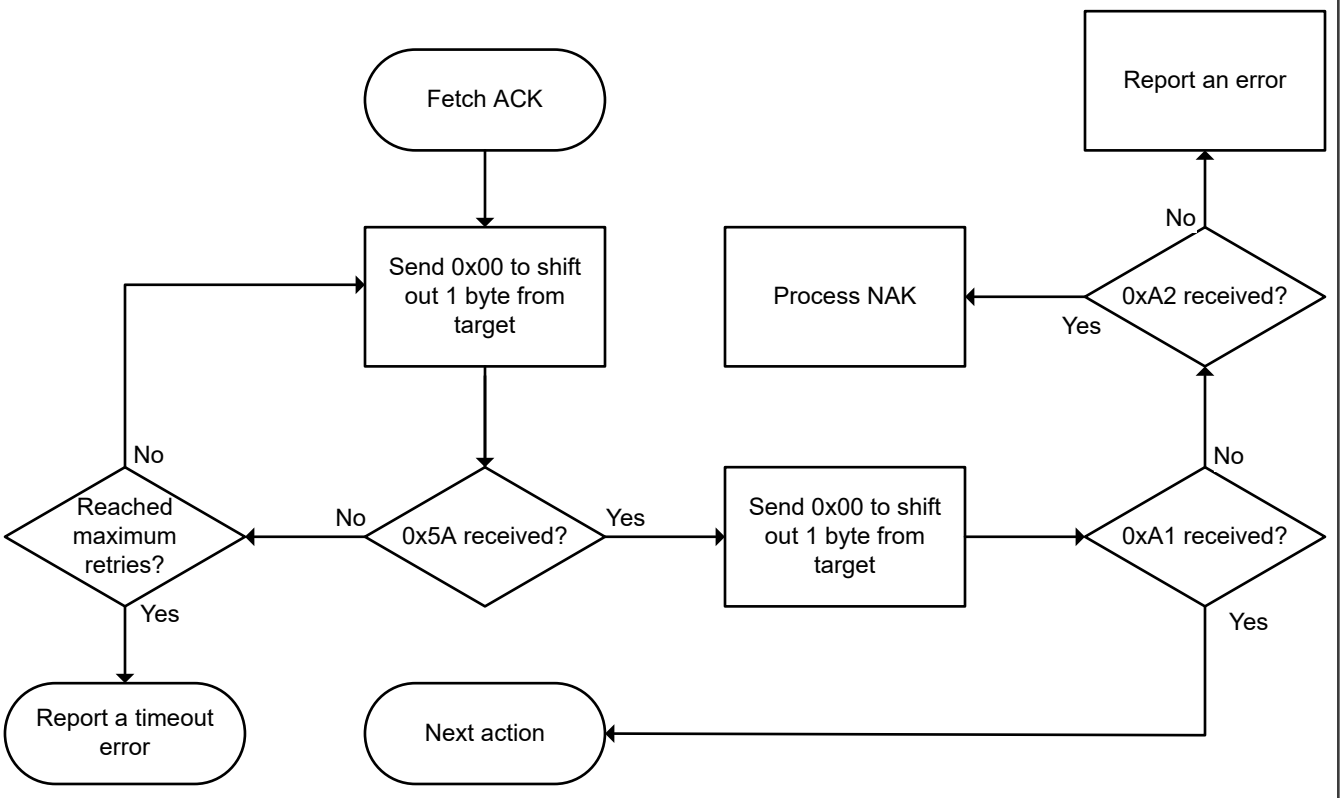


Figure 41. Host reads ACK from target via SPI

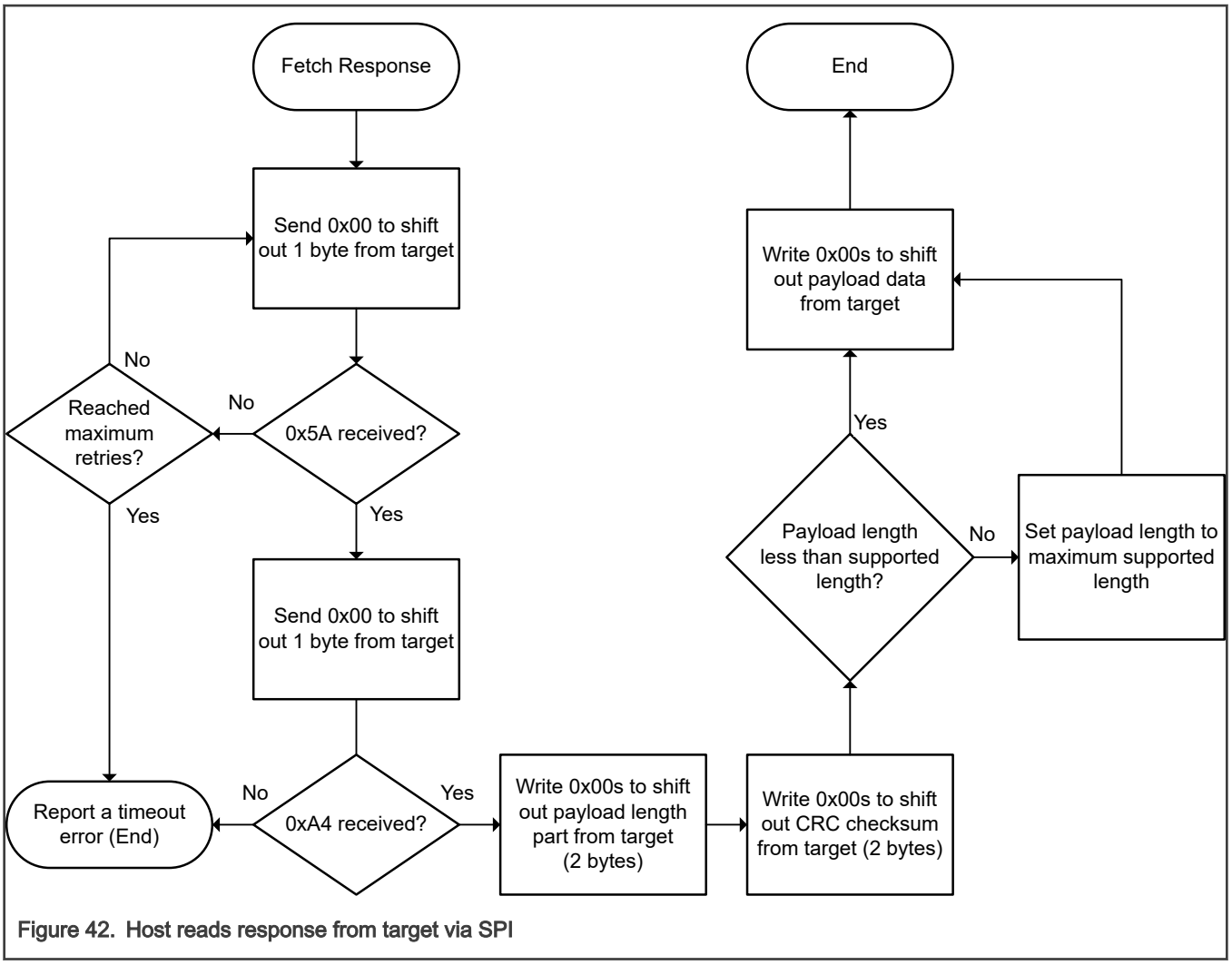


Figure 42. Host reads response from target via SPI

See Bootloader packet types for more details on SPI ISP command format, SPI ISP response format, and SPI ISP data format.

See Bootloader command set for more details on SPI ISP commands.

11.9.2 Host read ACK(5a,a1) packet timing restriction

If IRQ pin is not used for the In-System Programming protocol, the host needs to follow the below data fetching timing for ACK from the device as the below figures shows.

- After sending out the command, the host should wait for at least 100 us before polling the start byte 0x5A of the ACK packet sent out by the device; If fetching data is not 0x5A, delay 100us, and do polling to reread 0x5A.
- For the ACK packet 0x5A, 0xA1, after the host receives the 0x5A, a 50us delay must add before fetching the 0xA1 from the device.

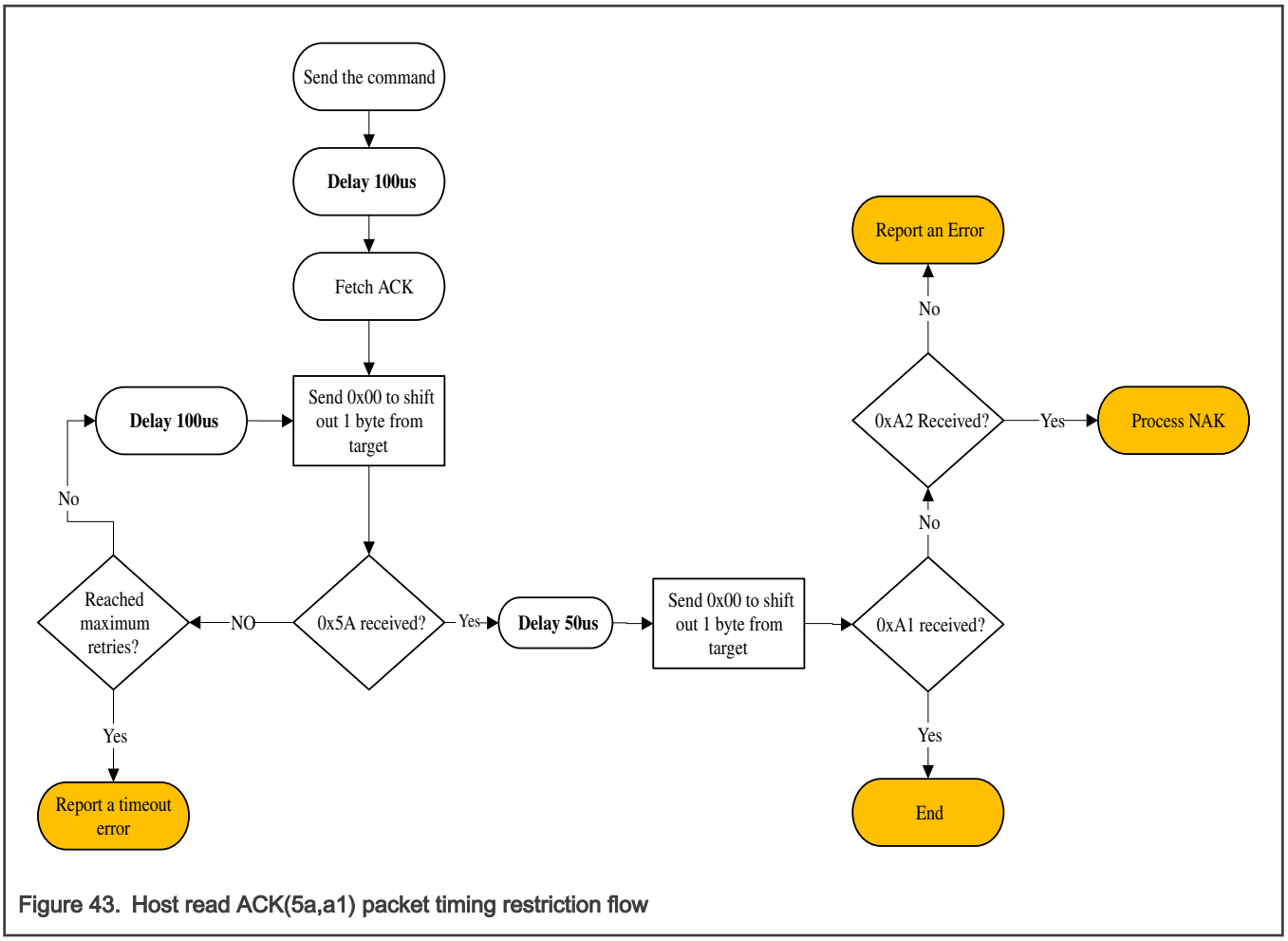


Figure 43. Host read ACK(5a,a1) packet timing restriction flow

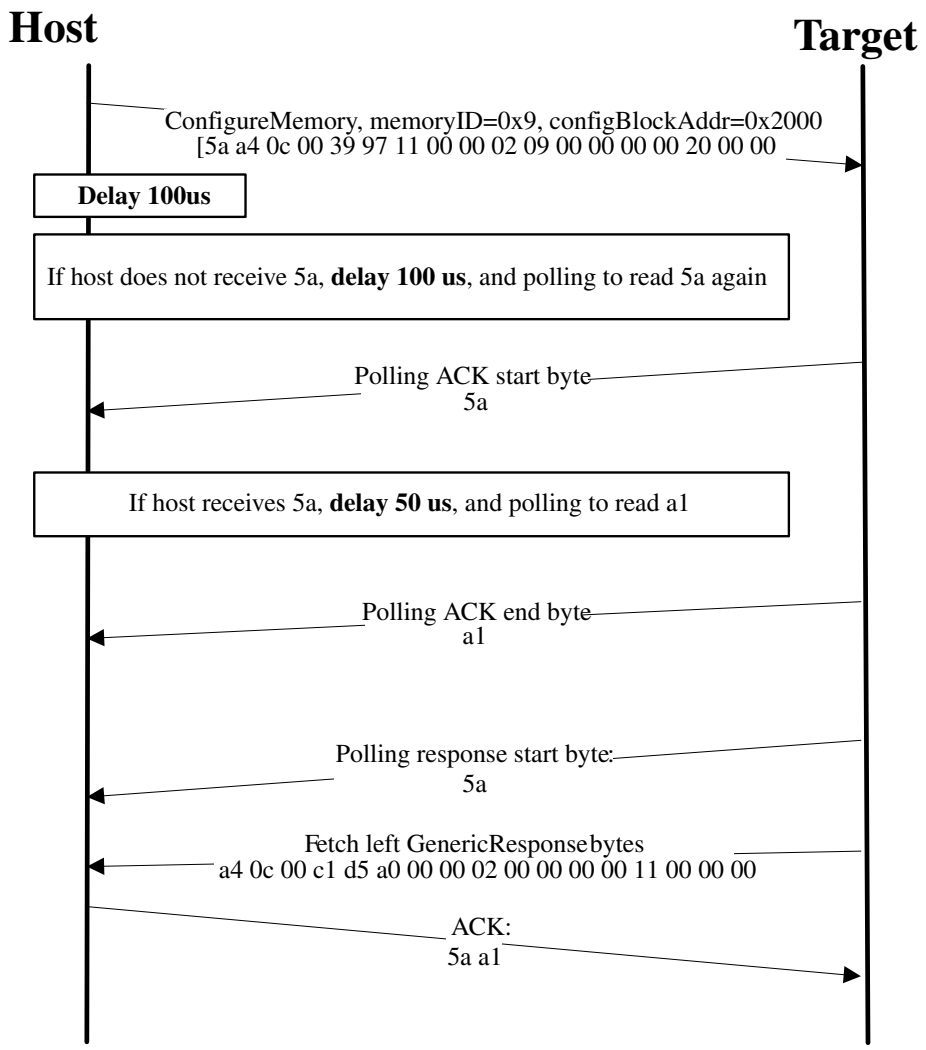


Figure 44. Host read ACK(5a,a1) packet timing restriction

11.10 USB In-System Programming

11.10.1 Introduction

The bootloader supports In-System Programming using the USB (USB0 and USB1) peripheral. The target is implemented as USB-HID device classes.

When transfer data through USB device class, USB does not use framing packets. Instead, the packetization, inherent in the USB protocol itself is used. The ability for the device to NAK out transfers (until they can be received) provides the required flow control. The built-in CRC of each USB packet provides the required error detection.

NOTE

USB interface needs an external XTAL supply, and thus you need to configure XTAL_FREQ within fuse BOOT_CFG5 correctly to match the actual frequency of XTAL for the use case. ROM assumes that a 24 MHz XTAL is used by default if this fuse region is not configured. Also, you need to configure OSC_HIGH_GAIN_MODE fuse bit according to the actual working mode of external XTAL. ROM assumes that you use XTAL in low power mode by default.

11.10.1.1 Device descriptor

The bootloader configures the default USB(USB0 and USB1) VID/PID/Strings as below:

Default VID/PID:

- VID = 0x1FC9
- PID = 0x0026

Default Strings:

- Manufacturer [1] = "NXP SEMICONDUCTOR INC".
- Product [2] = "USB HID DEVICE".

The USB (USB0 and USB1) VID, PID can be customized using the fuse BOOT_CFG6. For example, the USB VID and PID can be customized by writing the new VID to the USB_VID[15:0] in the fuse and the new PID to the USB_PID[15:0] in the fuse.

11.10.1.2 Endpoints

The HID peripheral uses three endpoints:

- Control (0).
- Interrupt IN (1).
- Interrupt OUT (2).

The Interrupt OUT endpoint is optional for HID class devices, but the MCU bootloader uses it as a pipe, where the firmware can NAK send requests from the USB host.

11.10.1.3 HID Reports

There are four HID reports defined and used by the bootloader USB HID peripheral. The report ID determines the direction and type of packet sent in the report; otherwise, the contents of all reports are the same.

Table 144. HID reports assigned for the bootloader

Report ID	Packet Type	Direction
1	Command	OUT
2	Data	OUT
3	Command	IN
4	Data	IN

Each report has a maximum size of 60 bytes. The maximum payload size is 56 bytes. In addition, there is a 4-byte report header that indicates the length (in bytes) of the payload and report id sent the packet.

The actual data sent in all of the reports looks like:

Table 145. Data format sent in USB HID packet

Report ID	Data
0	Report ID
1	Padding

Table continues on the next page...

Table 145. Data format sent in USB HID packet (continued)

2	Packet Length LSB
3	Packet Length MSB
4	Packet[0]
5	Packet[1]
6	Packet[2]
	...
N+4-1	Packet[N-1]

This data includes the Report ID, which is required if more than one report is defined in the HID report descriptor. The actual data sent and received has a maximum length of 35 bytes. The Packet Length header is written in little-endian format, and it is set to the size (in bytes) of the packet sent in the report. This size does not include the Report ID or the Packet Length header itself. During a data phase, a packet size of 0 indicates a data phase abort request from the receiver.

See Command packet for more details on USB ISP command format, and USB ISP response format.

See Response packet for more details on USB ISP data format.

See Bootloader command set for more details on USB ISP commands.

Chapter 12
Power Overview

12.1 Introduction

Power Management Controller (PMC) and Sleep Controller (SLEEPCON) handle the distribution, control and management of power throughout the chip.

- PMC controls power mode transitions, power domain power switching (on or off), and the main and peripheral memory power switching.
- SLEEPCON controls main clock gating per domain in low power modes and clock sources in power down modes.

12.2 Power architecture

The chip has nine separate power domains and multiple distinct power rails. At any given time, domains can be active or placed in a lower-power state to meet the performance requirements of a system application. This power architecture allows for a more granular distribution of power where and when needed.

The chip includes separate Compute and Sense processing domains that can independently enter or exit Sleep and Deep Sleep modes. In addition, separate Media and Common domains on power rail VDDN contain modules that need a higher voltage to operate, while other domains can operate at a lower voltage.

12.2.1 Power domains

A power domain is a collection of circuits that can be powered off even when a voltage source is still supplied. The table below summarizes the power domains within the chip, and the following figure shows the placement of modules within these domains.

Table 146. Power domain summary

Power domain	Modules
VDD2_COMP	CPU0, NPU, and Compute Domain modules
VDD2_DSP	HiFi4 DSP
VDD2_MEDIA	Media Domain modules on the VDD2 power rail
VDD2_COM	Common Domain modules on the VDD2 power rail
VDD1_SENSE	CPU1, Sense Domain (private region) modules, and Common Domain modules
VDDN_MEDIA	Media Domain modules on the VDDN power rail
VDDN_COM	Common Domain modules on the VDDN power rail
VDD1V8_AON	Always-on modules, also includes RESETN, PMIC_MODE0, PMIC_MODE1, PMIC_IRQN, XTALIN32KHZ, and XTALOUT32KHZ pins
VDD1V8_PMC	PMC

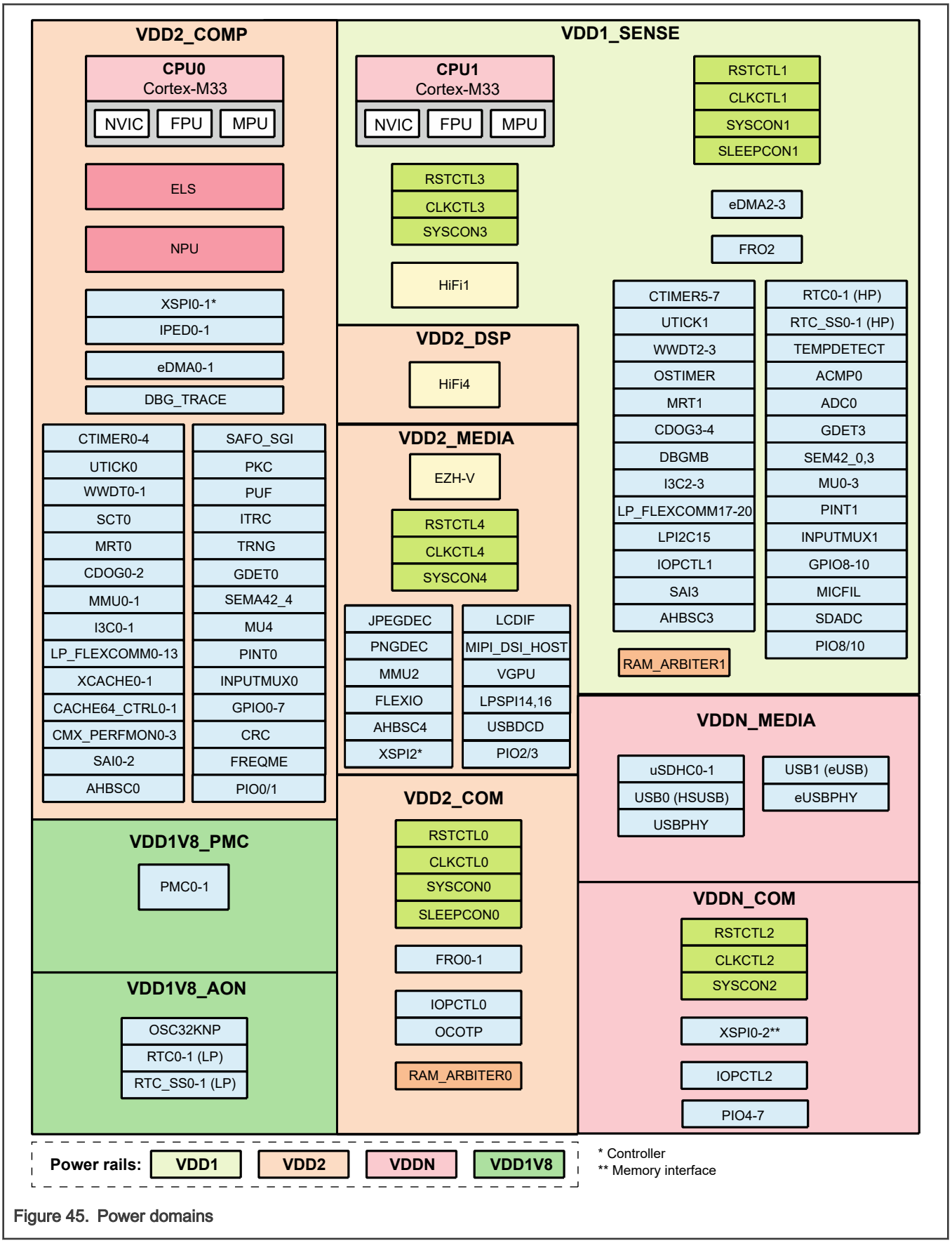


Figure 45. Power domains

Below is a complete list of modules within their respective power domains.

Table 147. Power domains

VDD1_SENSE	VDD2_COMP	VDD2_MEDIA	VDDN_MEDIA
CPU1	CPU0	EZH-V	uSDHC0-1
RSTCTL1,3	ELS	RSTCTL4	USB0-1
CLKCTL1,3	NPU	CLKCTL4	USBPHY
SYSCON1,3	XSPI0-1 (Controller)	SYSCON4	eUSBPHY
SLEEPCON1	IPED0-1	JPEGDEC	**VDDN_COM**
HiFi1	eDMA0-1	PNGDEC	RSTCTL2
eDMA2-3	DBG_TRACE	MMU2	CLKCTL2
FRO2	CTIMER0-4	FLEXIO	SYSCON2
CTIMER5-7	UTICK0	AHBSC4	XSPI0-2 (Memory interface)
UTICK1	WWDT0-1	XSPI2 (Controller)	IOPCTL2 (GPIO4-7)
WWDT2-3	SCT0	LCDIF	PIO4-7
OSTIMER	MRT0	MIPI_DSI_HOST	
MRT1	CDOG0-2	VGPU	
CDOG3-4	MMU0-1	LPSPI14, 16	
DBGMB	I3C0-1	USBDCD	
I3C2-3	LP_FLEXCOMM0-13	**VDD2_COM**	**VDD1V8_AON**
LP_FLEXCOMM17-20	XCACHE0-1	RSTCTL0	OSC32KNP
LPI2C15	CACHE64_CTRL0-1	CLKCTL0	RTC0-1 (LP)
IOPCTL1 (GPIO8-10)	CMX_PERFMON0-3	SYSCON0	RTC_SS0-1 (LP)
SAI3	SAI0-2	SLEEPCON0	
AHBSC3	AHBSC0	FRO0-1	
RTC0-1 (HP)	SAFO_SGI	IOPCTL0 (GPIO0-3)	
RTC_SS0-1 (HP)	PKC	OCOTP	
TEMPDETECT	PUF	RAM_ARBITER0	
ACMP0	ITRC	PIO2/3	
ADC0	TRNG	**VDD2_DSP**	**VDD1V8_PMC**
GDET3	GDET0	HiFi4	PMC0-1
SEMA42_0,3	SEMA42_4		
MU0-3	MU4		
PINT1	PINT0		
INPUTMUX1	INPUTMUX0		
GPIO8-10	GPIO0-7		
MICFIL	CRC		

Table continues on the next page...

Table 147. Power domains (continued)

SDADC RAM_ARBITER1 PIO8/10	FREQME PIO0/1		

12.2.2 Power supplies

The chip has separate power supplies for the digital logic: VDD1, VDD2, VDDN, VDD1V8, and VDD1V8_AO. In addition, independent power supplies are available for the IO port pins.

Table 148. Power supply summary

Power supplies	Description
VDD1	For VDD1_SENSE domain
VDD2	For VDD2_COMP, VDD2_DSP, VDD2_MEDIA, and VDD2_COM domains
VDDN	For high-speed logic in VDDN_MEDIA domain, analog interfaces, and VDDN_COM domain
VDD1V8	For 1.8 V logic
VDD1V8_AO	For Always-on logic, such as power management

12.2.2.1 Main power from on-chip PMC

The on-chip PMC can supply power to the main power rails, as shown below.

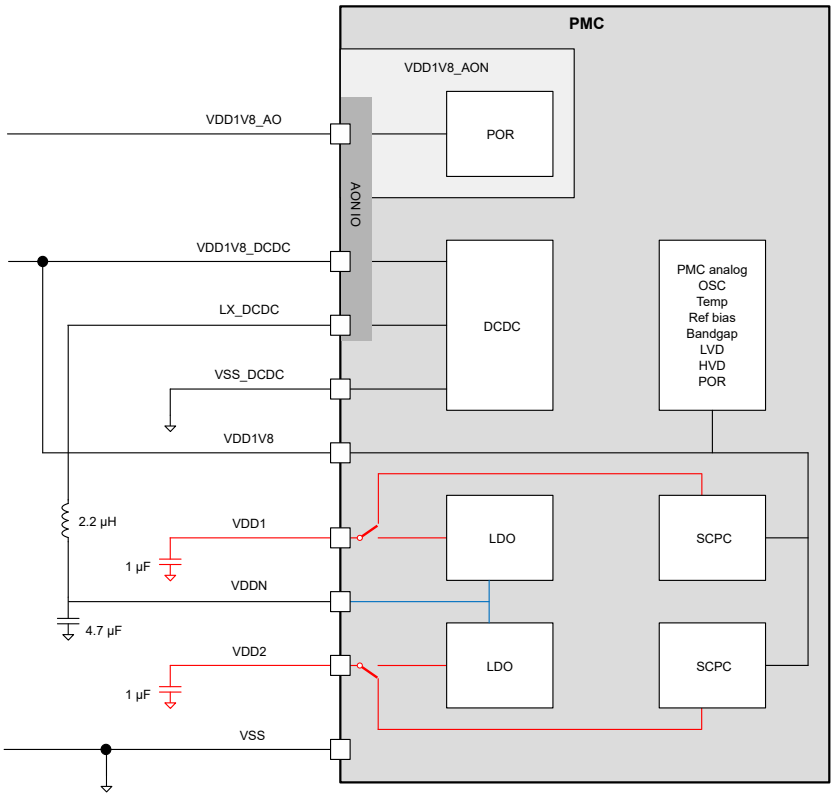


Figure 46. Main power connections from on-chip PMC

12.2.2.2 Main power from external PMIC

An optional external PMIC can supply power to the main power rails, as shown below.

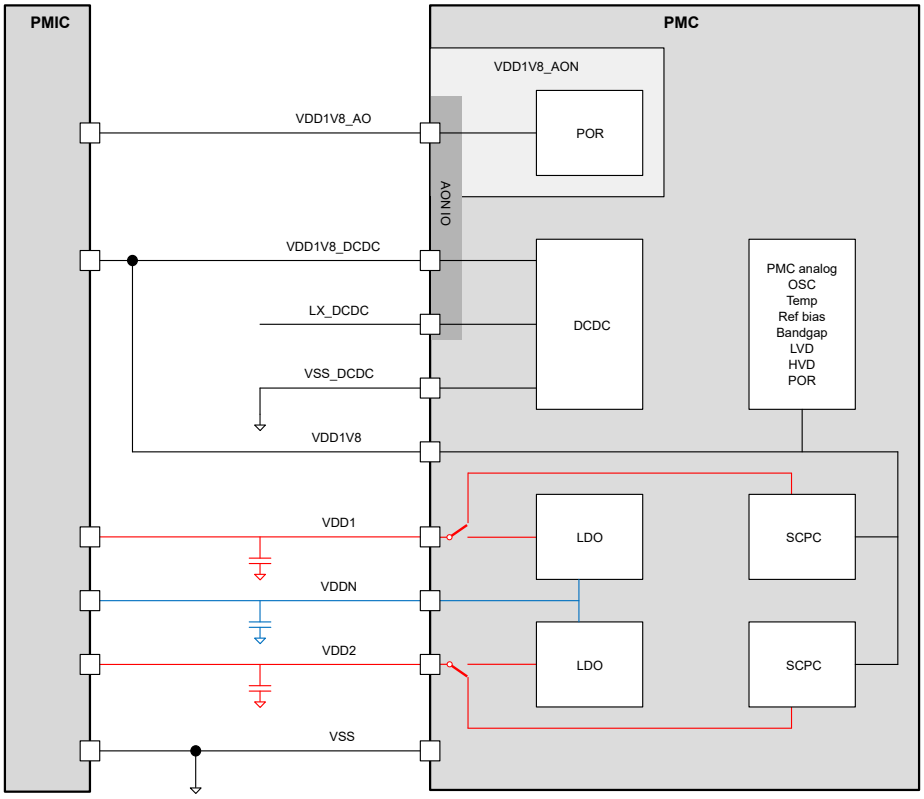


Figure 47. Main power connections from external PMIC

12.2.3 Power switches

Below shows the power switch architecture for each of the domains. The VDD1V8_AON and VDD1V8_PMC domains are not included because they are not switchable.

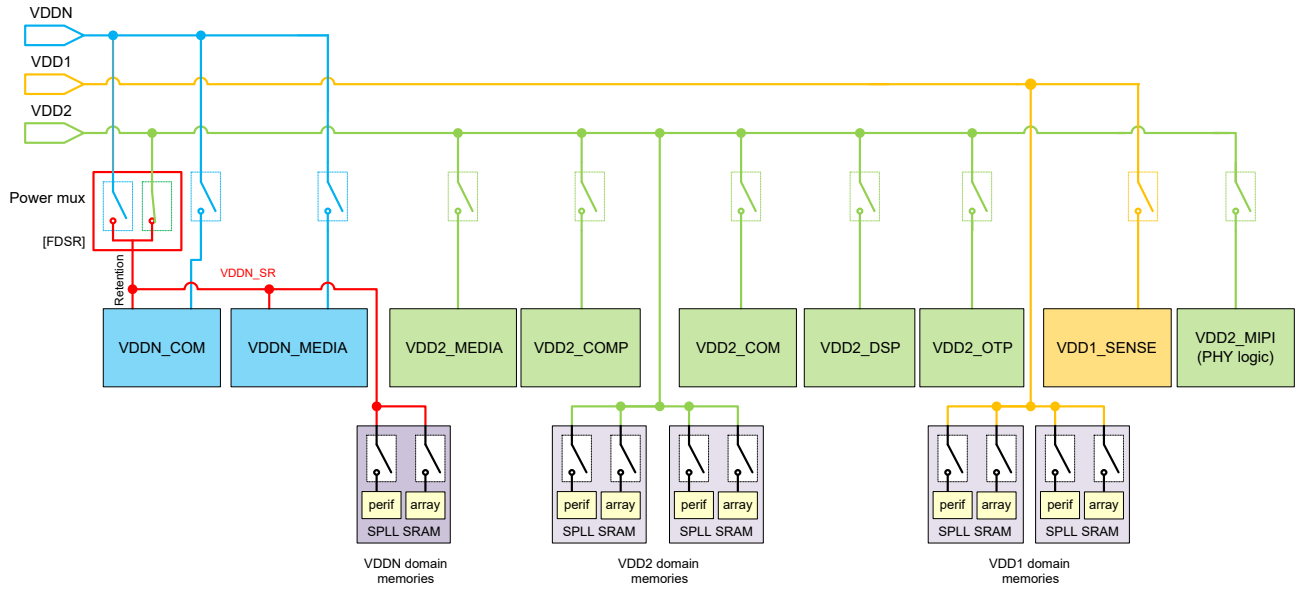


Figure 48. Power switch architecture

The VDDN domains (VDDN_COM and VDDN_MEDIA) have a power mux to save the uSDHC/XSPI content. The power mux switches from VDDN during Active mode to VDD2 during DSR mode. The power mux supplies VDDN_SR for retention logic in VDDN_MEDIA and VDDN_COM.

The power switch architecture establishes certain state dependencies and requirements between each of the power domains. For example, if VDD1_SENSE is off, all other domains (except VDD1V8_AON and VDD1V8_PMC) must be off. Below shows the possible states of other domains when each one of them is off.

Table 149. Power switch dependencies

When a power domain below is off...	These power domains must be...								
	VDD1_SENSE	VDD2_COM	VDD2_COMP	VDD2_DSP	VDD2_MEDIA	VDDN_MEDIA	VDDN_COM	VDD1V8_PMC	VDD1V8_AON
VDD1_SENSE	-	Off	Off	Off	Off	Off	Off	On or off	On
VDD2_COM	On or off	-	Off	Off	Off	Off	Off	On or off	On
VDD2_COMP	On or off	On or off	-	Off	On or off	On or off	On or off	On or off	On
VDD2_DSP	On or off	On or off	On or off	-	On or off	On or off	On or off	On or off	On
VDD2_MEDIA	On or off	On or off	On or off	On or off	-	Off	On or off	On or off	On
VDDN_MEDIA	On or off	On or off	On or off	On or off	Off	-	On or off	On or off	On
VDDN_COM	On or off	On or off	On or off	On or off	Off	Off	-	On or off	On
VDD1V8_PMC	Off	Off	Off	Off	Off	Off	Off	-	On or off
VDD1V8_AON	Off	Off	Off	Off	Off	Off	Off	Off	-

NOTE

Software must maintain the power domain state relationships shown in the table above. Hardware does not prevent illegal power switch configurations.

The power state dependencies between each of the power domains are also listed below:

- VDD1V8_AON, VDD1_SENSE, VDD2_COMP and VDD2_COM are default on after VDD1V8_PMC is powered on. All other domains are default off after initial power up.
- VDD2_DSP depends on VDD2_COMP. If VDD2_COMP is off, VDD2_DSP also must be off.
- VDD2_MEDIA and VDDN_MEDIA share one power switch control, so they both must be on or off together.
- VDDN_COM can be off only when both VDD2_MEDIA and VDDN_MEDIA are off.
- VDD2_COM can be off only when VDD2_COMP, VDD2_DSP, VDD2_MEDIA, VDDN_MEDIA and VDDN_COM are all off.
- VDD1_SENSE can be off only when VDD2_COM, VDD2_COMP, VDD2_DSP, VDD2_MEDIA, VDDN_MEDIA and VDDN_COM are all off.
- VDD1V8_PMC can be off only when VDD1_SENSE, VDD2_COM, VDD2_COMP, VDD2_DSP, VDD2_MEDIA, VDDN_MEDIA and VDDN_COM are all off. This is FDPD mode. In this mode, all other domains are powered off automatically, so no software enforcement is needed after entering that mode (and turning off the external power).

12.3 Power modes

The chip supports different power modes that moderate power consumption to match the performance requirements of the system applications active at any given time.

12.3.1 Active mode

In Active mode, clocks to a CPU, memories and modules are enabled. VDD2_COMP domain is active after reset, and the boot values of PMC.PDRUNCFG*n*, SLEEPCON.RUNCFG, and CLKCTL.PSCCTL*n* determine the power configuration requested by the domain. Runtime software can change the power configuration, programming modules to be in a normal power, low power, or off state.

12.3.1.1 Power optimization in Active mode

To achieve the performance requirements of a given system application while minimizing power consumption, Active mode permits the following power-saving options when possible:

- Configure CLKCTL.PSCCTL registers to control which memories and modules are running. To save power, turn off functional clocks that are not needed by the application. Clocks can be dynamically turned off and on as needed.
- For analog modules such as FRO, LPOSC, and PLL, configure SLEEPCON.RUNCFG to control the power. As with clock controls, turn off analog modules that are not needed by the application to save power. Note: If powered off, these analog modules require additional set up time for stabilization, whereas FRO outputs can be gated to save dynamic power without additional set up time.

  NOTE

  We recommend to power down FRO without clock gating by configuring SLEEPCON.RUNCFG[FRO*n*_PD] and [FRO*n*_GATE] bits.

- For RAM, configure PMC.PDRUNCFG2-5 to control the power. Use SLEEPCON.RUNCFG to control the RAM Arbiter clocks.
- In general, a lower clock frequency means lower power consumption. So running a CPU and other system functions at a frequency just sufficient for the application (plus some margin) saves power. Select the best-suited clock sources for system functions from among the several options in the CLKCTL registers. Also, running PLLs at a lower VCO frequency saves power. Configure CLKCTL2.MAINPLL0PFD[PFD*n*_CLKGATE] and AUDIOPLL0PFD[PFD*n*_CLKGATE] bits to gate off unused PLL PFDs.
- Many modules use individual functional clocks with their own local clock dividers. Use these clock divider or gating registers to shut down individual module clocks if another system function still requires an upstream shared clock source.

12.3.2 Sleep mode

In Sleep mode, power domains remain powered, but the CPU clock is gated off via a WFI/WFE instruction. Execution of instructions is suspended until either a reset or an interrupt occurs. Module functions, if selected to be clocked in CLKCTL.PSCCTL registers, continue operation and may generate interrupts to cause the processor to resume execution.

Sleep mode eliminates dynamic power used by the processor itself. Memory systems, related controllers, and internal buses continue to operate. The processor state and registers, module registers, and internal SRAM values are maintained. The logic levels of pins remain static unless affected by modules that remain in operation.

In Sleep mode, the configuration programmed into PMC.PDRUNCFG*n* and SLEEPCON.RUNCFG remain in effect.

12.3.2.1 Entering Sleep mode

CPU0 and CPU1 software can independently initiate Sleep mode. To enter Sleep mode, follow these steps:

1. In the NVIC, enable all interrupts that are needed to wake up the domain.
2. Configure PMC.PDRUNCFG*n* and SLEEPCON.RUNCFG to reflect any functions that need to be on or off during Sleep mode.
3. If you modify a PMC.PDRUNCFG*n* register, set PMC.CTRL[APPLYCFG] for the changes to take effect.
4. Execute a WFI/WFE instruction to enter Sleep mode.

12.3.2.2 Exiting Sleep mode

Sleep mode is exited automatically when an interrupt arrives at a CPU, or when a reset occurs. After a wake-up caused by an interrupt, CPU clocking resumes and the domain enters Active mode. Changes to PMC.PDRUNCFG*n* or SLEEPCON.RUNCFG done during the Sleep mode entry sequence can be undone to return to the previous Active condition. If a reset occurs, the domain enters the default configuration in Active mode.

12.3.2.3 Power optimization in Sleep mode

Power optimization in Sleep mode is similar to optimizing in Active mode:

- Select modules that are to remain powered during Sleep through PMC.PDRUNCFG*n* and SLEEPCON.RUNCFG.
- Enabled clocks remain running, so disable any clocks not needed.

12.3.3 Deep Sleep mode

In Deep Sleep mode, SLEEPCON.SLEEPCFG and PMC.PDSLEEPCFG*n* take effect. You should shut off specific bus clocks and power down unused clock sources, ADC0, the main domain and module clocks. You can also switch off unused power domains; however, the power domain containing the controlling SLEEPCON instance must remain powered to handle a wake-up event. For example, VDD2_COM must be powered to allow SLEEPCON0 to wake VDD2_COMP.

GPIO Pin Interrupts, GPIO Group Interrupts, and selected modules (such as USB, MICFIL, LP_FLEXCOMM, WWDT, RTC, and UTICK) and primary clock sources (such as FRO), the RTC oscillator, and the LPOSC can be left running in Deep Sleep mode. For supporting modules, DMA can operate in Deep Sleep mode; see Temporary wakeup for DMA.

12.3.3.1 Entering Deep Sleep mode

CPU0 and CPU1 software can independently initiate Deep Sleep mode. To enter Deep Sleep mode, follow these steps:

1. Configure PMC.PDSLEEPCFG*n* and SLEEPCON.SLEEPCFG to reflect any functions that need to be on or off.
2. For unused modules, use SLEEPCON to perform the low-power request-and-acknowledge handshake before gating a module's clock in its associated CLKCTL.PSCCTL register.
3. Configure SLEEPCON.WAKEUPEN*n* registers to enable the desired wakeup sources.
4. Configure SLEEPCON.HW_WAKE registers to enable the desired DMA wakeup sources.
5. For modules selected as wakeup sources, configure the appropriate wakeup conditions to generate interrupts.

6. In the associated NVIC, enable all interrupts that are needed to wake up the domain.
7. Configure the CPU's SCR register to enable SLEEPDEEP.
8. Execute a WFI/WFE instruction.

12.3.3.2 Exiting Deep Sleep mode

A domain can wake from Deep Sleep mode using many options, such as:

- A watchdog interrupt or RTC interrupt
- A reset from RESETN (RESETN pin assertion also resets the system.)
- A reset from the WWDT
- A wake-up signal from a module that is operating in Deep Sleep mode and capable of an asynchronous interrupt
- GPIO interrupts
- PINT pin interrupts
- PMIC_IRQN pin through PMC0 interrupt

See Wake-up sources for a complete list of wake-up sources in Deep Sleep mode.

NOTE

An interrupt used as a wake-up source must also be enabled in NVIC and in SLEEPCON.WAKEUPEN*n*.

12.3.3.3 Configuring MIPI_DSI_HOST for Deep Sleep entry and exit

For applications using MIPI_DSI_HOST connected to an external display, include the steps below before entering Deep Sleep:

1. Configure the display for Deep Sleep mode.
2. Disable MIPI_DSI_HOST (Power Down Register MIPI_DSI_HOST.PD_DPHY[PD_DPHY]).
3. Configure PMC.PDSLEEPCFG0[V2MIPI_PD].

After exiting Deep Sleep, follow the steps below:

1. Enable MIPI_DSI_HOST.
2. Configure the display to exit from Deep Sleep mode.

12.3.3.4 Temporary wakeup for DMA

Some modules support the option to send a DMA request in Deep Sleep mode. In this case, the domain wakes up just enough to perform the requested DMA before returning to Deep Sleep mode without waking the CPU. Follow the general enablement procedure below for the modules that support temporary DMA wake-up before entering Deep Sleep mode.

1. Configure the required module operation for normal DMA activity, including FIFO transfer threshold watermarks if available.
2. Configure the DMA controller, including a transfer complete interrupt.
3. Enable the DMA interrupt in the NVIC.
4. Enable DMA wakeup for the module by writing 1 to the related HW_WAKE[HWWK_SRC*n*] bit in SLEEPCON0 or SLEEPCON1.

NOTE

If a DMA transaction hangs, the chip remains in a higher power state until an interrupt or other wakeup source can wakeup the core.

An interrupt that occurs during a DMA wakeup causes an exit from Deep Sleep mode without affecting the DMA transfer.

12.3.3.4.1 Modules supporting DMA wakeup

The modules that support temporary DMA wakeup are listed below.

Table 150. Modules supporting temporary DMA wakeup

Module	DMA wakeup enable bit
FLEXIO	SLEEPCON0.HW_WAKE[HWWK_SRC1] or SLEEPCON1.HW_WAKE[HWWK_SRC1]
LP_FLEXCOMM0-13	SLEEPCON0.HW_WAKE[HWWK_SRC2]
LP_FLEXCOMM17-20	SLEEPCON1.HW_WAKE[HWWK_SRC2]
LPSPI14, LPSPI16	SLEEPCON0.HW_WAKE[HWWK_SRC3]
SAI0-2	SLEEPCON0.HW_WAKE[HWWK_SRC4]
SAI3	SLEEPCON1.HW_WAKE[HWWK_SRC4]

12.3.3.4.2 DMA wakeup considerations for communications modules

When configuring a serial communications module for DMA wakeup, you might need to enable both interrupts and DMA requests. For example, FIFO servicing requires enabling the relevant DMA request, but errors are generally handled only through interrupts. Errors include both FIFO underflow and overflow, and also could include protocol errors, such as parity errors or NACKs.

12.3.3.5 Double wakeup considerations

If the wakeup is due to a HW_WAKE event and an overlapping normal wakeup event occurs, the CPU starts executing software immediately. If software subsequently attempts to access memories that are in the process of being powered back on, a memory fault can occur. Avoid configuring a module to trigger both types of wakeup simultaneously.

12.3.3.6 Power optimization in Deep Sleep mode

Deep Sleep mode is configurable, supporting the option to turn off nearly all power consumption on a domain power supply at the cost of a longer wake-up time. Compute/Sense domain supply voltage can also be reduced to the minimum retention level in this mode.

NOTE

When lowering supply voltages connected to clocks, reduce their frequencies according to the specifications provided in the data sheet as needed.

Power consumption in Deep Sleep mode is determined primarily by which wake-up sources remain enabled. Dynamic power used by the processor itself, memory systems and related controllers, and internal buses (other than those specifically designated to remain functional) is eliminated. Serial peripherals and pin interrupts configured as wake-up sources contribute to the power consumption only to the extent that they are clocked.

12.3.3.7 Deep Sleep Async mode

Deep Sleep Async mode is a special sub-mode of Deep Sleep. Generally the operation is the same as Deep Sleep (SLEEPCON.SLEEPCFG and PMC.PDSLEEPCFGn take effect), but to enter Deep Sleep Async:

- The compute domain can enter Deep Sleep Async while the sense domain stays active, or both compute and sense domains can enter Deep Sleep Async.
- Clock sources within the domain(s), bus clocks, and peripheral clocks within the domain(s) must be disabled. If both domains are in Deep Sleep Async, then the only allowed clock is the RTC oscillator in the VDD1V8_AON domain.

Disabling all of the clocks allows for decreasing the voltage even lower than regular Deep Sleep. See the i.MX RT700 Crossover Microcontroller Data Sheet.

Only modules that can support asynchronous operation, such as GPIO, can be used to wakeup from Deep Sleep Async.

12.3.3.7.1 Entering Deep Sleep Async mode

To enter Deep Sleep Async mode, follow these steps:

1. Configure PMC.PDSLEEPCFGn and SLEEPCON.SLEEPCFG to reflect any functions that need to be on or off. Clock sources within the domain(s) must be configured to power down. If both compute and sense are entering Deep Sleep Async, then PLLs must be powered down.
2. For unused modules, use SLEEPCON to perform the low-power request-and-acknowledge handshake before gating a module's clock in its associated CLKCTL.PSCCTL register.
3. Configure SLEEPCON.WAKEUPENn registers to enable the desired wakeup sources.
4. For modules selected as wakeup sources, configure the appropriate wakeup conditions to generate interrupts.
5. In the associated NVICs, enable all interrupts that are needed to wake up each domain.
6. Configure CPU's SCR register to enable SLEEPDEEP.
7. Execute a WFI/WFE instructions.
8. After the domain has gone to Deep Sleep Async mode, then the voltage can be decreased.

12.3.3.7.2 Exiting Deep Sleep Async mode

A domain can wake from Deep Sleep Async using interrupts from modules that support asynchronous operations such as GPIOs. See Table 164 for a complete list of wake-up sources in Deep Sleep Async mode.

On wakeup from Deep Sleep Async, the VDD1/VDD2 voltages must be raised back to active levels corresponding to the frequency used by each domain.

NOTE

An interrupt used as a wake-up source must also be enabled in NVIC and in SLEEPCON.WAKEUPn.

12.3.4 Deep Sleep Retention (DSR)

Deep Sleep Retention (DSR) mode allows significant portions of the chip to be powered-down while maintaining some level of state retention, particularly the state of CPU0 and other system-critical functions. Depending on which portions of the chip are to be retained and power-gated, power management hardware automatically ensures that no operations are ongoing and that clocks are parked properly before entering DSR.

You can switch off unused power domains; however, the power domain containing the controlling SLEEPCON instance must remain powered to handle a wake-up event. For example, VDD2_COM must be powered to allow SLEEPCON0 to wake VDD2_COMP.

DSR is a deeper low-power state than Deep Sleep mode. Unlike Deep Sleep mode, DSR does not provide the option to keep selected module clocks on for wake-up. For example, if the Compute, Media, and Common domains are placed in DSR, the modules supporting DSR would have their states retained and all modules within those domains would not operate.

Below shows a complete list of modules that support DSR.

Table 151. Modules supporting DSR

VDD1_SENSE	VDD2_COMP	VDD2_MEDIA	VDD2_COM	VDDN_COM
CPU1	CPU0	EZH-V	RSTCTL0	RSTCTL2
CLKCTL1,3	ELS	RSTCTL4	CLKCTL0	CLKCTL2
FRO2	XSPI0-1 (Controller)	CLKCTL4	SYSCON0	SYSCON2
RSTCTL1,3	IPED0-1	SYSCON4	SLEEPCON0	XSPI0-2 (Memory interface)
SYSCON1,3	CDOG0-2	MMU2	FRO0-1	

Table continues on the next page...

Table 151. Modules supporting DSR (continued)

SLEEPCON1 AHBSC3 IOPCTL1 (GPIO8-10) GDET3 CDOG3-4 LPI2C15	MMU0-1 CACHE64_CTRL0-1 AHBSC0 ITRC GDET0	AHBSC4 XSPI2 (Controller) LCDIF MIPI_DSI_HOST -	IOPCTL0 (GPIO0-3) OCOTP	IOPCTL2 (GPIO4-7)
			VDD2_DSP	**VDDN_MEDIA**
			-	uSDHC0-1
			VDD1V8_AON	**VDD1V8_PMC**
			-	-

12.3.4.1 Sense domain entering DSR

The Sense domain can enter DSR mode only when all other domains are in DSR mode. CPU1 software must first check that all other domains are in DSR before entering DSR. If the condition is true, CPU1 can program the RTC timer to schedule a wake-up and enter DSR mode. If the condition is false, CPU1 software might choose to go into a shallower low power mode (Sleep or Deep Sleep mode) because the Sense domain cannot enter DSR mode in this case.

NOTE

When both the Compute and Sense domains are in DSR, a wake-up event for Compute requires both domains to exit DSR because part of the Sense domain needs to be active for Compute to operate.

12.3.4.2 DSR combinations

Only certain combinations of logic being place in DSR mode are valid. Because DSR involves power gating, the rules are the same as for power gating of domains. See Table 149 for a full list of rules.

The following sections present example valid combinations and their wake-up sources.

12.3.4.2.1 Compute, Media, Sense, and Common in DSR (Full DSR)

The following table summarizes the power domain states when the Compute, Media, Sense, and Common processing domains are in DSR.

Table 152. Compute, Media, Sense, and Common in DSR

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Off
VDD2_MEDIA	Off
VDDN_MEDIA	Off
VDD2_COM	Off
VDDN_COM	Off
VDD1V8_PMC	Powered by VDD1V8
VDD1V8_AON	Powered by VDD1V8_AO

Full DSR (FDSR) is a chip level mode. To enter FDSR both CPUs must execute WFI/WFE with their corresponding FDSR bit set.

Because Compute, Media and Sense are all in DSR mode, the only source of the wake-ups can be the external wake-ups similar to DPD and FDPD.

Table 153. Wake-up sources for FDSR

Source	Configuration
PMIC_IRQN, RTC_ALARM, RTC_WAKE	Enable the related function in SLEEPCON.WAKEUPEN*n*.
RESETN	Configuration is not needed. **NOTE** Asserting RESETN wakes the chip but also causes the whole chip to reset.

12.3.4.2.2 Compute, Media, and Common in DSR (Hub mode)

The following table summarizes the power domain states when the Compute, Media, and Common processing domains are in DSR.

Table 154. Compute, Media, and Common in DSR

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Powered by VDD1
VDD2_MEDIA	Off
VDDN_MEDIA	Off
VDD2_COM	Off
VDDN_COM	Off
VDD1V8_PMC	Powered by VDD1V8
VDD1V8_AON	Powered by VDD1V8_AO

This low-power combination with Compute, Media, and Common in DSR is known as Hub mode. In Hub mode, the chip can collect and process data at low power. Use Hub mode for applications that only need the Sense domain functions until some condition requires activating other chip resources.

Because the Sense domain remains active in Hub mode, you have more options for waking the chip. You can program timers to wake the Compute domain. Alternatively, CPU1 can communicate a wake-up event through the messaging unit to interrupt CPU0, causing it to wake.

Table 155. Wake-up sources for Compute domain (Hub mode)

Source	Configuration
Interrupts from VDD1_SENSE modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
OSTIMER interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.
MU interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.

Table continues on the next page...

Table 155. Wake-up sources for Compute domain (Hub mode) (continued)

Source	Configuration
ADC0, ACMP, MICFIL, PMIC_IRQN, RTC_ALARM, RTC_WAKE	Enable the related function in SLEEPCON.WAKEUPEN*n*.
RESETN	Configuration is not needed. **NOTE** Asserting RESETN wakes the chip but also causes the whole chip to reset.

12.3.4.2.3 Compute and Media in DSR

The following table summarizes the power domain states when the Compute and Media processing domains are in DSR.

Table 156. Compute and Media in DSR

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Powered by VDD1
VDD2_MEDIA	Off
VDDN_MEDIA	Off
VDD2_COM	Powered by VDD2
VDDN_COM	Off
VDD1V8_PMC	Powered by VDD1V8
VDD1V8_AON	Powered by VDD1V8_AO

Because both the Sense and Common domains remain active in this scenario, you have more options for waking the chip. You can program timers to wake the Compute domain. Alternatively, CPU1 can communicate a wake-up event through the messaging unit to interrupt CPU0, causing it to wake.

Table 157. Wake-up sources for Compute domain (Compute and Media in DSR)

Source	Configuration
Interrupts from VDD2_COM modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
Interrupts from VDD1_SENSE modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
OSTIMER interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.
MU interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.
ADC0, ACMP, MICFIL, PMIC_IRQN, RTC_ALARM, RTC_WAKE	Enable the related function in SLEEPCON.WAKEUPEN*n*.

Table continues on the next page...

Table 157. Wake-up sources for Compute domain (Compute and Media in DSR) (continued)

Source	Configuration
RESETN	Configuration is not needed. **NOTE** Asserting RESETN wakes the chip but also causes the whole chip to reset.

12.3.4.2.4 Compute in DSR

The following table summarizes the power domain states when the Compute processing domain is in DSR.

Table 158. Compute in DSR

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Powered by VDD1
VDD2_MEDIA	Powered by VDD2
VDDN_MEDIA	Powered by VDDN
VDD2_COM	Powered by VDD2
VDDN_COM	Powered by VDDN
VDD1V8_PMC	Powered by VDD1V8
VDD1V8_AON	Powered by VDD1V8_AO

Because the Sense, Media, and Common domains all remain active in this scenario, you have more options for waking the chip. You can program timers to wake the Compute domain. Alternatively, CPU1 can communicate a wake-up event through the messaging unit to interrupt CPU0, causing it to wake.

Table 159. Wake-up sources for Compute domain (Compute in DSR)

Source	Configuration
Interrupts from VDDN_COM modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
Interrupts from VDDN_MEDIA modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
Interrupts from VDD2_MEDIA modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
Interrupts from VDD2_COM modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
Interrupts from VDD1_SENSE modules that remain active	Enable the interrupt in both the module itself and the associated interrupt controller.
OSTIMER interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.

Table continues on the next page...

Table 159. Wake-up sources for Compute domain (Compute in DSR) (continued)

Source	Configuration
MU interrupt	Enable the interrupt in both the module itself and the associated interrupt controller.
ADC0, ACMP, MICFIL, PMIC_IRQN, RTC_ALARM, RTC_WAKE	Enable the related function in SLEEPCON.WAKEUPEN*n*.
RESETN	Configuration is not needed. **NOTE** Asserting RESETN wakes the chip but also causes the whole chip to reset.

12.3.5 Deep Power Down (DPD) mode

In DPD mode, power and clocks are shut off to the entire chip with the exception of the RTC. SRAM content and module registers (other than those in the RTC) are not retained. All functional pins are tristated as long as VDDIO power rails are supplied externally. On-chip regulators are off. VDD1V8 and VDD1V8_AO remain on. PMC retains state. VDD1V8 monitors stay on. Voltage rail definitions are retained. TRIM is maintained allowing for less voltage margins needed on wake-up.

DPD is a chip level mode. To enter DPD both CPUs must execute WFI/WFE with their corresponding DPD bit set.

The following table summarizes the power domain states in DPD mode.

Table 160. Power domain states in DPD mode

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Off
VDD2_MEDIA	Off
VDDN_MEDIA	Off
VDD2_COM	Off
VDDN_COM	Off
VDD1V8_PMC	Powered by VDD1V8
VDD1V8_AON	Powered by VDD1V8_AO

12.3.5.1 Entering DPD mode

DPD mode has no configuration options. All clocks, the core, and all peripherals are powered down. Only the RTC is powered, as long as power is supplied to VDD1V8_AO.

To enter DPD mode when using the RTC for waking up, follow these steps:

1. Set up the RTC high resolution timer. Write to the VAL bits in WAKE register. [Write to the RTC VAL register.] This starts the high resolution timer if enabled. Alternatively, another option is to use the 1-Hz alarm timer.
2. Set the associated PMC PDSLEEPCFG0[DPD] bit.
3. Configure the CPU's SCR register to enable SLEEPDEEP.
4. Execute a WFI/WFE instruction.

12.3.5.2 Exiting DPD mode

A wake-up event triggers exit from DPD mode. The chip undergoes a power-on-reset and system boot. After boot, all registers are in their reset states except those retained by the VDD1V8_AO supply, such as the RTC registers.

Table 161. Wake-up sources for DPD

Source	Configuration
PMIC_IRQN, RTC_ALARM, RTC_WAKE	Enable the related function in SLEEPCON.WAKEUPEN*n*.
RESETN	Not needed.

12.3.6 Full Deep Power Down (FDPD) mode

Similar to DPD mode, FDPD mode is the chip's lowest power mode option where only VDD1V8_AO is powered. An RTC event, an external interrupt, or an external reset can trigger wake-up. The PMIC mode control pins remain powered to enable an optional external PMIC upon a wake-up event.

FDPD is a chip level mode. To enter FDPD both CPUs must execute WFI/WFE with their corresponding FDPD bit set.

The following table summarizes the power domain states in FDPD mode.

Table 162. Power domain states in FDPD mode

Power domain	Power switched
VDD2_COMP	Off
VDD2_DSP	Off
VDD1_SENSE	Off
VDD2_MEDIA	Off
VDDN_MEDIA	Off
VDD2_COM	Off
VDDN_COM	Off
VDD1V8_PMC	Off
VDD1V8_AON	Powered by VDD1V8_AO

12.4 Power mode transitions

The VDD2_COMP and VDD1_SENSE domains can move from Active to Sleep or Deep Sleep from the Active mode of operation. VDD2_COMP can further move to DSR from Active mode, but VDD1_SENSE cannot independently move to DSR before VDD2_COMP has entered DSR. The deeper low-power modes (DPD and FDPD) can occur only when both domains signal their intent to enter these lower power modes. The valid combinations of power modes are shown below:

Table 163. Valid power mode combinations of the VDD2_COMP and VDD1_SENSE domains

		VDD2_COMP power modes					
		Active	Sleep	Deep Sleep	DSR	DPD	FDPD
VDD1_SENSE power modes	Active	Valid	Valid	Valid	Valid	—	—
	Sleep	Valid	Valid	Valid	Valid	—	—
	Deep Sleep	Valid	Valid	Valid	Valid	—	—
	DSR	—	—	—	Valid	—	—
	DPD	—	—	—	—	Valid	—
	FDPD	—	—	—	—	—	Valid

Below is the overall state diagram of power mode transitions:

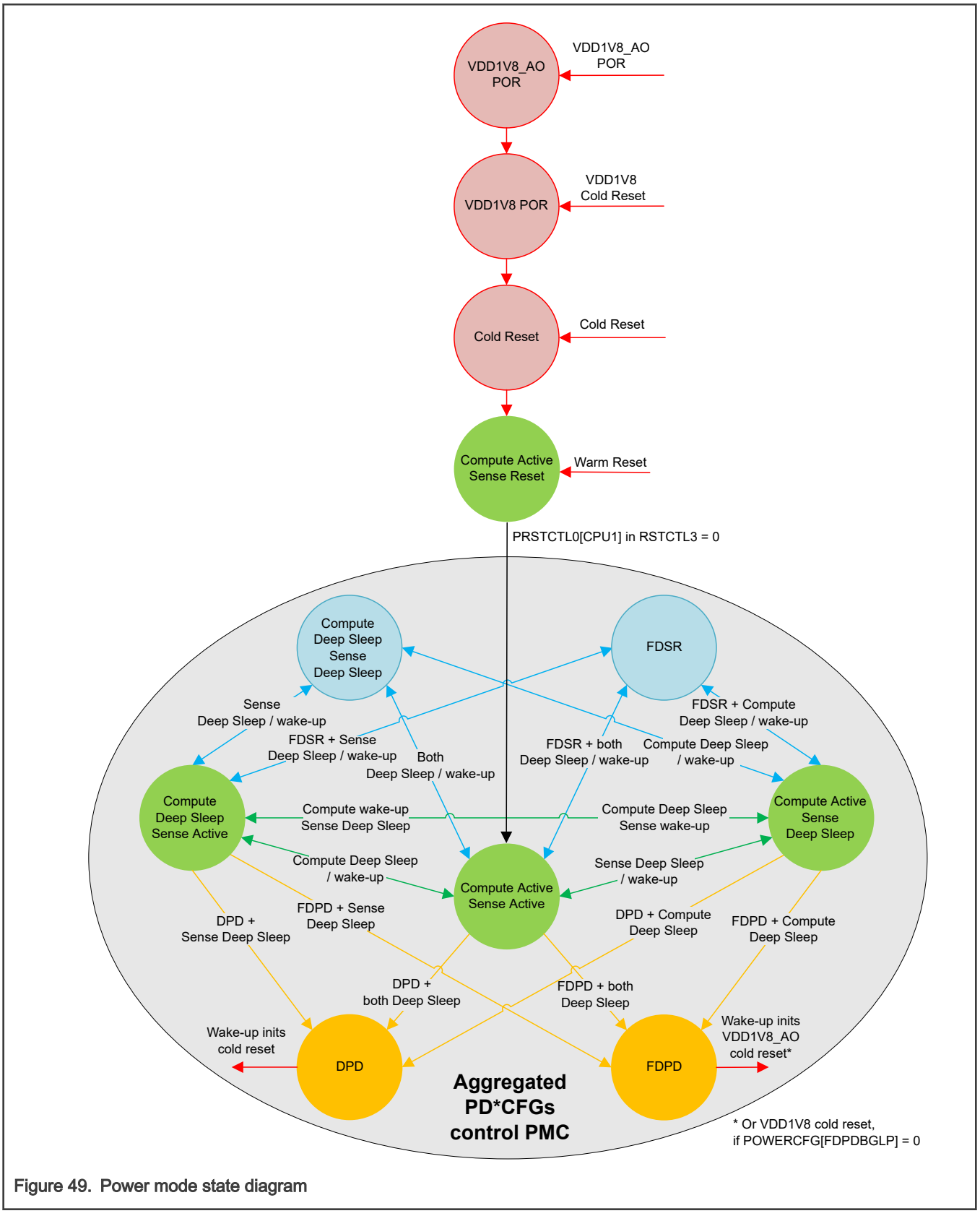


Figure 49. Power mode state diagram

12.4.1 Wake-up sources

The tables below list all wake-up sources for CPU0 and CPU1 in the different power modes. In the “Async Interrupt” column:

- “Y” means the source is asynchronous and can be asserted without a bus clock.
- “Available” means the source interrupt can be either synchronous or asynchronous. When the bus clock is gated off, an asynchronous interrupt can still be trigged by the module’s functional clock.

In the “CPU0” and “CPU1” columns, “Y” means the source can wake up the CPU if:

- For Sleep mode, the module must be clocked, and the module’s power domain must be powered. The corresponding NVIC channel must also be enabled.
- For Deep Sleep mode, the bus clock is off, so only modules capable of asynchronous interrupts can be wake up sources. The module’s functional clock must not be gated, and the corresponding SLEEPCON must be powered.

Table 164. Wake-up sources for different power mode

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
RESETN pin	Y	Y	Y	Y	Y	Y	Y	Y	Y
ACMP0: interrupt request	Available	Y	Y	Y	Y	Y[1]	Y[1]		
ADC0: interrupt request	Available	Y	Y	Y	Y				
CDOG0: interrupt request		Y							
CDOG1: interrupt request		Y							
CDOG2: interrupt request		Y							
CDOG3: interrupt request			Y						
CDOG4: interrupt request			Y						
ELS50: interrupt request		Y							
CPU0: CTI interrupt request (with BIST interrupt)		Y							
CPU1: CTI interrupt request			Y						
CTIMER0: interrupt request	Available	Y		Y					
CTIMER1: interrupt request	Available	Y		Y					
CTIMER2: interrupt request	Available	Y		Y					
CTIMER3: interrupt request	Available	Y		Y					
CTIMER4: interrupt request	Available	Y		Y					
CTIMER5: interrupt request	Available		Y		Y				
CTIMER6: interrupt request	Available		Y		Y				
CTIMER7: interrupt request	Available		Y		Y				
LCDIF: interrupt request		Y	Y						
MIPI_DSI_HOST: interrupt request		Y	Y						
eDMA0: Channel 0 – 15 interrupt		Y							

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
eDMA1: Channel 0 – 15 interrupt		Y							
eDMA2: Channel 0 – 7 interrupt			Y						
eDMA3: Channel 0 – 7 interrupt			Y						
EZH-V: interrupt request		Y	Y						
LP_FLEXCOMM0: interrupt request	Available	Y		Y					
LP_FLEXCOMM1: interrupt request	Available	Y		Y					
LP_FLEXCOMM2: interrupt request	Available	Y		Y					
LP_FLEXCOMM3: interrupt request	Available	Y		Y					
LP_FLEXCOMM4: interrupt request	Available	Y		Y					
LP_FLEXCOMM5: interrupt request	Available	Y		Y					
LP_FLEXCOMM6: interrupt request	Available	Y		Y					
LP_FLEXCOMM7: interrupt request	Available	Y		Y					
LP_FLEXCOMM8: interrupt request	Available	Y		Y					
LP_FLEXCOMM9: interrupt request	Available	Y		Y					
LP_FLEXCOMM10: interrupt request	Available	Y		Y					
LP_FLEXCOMM11: interrupt request	Available	Y		Y					
LP_FLEXCOMM12: interrupt request	Available	Y		Y					
LP_FLEXCOMM13: interrupt request	Available	Y		Y					
CLKCTL0: FRO0 in VDD2 domain Async interrupt	Y	Y	Y	Y	Y				
CLKCTL0: FRO1 in VDD2 domain Async interrupt	Y	Y	Y	Y	Y				
CLKCTL3: FRO2 in VDD1 domain Async interrupt	Y	Y	Y	Y	Y				
LPSPI14: interrupt request	Available	Y		Y					
LPI2C15: interrupt request	Available	Y	Y	Y	Y				
LPSPI16: interrupt request	Available	Y		Y					
LP_FLEXCOMM17: interrupt request	Available		Y		Y				

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
LP_FLEXCOMM18: interrupt request	Available		Y		Y				
LP_FLEXCOMM19: interrupt request	Available		Y		Y				
LP_FLEXCOMM20: interrupt request	Available		Y		Y				
FLEXIO: interrupt request	Available	Y	Y	Y	Y	Y[2]	Y[2]		
FREQME: interrupt request		Y							
GDET0: interrupt request		Y							
GDET3: interrupt request		Y	Y						
GLIKEY0: interrupt		Y							
GLIKEY1: interrupt			Y						
GLIKEY2: interrupt		Y							
GLIKEY3: interrupt		Y							
GLIKEY4: interrupt			Y						
GLIKEY5: interrupt		Y							
VGPU: interrupt from graphics core (Domain HCLK)		Y	Y						
HiFi1: interrupt request			Y						
HiFi4: interrupt request		Y							
I3C0: interrupt request	Available	Y		Y					
I3C1: interrupt request	Available	Y		Y					
I3C2: interrupt request	Available		Y		Y				
I3C3: interrupt request	Available		Y		Y				
PMIC_IRQN pin	Y	Y	Y	Y	Y	Y	Y	Y	Y
ITRC: interrupt, cleared by software	Y	Y		Y		Y			
JPEGDEC: ORed Context 0-3 interrupt request		Y	Y						
Sense Domain Hypervisor interrupt		Y	Y						
Sense Domain Secure violation interrupt		Y	Y						
Media Domain Hypervisor interrupt		Y							

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
Media Domain Secure violation interrupt		Y							
MICFIL: Hardware Voice Activity Detector interrupt or error interrupt	Available	Y	Y	Y	Y				
MICFIL: interrupt request for read data or Error	Available	Y	Y	Y	Y				
MMU0: interrupt request		Y							
MMU1: interrupt request		Y							
MMU2: interrupt request		Y	Y						
MRT0: ORed interrupt request		Y							
MRT1: ORed interrupt request			Y						
MU0: MUA, CPU0 to HiFi1	Y	Y		Y					
MU0: MUB, HiFi1 to CPU0	Y								
MU1: MUA, CPU0 to CPU1	Y	Y		Y					
MU1: MUB, CPU1 to CPU0	Y		Y		Y				
MU2: MUA, HiFi4 to CPU1	Y								
MU2: MUB, CPU1 to HiFi4	Y		Y		Y				
MU3: MUA, CPU1 to HiFi1	Y		Y		Y				
MU3: MUB, HiFi1 to CPU1	Y								
MU4: MUA, CPU0 to HiFi4	Y	Y		Y					
MU4: MUB, HiFi4 to CPU0	Y								
NPU: interrupt request		Y							
OSTIMER: Event timer CPU0 wake-up/interrupt	Available	Y		Y					
OSTIMER: Event timer CPU1 wake-up/interrupt	Available		Y		Y				
OSTIMER: Event timer HiFi1 wake-up/interrupt	Available								
OSTIMER: Event timer HiFi4 wake-up/interrupt	Available								
Compute Domain P-Bus Hypervisor interrupt		Y							
Compute Domain P-Bus Secure violation interrupt		Y							

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
PINT0: interrupt request 0 – 7	Available	Y		Y		Y			
PINT1: interrupt request 0 – 3	Available		Y		Y		Y		
PKC: Public Key Cryptography interrupt		Y							
PNGDEC: interrupt request		Y	Y						
PVTS0: PVTS0CPU0 interrupt		Y							
PVTS0: PVTS0HiFi4 interrupt		Y							
PVTS1: PVTS1CPU1 interrupt			Y						
PVTS1: PVTS1HiFi1 interrupt			Y						
PUF: interrupt request		Y							
GPIO0: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO1: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO2: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO3: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO4: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO5: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO6: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO7: interrupt request, channel 0 – 1	Available	Y		Y		Y			
GPIO8: interrupt request, channel 0 – 1	Available		Y		Y		Y		
GPIO9: interrupt request, channel 0 – 1	Available		Y		Y		Y		
GPIO10: interrupt request, channel 0 – 1	Available		Y		Y		Y		
RTC0: alarm interrupt to Compute Domain	Y	Y		Y		Y		Y	
RTC0: wakeup interrupt to Compute Domain	Y	Y		Y		Y		Y	

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
RTC1: alarm interrupt to Sense Domain	Y		Y		Y		Y		Y
RTC1: wakeup interrupt to Sense Domain	Y		Y		Y		Y		Y
OCOTP: Warning and Integrity Error interrupt		Y							
OSCCA: OSCCA interrupt request		Y							
SAI0: TX/RX interrupt	Available	Y		Y					
SAI1: TX/RX interrupt	Available	Y		Y					
SAI2: TX/RX interrupt	Available	Y		Y					
SAI3: TX/RX interrupt	Available		Y		Y				
SCT0: interrupt request	Y	Y							
SDADC: interrupt request		Y	Y						
TEMPDETECT interrupt request 0		Y	Y						
TEMPDETECT interrupt request 1		Y	Y						
TRNG: interrupt		Y							
USB0: HSUSB interrupt request		Y	Y						
USB1: eUSB interrupt request		Y	Y						
USBPHY: UTM interrupt request		Y	Y						
uSDHC0: interrupt request		Y	Y						
uSDHC1: interrupt request		Y	Y						
UTICK0: interrupt request	Available	Y		Y					
UTICK1: interrupt request	Available		Y		Y				
WWDT0: interrupt request	Y	Y		Y					
WWDT1: interrupt request	Y	Y		Y					
WWDT2: interrupt request	Y		Y		Y				
WWDT3: interrupt request	Y		Y		Y				
XSPI0: ORed interrupt and Flash Error interrupt		Y							
XSPI1: ORed interrupt and Flash Error interrupt		Y							
XSPI2: ORed interrupt and Flash Error interrupt		Y	Y						

Table continues on the next page...

Table 164. Wake-up sources for different power mode (continued)

Source	Async interrupt	Sleep		Deep Sleep		Deep Sleep Async		FDSR	
		CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
PMC1: Sense Domain interrupt			Y		Y		Y		
PMC0: Compute Domain interrupt		Y		Y		Y			

1. Filters and window mode must be disabled for Deep Sleep Async, because these functions require a clock.
2. Only pin interrupts (PSF) can be used for Deep Sleep Async wakeup.

12.5 SLEEPCON and PMC

The two SLEEPCON instances, SLEEPCON0 and SLEEPCON1, initiate clock gating with CLKCTL and control which clock sources to power down. Each SLEEPCON contains two register banks: SLEEPCFG and RUNCFG. The power status of the associated CPU determines which register bank controls the power state of the modules:

- When CPU is in Deep Sleep or a deeper low-power mode, SLEEPCFG applies.
- When CPU is in Active or Sleep mode, RUNCFG applies.

Software needs to configure the SLEEPCON registers before entering Deep Sleep. After SLEEPCON has completed its tasks, it automatically transfers control to PMC to start taking action.

PMC controls power mode transitions. Like SLEEPCON, PMC also has two register banks: PDSLEEPCFG and PDRUNCFG. PMC uses the power status from SLEEPCON to select which register bank takes effect. Because the two configuration values coming from the Compute and Sense domains may be different, PMC aggregation ensures the higher power mode always wins.

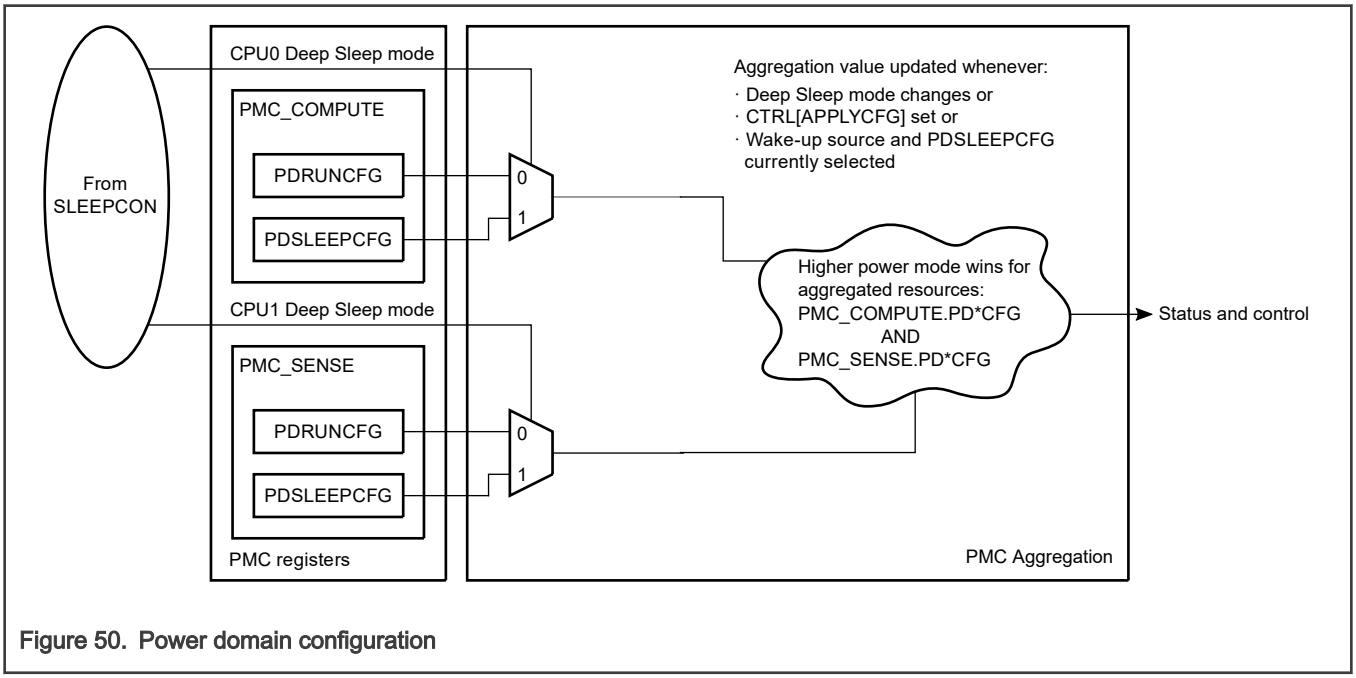


Figure 50. Power domain configuration

12.5.1 SLEEPCON dependencies

SLEEPCON0 and SLEEPCON1 are both sleep configuration modules but have the following differences and dependencies:

- CPU0 can access SLEEPCON0.
- CPU1 can access SLEEPCON1.

- SLEEPCON1 controls the modules in the VDD1_SENSE domain (default state is all off), clock source PLL and OSC in VDDN_COM, and the RAM_ARBITER0 main clock in VDD2_COM and VDD2/VDDN_MEDIA. SLEEPCON0 cannot control private modules in the VDD1_SENSE domain.
  - This means before VDD2 goes off, CPU0 software must appropriately configure modules outside the VDD1_SENSE domain. CPU1 software has no control over the power states of modules connected to VDD2 or VDDN.
  - For example, if a VDD2_MEDIA module needs to remain active, VDD2 must remain on.
- Because shared modules in VDD1_SENSE have duplicated controls in SLEEPCON1 and SLEEPCON0, if either SLEEPCON configures a shared module to be on, it remains on. Both SLEEPCONs must program a shared module to be off for it to be off.

12.5.2 Analog module power control in SLEEPCON

For analog modules such as FRO, LPOSC, PLL, the power control is in SLEEPCON:

- In Active mode and in Sleep mode, RUNCFG0 controls the power.
- In Deep Sleep mode and in DSR mode, SLEEPCFG0 controls the power.

The table below shows the module control fields and their bit numbers in the SLEEPCON0 and SLEEPCON1 configuration registers.

Table 165. SLEEPCON power control fields

Field	Description	SLEEPCON0		SLEEPCON1	
		SLEEPCFG0	RUNCFG0	SLEEPCFG0	RUNCFG0
FRO2_GATE	FRO2 Gating	[31]	[31]	[31]	[31]
FRO0_GATE	FRO0 Gating	[29]	[29]	-	-
ADC0_PD	ADC0 Power Down	[16]	[16]	[16]	[16]
AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down	[15]	[15]	[15]	[15]
AUDPLLANA_PD	Audio PLL Analog Function Power Down	[14]	[14]	[14]	[14]
PLLLDO_PD	Main PLL Internal Regulator Power Down	[13]	[13]	[13]	[13]
PLLANA_PD	Main PLL Analog Function Power Down	[12]	[12]	[12]	[12]
LPOSC_PD	LPOSC Power Down	[11]	[11]	[11]	[11]
FRO2_PD	FRO2 Power Down	[10]	[10]	[10]	[10]
FRO1_PD	FRO1 Power Down	[9]	[9]	-	-
FRO0_PD	FRO0 Power Down	[8]	[8]	-	-
XTAL_PD	Xtal Oscillator Power Down	[7]	[7]	[7]	[7]

To reduce the amount of inter-processor communication required to manage common resources controlled by both CPU0 and CPU1, power-control fields are available in the registers of both SLEEPCON0 and SLEEPCON1. AND logic determines what power state takes effect. For example, FR02 is powered down only if (SLEEPCON0[FRO2_PD] = 1) AND (SLEEPCON1[FRO2_PD] = 1) is true. If a CPU does not use a common resource, that CPU must put that resource in a power-down state.

12.6 RAM Arbiter and shared memory power considerations

The RAM Arbiter modules manage access to the on-chip shared RAM memory partitions. RAM_ARBITER0 in the VDD2_COM power domain manages accesses to the VDD2 shared memories. RAM_ARBITER1 in the VDD1_SENSE power domain manages accesses to the VDD1 shared memories.

This section describes power saving options and considerations for the RAM Arbiter modules.

12.6.1 Memory power switches

Each shared memory partition has two power switches that can independently turn off power to their periphery (address decoding and IOs) and array (memory content). If software does not need to access the memory partition, turn off the periphery power. If software does not need to retain the partition's memory content, turn off the array power also.

For active (run) mode, the PMC PDRUNCFG registers control these switches. For low-power modes, the PMC PDSLEEPCFG registers control these switches.

12.6.2 RAM Arbiter clocking

The RAM Arbiter modules do not have their own dedicated clocks. Instead, they use the clocks of the transaction initiators accessing the shared memory. These clocks, listed below, should be turned off whenever possible to maximize power saving.

Table 166. RAM_ARBITER0 clocks

Clock	Power control
CPU0 bus clock (COMPUTE_MAIN_CLK)	Automatically gated when no access
VDD1_SENSE Access RAM_ARBITER0 Clock (COMMON_RAM_CLK)	CLKCTL1.PSCCTL1[SENSE_ACCESS_RAM_ARBITER0]
HiFi4 clock	Clock off HiFi4 to turn it off
NPU clock	Clock off NPU to turn it off
VDD2_MEDIA Access RAM_ARBITER0 Clock (MEDIA_MAIN_CLK)	CLKCTL0.PSCCTL5[MEDIA_ACCESS_RAM_ARBITER0]

Table 167. RAM_ARBITER1 clocks

Clock	Power control
CPU1 bus clock (SENSE_MAIN_CLK)	Automatically gated when no access
VDD2_COMP Access RAM_ARBITER1 Clock (SENSE_RAM_CLK)	CLKCTL0.PSCCTL5[COMP_ACCESS_RAM_ARBITER1]
HiFi4 Access RAM_ARBITER1 Clock (SENSE_RAM_CLK)	CLKCTL0.PSCCTL5[HiFi4_ACCESS_RAM_ARBITER1]
HiFi1 clock	Clock off HiFi1 to turn it off
Media Access RAM_ARBITER1 Clock (SENSE_RAM_CLK)	CLKCTL3.PSCCTL0_COMP[MEDIA_ACCESS_RAM_ARBITER1] or CLKCTL3.PSCCTL0_SENS[MEDIA_ACCESS_RAM_ARBITER1]

12.6.3 Clock switching

Because a shared RAM partition being accessed connects to and remains parked on the last initiator's clock, a new access from another initiator with a different clock triggers clock switching inside the RAM Arbiter. This clock switching requires that both the last clock and the new clock are available, so prematurely turning off the last access clock prevents the clock switching and blocks any new access to that partition.

12.6.4 RAM auto clock gating when using CPU bus clock

When a RAM partition's clock is parked on the associated CPU bus clock, the partition is clocked only when accessed. That is, hardware automatically gates the CPU bus clock connection whenever possible to save power.

12.6.5 Safely turning off an initiator's clock using CPU dummy reads

In a power-sensitive system with multiple initiators sharing RAM partitions, system applications must know when initiators would use partitions and manage the initiators' clocks accordingly. If an initiator that does not use the CPU bus clock shares a RAM partition with others, the associated CPU (CPU0 for RAM_ARBITER0, CPU1 for RAM_ARBITER1) should first perform a dummy read to that partition before turning off that initiator's clock.

The CPU dummy read triggers the RAM Arbiter to switch the partition's clock to the CPU bus clock, so the clock turns on only during transactions to save power. See RAM auto clock gating when using CPU bus clock.

12.6.5.1 Cache considerations for CPU0 dummy reads

The CPU0 dummy reads must miss the cache to be effective. To ensure a RAM Arbiter is physically accessed, use one of the following methods:

- Access a non-cacheable region if available per the MPU configuration.
- Disable the cache, perform the dummy read, then re-enable the cache.
- Invalidate the cache line with a line command to the target address before performing the dummy read.

12.6.6 Accessing RAM_ARBITER0 with Compute in Deep Sleep

If a system application requires CPU1 or EZH-V to access RAM_ARBITER0 partitions while the Compute domain is in Deep Sleep mode with COMPUTE_MAIN_CLK off, CPU1 or EZH-V must perform a dummy read first before turning off the Compute domain bus clock.

12.6.7 Power saving options in SYSCON

In addition to the power switches, the shared memories have power saving options controlled by SYSCON registers:

- When no processor core or other memory transaction initiator requires a shared memory partition, gate the partition's input clock to save power as follows:
  1. Perform a dummy read to the target partition to establish the CM33 bus clock connection.
  2. Clear the corresponding SRAM_CLKGATE_CTRL[PARTITIONn] bit to gate the clock.

12.7 Chip operation in low-power modes

Chip components can operate differently depending on the power mode.

12.7.1 Module operation in low-power modes

The following table shows the functional states of the modules in low-power modes.

Table 168. Module states in low-power modes

Module or clock name	Main controlling entity	Active	Sleep	Deep Sleep	DSR	DPD	FDPD
FRO2 (325 MHz)	CPU1	On	Software-configured	Software-configured	Off with state retained	Off	Off

Table continues on the next page...

Table 168. Module states in low-power modes (continued)

Module or clock name	Main controlling entity	Active	Sleep	Deep Sleep	DSR	DPD	FDPD
LPOSC (1M_LPOSC)	CPU1	On	Software-configured	Software-configured	Off with state retained	Off	Off
FRO0 (325 MHz)	CPU0	On	Software-configured	Software-configured	Off with state retained	Off	Off
FRO1 (192 MHz)	CPU0	On	Software-configured	Software-configured	Off with state retained	Off	Off
Main crystal oscillator (MHz OSC)	CPU0	Software-configured[1]	Software-configured	Software-configured	Off	Off	Off
System PLL and Audio PLL (528 MHz)	CPU0	Software-configured	Software-configured	Software-configured	Off	Off	Off
OSC32KNP (RTC and RTC oscillator)	CPU0	Software-configured	Software-configured	Software-configured	Software-configured	Software-configured	Software-configured
SRAM memory arrays and peripheries	Controlled by both based on ownership	Software-configured	Software-configured	Software-configured	Software-configured	Off	Off
Boot ROM	CPU0	On	On	Off	Off	Off	Off
Other digital modules	Some controls are with CPU0, others with CPU1, rarely shared except few like OSTIMER etc.						
ADC	Shared between CPU0 and CPU1	Software-configured	Software-configured	Software-configured	Off	Off	Off
MIPI_DSI_HOST	CPU0	Software-configured	Software-configured	Software-configured	Off	Off	Off
DSP	Both CPU0 and CPU1 have their own DSPs	Software-configured	Software-configured	Software-configured	Off	Off	Off
OCOTP	CPU0	Software-configured	Software-configured	Software-configured	Off	Off	Off

1. Software-configured = On or off, clock-gated, or power-gated

12.7.2 Power switches in low-power modes

The following table shows the states of the power switches in low-power modes.

Table 169. Power switch states in low-power modes

Power switch	Controlling logic	Compute Domain power modes					
		Active	Sleep	Deep Sleep	DSR	DPD	FDPD
PS0_VDD2	VDDC switch for VDD2_COMP	Optional	Dep[1]	Dep	Dep	Dep	Dep
PS1_VDD2	VDD switch for VDD2_COMP	Ena[2]	Ena	Ena	Dis[3]	Dis	Dis
PS2_VDD2	VDDC switch for VDD2_MEDIA	Optional	Dep	Dep	Dep	Dep	Dep
PS3_VDD2	VDD switch for VDD2_MEDIA	Ena	Ena	Ena	Dis	Dis	Dis
PS4_VDD2	VDDC switch for VDD2_COM	Optional	Dep	Dep	Dep	Dep	Dep
PS5_VDD2	VDD2_COM	Ena	Ena	Ena	Dis	Dis	Dis
PS6_VDD2	VDD2_DSP	Prog[4]	Prog	Ena	Dis	Dis	Dis
PS7_VDD2	Memory periphery	Ena	Ena	Ena	Dis	Dis	Dis
PS8_VDD2	Memory array	Ena	Ena	Ena	Ena	Dis	Dis
PS0_VDDN	VDDN_MEDIA for DSR Power mux for retained logic	Ena	Ena	Ena	Dis	Dis	Dis
PS1_VDDN	VDDN_MEDIA for DSR Power mux for retained logic	Ena	Ena	Ena	Ena	Dis	Dis
PS2_VDDN	VDDN_MEDIA for non-retained logic	Ena	Ena	Ena	Dis	Dis	Dis
PS3_VDDN	VDDN_COM	Ena	Ena	Ena	Dis	Dis	Dis

1. Dep = Depends. State of power switch depends on domain controlling the logic
2. Ena = Enabled. Power switch closed
3. Dis = Disabled. Power switch open
4. Prog = Programmable. Power switch is programmable based on specific application use-case

Chapter 13
Clock Overview

13.1 Overview

The chip supports domain-specific clock controllers (CLKCTL) to generate the peripheral clock sources. The domain-specific clocking approach helps reduce power needs through the following methods:

- Better clock control by utilizing divided clock sources to each power domain
- Local clock generation logic (muxing, dividers, gates) for each power domain
- Local peripheral clock and reset control for each power domain

13.1.1 System diagram

The chip's clock overview diagram is shown below:

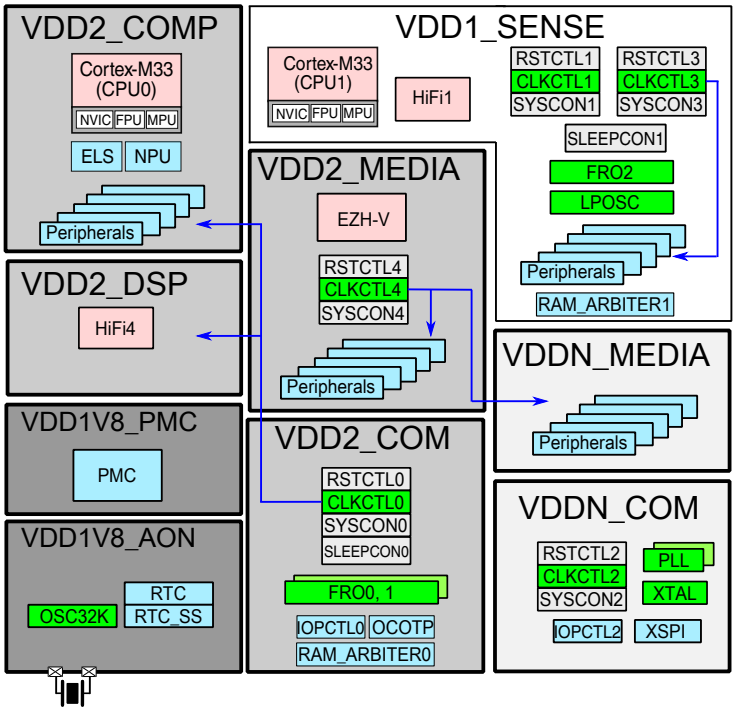


Figure 51. Clock overview

13.1.2 Clocking modules and components

The clock generation and distribution are implemented by the following blocks:

Table 170. Clocking components

IP module	Description
Clock controller (CLKCTL)	The CLKCTL module provides control for clock generation, division, distribution, and coarse-level gating. See Clock Controller (CLKCTL) for information on the CLKCTL architecture, functional description and programming model.
FRO Clock Generator(FRO)	The FRO is the free-running oscillators. The FROs are used to work as the primary clocking sources for the Compute and Sense Domains.
Nano Power 32 kHz Oscillator (OSC32KNP)	The OSC32KNP interfaces the external 32 kHz crystal oscillator.

13.2 Clocking considerations

There are 2 processors in the Compute (CPU0) and Sense (CPU1) domains respectively which follow semi-autonomous execution environments. During boot, the CPU0 boots first. Subsequently the CPU1 comes up, if CPU0 boots it. Overall, the architecture supports multiple schemes of handling shared resources – however in the context of clocking control in active mode inter-CPU messaging mechanism is used to ensure only one execution entity has control of the resource at any one point of time. The inter-CPU mailbox mechanism has the following features:

- Provide means for Inter-processor communication allowing multiple CPUs to share resources and communicate with each other in a simple manner
- Each CPU can claim a shared resource if it is available
- Provides a mutual exclusion configuration for the communication handshake

13.3 Clock sources

Table 171. Clock sources

Clock source	Description	Source at boot?
FRO	FROs are tunable, and are used to work as the primary clocking sources for the Compute and Sense Domains. FRO0 and FRO2 support configurable frequencies from 150 MHz to 325 MHz. FRO1 supports a fixed frequency of 192 MHz, and is used for bootup.	FRO0 - no FRO1 - yes FRO2 - no
Main PLL (528 MHz)	The main PLL is used when high frequencies (> 300 MHz) are required, as well as when strict clocking specifications such as jitter and duty cycle are needed. The PLL sources the main-compute domain and has individual outputs for the HiFi4. Input frequencies supported are 16, 19.2, 24, 26, 30 or 32 MHz	no
Audio PLL (528 MHz)	The audio PLL is used for audio purposes with strict jitter requirements. The clock inputs and acceptable input frequencies are the same as with the main PLL.	no
External clock (clkin)	The external clock brings in the possibility to bring in an external clock source	yes
XTAL oscillator (osc_out)	The Xtal oscillator brings in the possibility to bring in an external clock source.	yes

Table continues on the next page...

Table 171. Clock sources (continued)

Clock source	Description	Source at boot?
1 MHz Low-Power Oscillator (1m_lposc)	One of the main uses of the 1 MHz low power oscillator is to boot up the chip. The output of the 1 MHz oscillator 1m_lposc also goes through a divider to generate a 32 kHz low frequency clock source which is widely used throughout chip. The sense domain is alive in the active mode of the chip – therefore the wake32k_clk can be used for the SIO, USB etc.	yes
Nano Power 32 kHz Oscillator (for RTC)	Supports RTC clocking by supplying the 32 kHz source clock. The 32 kHz crystal oscillator is external. The 32 kHz clock may not be available for Cold Boot, but it will be available for low-power wakeup and reset	yes

See Clock Controller (CLKCTL) for more information.

13.4 Generated clock sources

In addition to the clock sources described in Clock sources section, there are also generated clock sources.

The clock sources and generated clock sources are inputs to the clock slice multiplexers. Clock source selection for generated clock sources and clock slice multiplexers are all done in the CLKCTL registers.

13.5 Clock root generation

Clock generation in the CLKCTL creates clock roots for on-chip peripherals. Each clock root has a dedicated clock slice. Each clock slice takes the clock source input and generates the clock root with the required frequency.

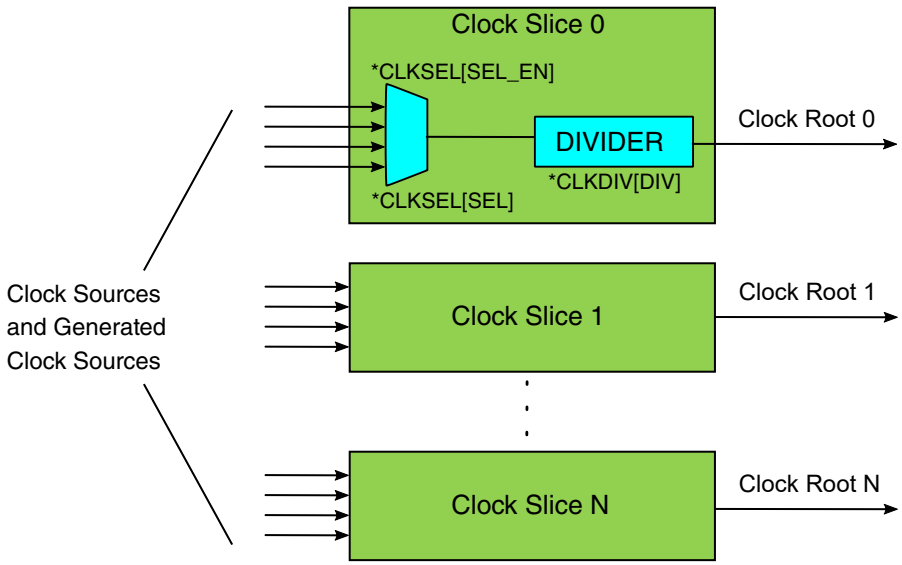


Figure 52. Clock root generation from clock slice

Clock source selection is in the CLKCTL functional clock select registers (SEL bitfield). Downstream from the clock source mux is the clock slice divider, which is controlled by the divider registers (DIV bitfield).

Each clock slice has a gate on the output of the functional clock select mux. The enable for these gates are also in the functional clock select registers (SEL_EN bitfield).

See Clock Controller (CLKCTL) for more information about clock generation and the clocking block diagrams.

13.6 Clock Gating

There are two levels of clock gating supported on the chip. An entire clock slice can be gated and all the downstream peripheral clocks connected to the slice by configuring the *CLKSRC[SEL_EN] field. This should only be done when all related peripherals connected to the clock slice are not in use. Peripheral clock gating is also supported. The peripheral clock gates are a distributed network of clock enables configured by the CLKCTL PSCCTL registers. For more information, please see the Clock Controller (CLKCTL) chapter.

Chapter 14
Graphics Overview

14.1 Overview

The graphics and display components on the chip are coupled together, and interface with on-chip memories (RAM_ARBITER) and external memory interfaces (XSPI) for efficient low-power performance.

14.1.1 Block diagram

The system diagram for the graphics, display, and camera modules is shown below:

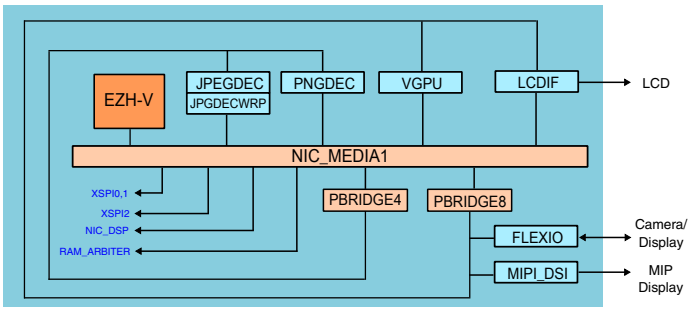


Figure 53. System diagram

14.1.2 Modules

The following table describes the graphics related modules in the chip.

Modules	Instances	Description	Main Features
EZH-V	1	The EZH-V is primarily implemented to provide a smart DMA engine for post-processing graphics data assembled by either or both the GPU and CPU, and then passing it to the FLEXIO for output. Processing operations may include data packing and endianness adjustment. The EZH-V is available for any other general-purpose handling that may be required and supports an extensive selection of trigger inputs from GPIOs and most on-chip peripherals.	• EZH-V Event / Algorithm Handler with 32 KB EZH code RAM and 64-bit I/D TCM.
VGPU	1	The VGPU provides a high-performance UI-based	• Open VG 1.1 • Linear paint

Table continues on the next page...

Table continued from the previous page...

Modules	Instances	Description	Main Features
		graphic core to enable hardware acceleration of the OpenVG API and display vector graphics. The DECNano (Decoder/ Encoder Core) series pixel compression and decompression fabric provides a unified lossy pixel compression and decompression infrastructure and support for pixel processing pipelines.	• Pixel input format • YUY2 input • INDEX 1/2/4/8 • DECNano compression / decompression • ETC2 decompression • Transformation support
JPEGDEC	1	High-performance JPEG decoder for still image and video decompression applications. JPEGDEC is a submodule of JPGDECWRAP.	• Decoding standard-compliant baseline and extended JPEG streams • Decompress video payload of several standard motion JPEG container formats.
JPGDECWRP	1	JPGDECWRP is a high-performance JPEG decoder for still image and video decompression applications. Contains DMA channels for bit stream fetching and decoded pixel data storing, as well as a register interface and chained descriptors for frame or bit stream buffer switching. There are two submodules, JPEG decoder wrapper and JPEG decoder core.	• Baseline and extended ISO/IEC 10918-1 JPEG compliance • Conversion of the pixel MBT format to linear format • Multi-bit-stream context switch based on frame • YUV444, YUV420, YUV422, RGB, ARGB, and Gray formats are supported • 8-bit and 12-bit pixels precision
PNGDEC	1	PNGDEC decodes pictures in the PNG source format. PNGDEC includes a DMA engine	• 64-bit DMA engines used to fetch PNG bitstreams from the system memory and feed them into PNGDEC, and store decoded RGB-A raw pixel data and ancillary data • Huffman encoded data (both static and dynamic) • Gray scale, RGB (8- or 16-bit depth) and indexed color (8-bit depth) with and without alpha channel
LCDIF	1	LCDIF is a high-performance, low-power display processor unit (DPU). It outputs video and graphic streams from a single output panel. The single output panel can	• Provides DPI or DBI for display output. • LCDIF uses the DBI interface by default. The supported DBI types include: - Type A (Fixed E mode or Clocked E mode) - Type B

Table continues on the next page...

Table continued from the previous page...

Modules	Instances	Description	Main Features
		connect to either a display pixel interface (DPI) device or a display bus interface (DBI) device to support RGB output.	• Provides post-processing computing functions, including: - Alpha blending - Gamma - Dither
MIPI_DSI	1	MIPI DSI controller interface for serial graphics display output that implements all protocol functions defined in the MIPI_DSI_HOST Specification. The MIPI_DSI_HOST controller provides an interface that allows communication with MIPI_DSI_HOST-compliant peripherals.	• Implements all three DSI Layers (Pixel to Byte packing, Low Level Protocol, Lane Management) • Support for Command and Video Modes • Supports virtual channels (dual display) • Host Version • Supports PHY Protocol: - 1 Clock lane - 2 data lanes - Supports MIPI Standard for D-PHY - High Speed Serializers and Deserializers - MIPI Alliance Specification for Display Serial Interface Version 1.1 compliant • Support for all three Video Mode packet sequences — Non-Burst Mode with Sync Pulses — Non-Burst Mode with Sync Events — Burst mode
NIC_MEDIA1	1	Bus interconnect for the graphics and media modules. AXI_SWITCH is an Arm CoreLink Network Interconnect component (NIC) that switches any initiator access to the target memory or peripheral. It has multiple target interfaces and initiator interfaces. It supports Arm CoreLink QoS-400 Network Interconnect Advanced Quality of Service (QoS) to arbitrate and regulate the transactions from various SoC initiators.	• Programmable maximum for outstanding transactions to include: — Separate maximum for read and write requests. — Combined maximum for all requests. — Fractional value to provide finer control. • Regulation of read and write request issuing rates to meet programmed traffic
XSPI	3	XSPI provides an interface to a single external SFM device.	• Programmable sequence engine for memory device commands and operations

Table continues on the next page...

Table continued from the previous page...

Modules	Instances	Description	Main Features
		This device can have up to 16 bidirectional data lines. Three XSPI Flash Interface modules support Octal and Quad SPI memory devices. Two of them (XSPI0 and XSPI1) are supported with a policy selector (CACHE64_POLSEL). The first XSPI instance (XSPI0) is primarily for code execution from off-chip SPI flash memory. It supports execute-in-place (XIP) and on-the-fly (OTF) decryption using the PRINCE module. It also provides a mechanism to shift a designated range of addresses to a different region of off-chip memory to support dual-image boot. The second XSPI instance (XSPI1) is primarily to access data from SDRAM (particularly for graphics). The XSPI1 is accessible by the HiFi4 DSP as well as the CPU0. Both XSPI0 and XSPI1 Interfaces will include a 32 KB cache with an CACHE64 cache controller.	• Single, dual, and quad modes of operation supported for Quad flash memories • Octal and single IO operation supported for Octal flash memories • Double Data Rate (DDR)/Double Transfer Rate (DTR) mode that generates data on every edge of the memory clock • Flash memory data strobe signal to support data sampling in DDR and Single Data Rate (SDR) mode • HyperRAM support • Direct memory mapping of all AHB memory accesses to the chip system memory space. • SNAND device support for NOR-compatible command through the IPS interface • X16 operation for PSRAM with EIGHT PAD support (similar to X8 mode). • Programmable bit skewing adjustment for dual DQS.
RAM_ARBITER	2	Internal SRAM interface	• 7.5 MB of system SRAM accessible by the VDD2_COMP and • VDD1_SENSE processing domain, DMA engines, and bus initiators
FLEXIO	1	Flexible IO controller can emulate a variety of serial and parallel communication protocol. FlexIO can be configured to provide a parallel interface to an LCD.	• Flexible 16-bit timers with support for a variety of trigger, reset, enable, and disable conditions.

14.2 Functional description

14.2.1 Application information

The chip has devoted pads to driving a memory-in-pixel (MIP) display. This typical application, supports 2-bit color, 6 data lines, horizontal sync and vertical sync. In addition, the chip can provide the panel back plane signals that are driven from the 32 kHz RTC clock. See NXP site for additional graphics support resources, including support from the SDK and GUI guider.

Applications

- Battery powered devices
- Multiple display options (ex. OLED 24-bit color display)
- Augmented reality glasses

Primary supported use cases - 500x500 display size

- 24-bit color graphics from internal and external memory using JPEG and PNG images at 70 fps.
- 888 color depth supporting full motion JPEG at 30 fps
- LCDIF supporting DPI and DBI mode, fetch decoded data from SRAM
- LCDIF display in VG layer
- LCDIF fetch vGPU render data from SRAM and display in overlayer0
- Use XPSI2 (PSRAM), XSPI0 (Flash), and RAM_ARBITER (SRAM) memories

14.2.2 MIPI display interface

The chip supports a MIPI panel display through the MIPI DSI interface. The MIPI DSI interfaces the LCDIF to provide packet support to the DPI and DBI interfaces. The DSI then interfaces the DPHY, which connects to the panel display pins.

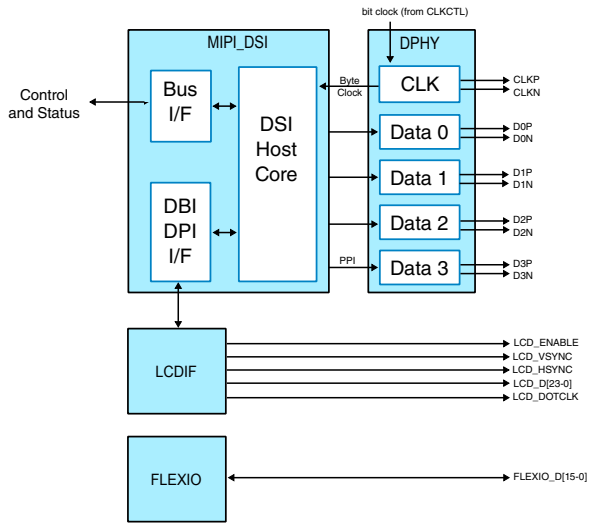


Figure 54. Display interface

14.2.3 FLEXIO interface

The FLEXIO can support a camera and display interface. The camera interface can couple with the DSI for an integrated camera and display solution. See the FLEXIO chapter for more information, including the pinmux sections for the associated pinout details.

- FLEXIO_D[7-0] – Samples data 0 to data 7
- FLEXIO_D8 – PCLK as internal shift clock, timer 0 decrements on PCLK
- FLEXIO_D9 – HREF as trigger to enable 16-bit counter (timer 0)

Chapter 15
Timers Overview

15.1 Introduction

The timer modules provide timers to support many different functions throughout the device. This chapter does not cover timers that are exclusive to particular domains and have no user interface.

15.1.1 Block diagram

The block diagram for the system level timer architecture is provided below.

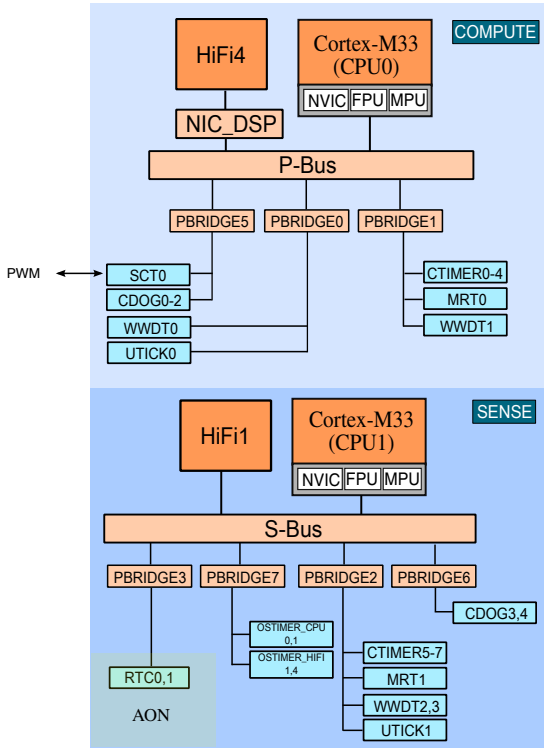


Figure 55. System timer block diagram

15.1.2 Modules

This chip includes the following timer modules.

Table 172. Timer modules

Modules	Instances	Description	Main feature
Real Time Clock (RTC)	RTC0,1 (HP) - Sense (VDD1) RTC0,1 (LP)- AON (VDD1V8)	RTC provides time keeping and calendaring functions. It also provides protection against external or internal tampering, protection against spurious memory or register updates, and battery operation.	• Basic clock functions — Separate counters for days, hour, minutes, and seconds

Table continues on the next page...

Table 172. Timer modules (continued)

Modules	Instances	Description	Main feature
			— Calendaring support—Separate counters for year, month, and day of the week ◦ Automatic adjustment for daylight saving with user-defined parameters ◦ Automatic adjustment for month and leap year — External clock support • Time zone offset • Programmable alarm with interrupt • Periodic interrupts • Hardware compensation to frequency and temperature changes For detailed information, see Real Time Clock (RTC).
Micro-Tick Timer (UTICK)	UTICK0 - Compute (VDD2) UTICK1 - Sense (VDD1)	UTICK is a 31-bit timer that counts down to 0 and then generates an interrupt. Therefore, it provides a fixed time interval between interrupts. It is a simple, ultra-low-power timer that can run and wake up the chip in reduced-power mode, including sleep and deep sleep modes.	• Wakes up the chip from Low-power mode. • Operates using interrupt or software polling mode. • Includes Capture registers that use external pin transitions as a trigger. For detailed information, see Micro-Tick Timer (UTICK).
Standard Counter and Timers (CTIMER)	CTIMER0 - Compute (VDD2) CTIMER1 - Compute (VDD2) CTIMER2 - Compute (VDD2) CTIMER3 - Compute (VDD2) CTIMER4 - Compute (VDD2) CTIMER5 - Sense (VDD1) CTIMER6 - Sense (VDD1)	CTIMER contains multiple timers each of which can generate interrupts or perform other actions at specified timer values a configured in four match registers. Each timer can also capture the timer value when an input signal transitions and optionally generate an interrupt at that transition. Using the match registers, you can configure a timer to output PWM pulses.	• Contains a 32-bit CTIMER with a programmable 32-bit prescaler • Contains Timer Control (TCR) and External trigger enable • Performs counter or timer operations • Selects the function clocks that may be asynchronous to other system clocks • Includes up to four 32-bit captures For detailed information, see Standard/Counter Timers (CTIMER).

Table continues on the next page...

Table 172. Timer modules (continued)

Modules	Instances	Description	Main feature
	CTIMER7 - Sense (VDD1)		
OS Event Timer (OSTIMER)	OSTIMER (OSTIMER_CPU0,1 and OSTIMER_HiFi1,4) - Sense (VDD1)	OSTIMER includes a shared timer and separate match and capture functions for CPU0, CPU1, HiFi1 and HiFi4.	• Provides a 64-bit Gray code counter • Supports separate functions for each CPU: — A capture register can copy the main counter value when triggered by a CPU request. — A match register can be compared to the main counter and can optionally generate an interrupt or wake-up event. For detailed information, see OS Event Timer (OSTIMER).
SCTimer (SCT)	SCT0 - Compute (VDD2)	SCT is a flexible timer module capable of creating complex PWM waveforms and performing advanced timing and control operations with minimal or no CPU intervention. It can operate as a single 32-bit counter or as two independent, 16-bit counters in Unidirectional or Bidirectional modes.	• Configurable as two 16-bit counters or one 32-bit counter • Configurable as up counters or up-down counters • Configurable number of match and capture registers • Ability to create time-proportioned PWM signals • Up to eight single-edge or four dual-edge PWM outputs For detailed information, see SCTimer (SCT).
Multi-Rate Timer (MRT)	MRT0 - Compute (VDD2) MRT1 - Sense (VDD1)	MRT provides a repetitive interrupt timer with 4 channels. MRT provides a repetitive interrupt timer with 4 channels.	• 24-bit interrupt timer. • 4 channels independently counting down from individually set values. • Repeat Interrupt, One-Shot Interrupt, and One-Shot Stall modes. • Programmable individual time intervals for each channel. For detailed information, see Multi-Rate Timer (MRT).
Windowed Watchdog Timer (WWDT)	WWDT0 - Compute (VDD2) WWDT1 - Compute (VDD2) WWDT2 - Sense (VDD1)	WWDT helps you reset or interrupt a microcontroller or core that is stuck in an infinite loop or is executing unintended code. If an application fails to reload or feed a watchdog (WDOG) timer within the predefined duration of time (the window), WWDT can generate a watchdog reset.	• Can reset or interrupt the chip within a programmable time • Watchdog reset indication flag • Clocking features: — Programmable 24-bit timer with an internal fixed (divided by 4) prescaler

Table continues on the next page...

Table 172. Timer modules (continued)

Modules	Instances	Description	Main feature
	WWDT3 - Sense (VDD1)		— Selectable time period in increments of four watchdog clocks — Watchdog clock (WDCLK) source having a selectable frequency • Can be configured to run in Low-Power mode For detailed information, see Windowed Watchdog Timer (WWDT).
Code Watchdog Timer (CDOG)	CDOG0 - Compute (VDD2) CDOG1 - Compute (VDD2) CDOG2 - Sense (VDD1) CDOG3 - Sense (VDD1) CDOG4 - Sense (VDD1)	CDOG provides mechanisms for detecting side-channel attacks and the execution of unexpected instruction sequences	• Secure Counter (SEC_CNT) detects altered software in the execution flow. • Instruction Timer (INST_TIMER) places a hard upper-limit on the interval between checks of the secure counter. For detailed information, see the I.MX RT700 Security Reference Manual.

Chapter 16
System Debug

16.1 Overview

It is necessary to be able to debug SoC and to ensure that access is properly controlled. Within the chip there are multiple objects which need to be debugged and an overall infrastructure to enable that to occur. The chip debug architecture is based on Arm CoreSight technology, with support for the Cortex-M33 cores.

16.1.1 Block diagram

The block diagram for the system level debug architecture is provided below.

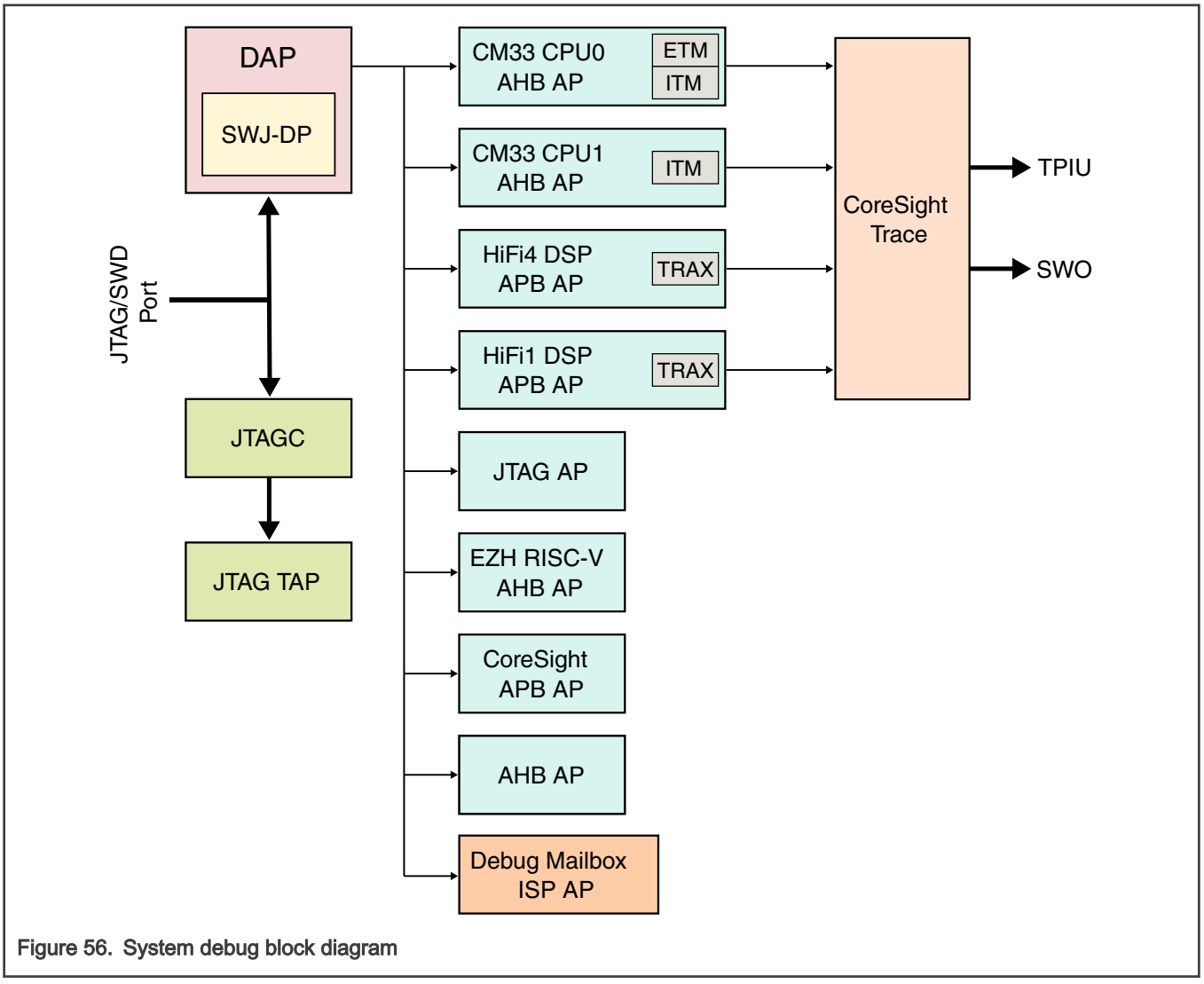


Figure 56. System debug block diagram

16.2 Functional description

16.2.1 Features

The debug infrastructure provides support for debug and trace of core and trace of chip modules. Some of the features that debug infrastructure provides are:

- Debug access control
- Debug Access Port (DAP)
- Run control
- CPU and interconnect traces
- Trace data export to system memory and off-chip
- Global trace Timestamp
- Embedded Cross Triggering (ECT)

The key features of the debug system include:

- Support the following cores and IP
  - 2x Cortex-M33 cores
  - HiFi4 DSP core
  - HiFi1 DSP core
  - EZH RISC-V core
- Support 1149.1 (JTAG) 5-pin interface
- Support Arm Serial Wire Debug (SWD) 2-pin interfaces. Through sequence recognized by the Arm DAP IP, the JTAG signals can change to SWD mode
- Support both non-intrusive trace and halt-mode debug options
- Support the combination of trace streams from cores and modules that support trace
- Support capture of trace using the Arm Trace Memory Controller (TMC). Trace can be captured into:
  - A 16 KB Embedded Trace FIFO (ETF) in chip trace block
  - System memory using the Arm Embedded Trace Router (ETR)
  - CoreSight Serial Wire Output (SWO)
- The SWJ-DP interprets the data coming in and routes to appropriate Access Port (AP).
- Timestamp distribution and the closely related system timer distribution in the chip

16.2.2 Debug Access Port (DAP)

DAP is a standard Arm component, comprising of several components. These components are used to access the DAP from an external debugger, and Access Ports to access on-chip debug system resources. The DAP supports 1149.1/Arm SW-DP interface, which means that the JTAG interface can be operated in either standard 5-pin JTAG-DP interface or in 2-pin SW-DP interface.

Serial Wire port (SWJ-DP) consists of a wrapper around the JTAG-DP and SW-DP. It selects JTAG or SWD as the connection mechanism and enables either JTAG-DP or SW-DP as the interface to the various on-chip debug/trace components.

16.2.3 AP Select Decoding

The recommendation for AP selects decoding is provided in the following table.

The selection of different access ports in the DAP is done based on the APSEL value set in the SELECT register of SWJ-DP. APSEL is 31:24 bit field of the DAP SELECT register. The APSEL will be decoded as given in the following table.

Table 173. Address Decoding for DAPBUSIC

Destination	DAPBUSIC_AP_SEL[31:24]
AHB-AP to CPU0 M33	00h (M0)
AHB-AP to CPU1 M33	01h (M1)
Debug Mailbox ISP-AP	02h (M2)
APB-AP0 to Advanced Peripheral Bus Interconnect (APBIC)	03h (M3) Refer to Decoding for APBIC
AHB-AP to system bus	04h (M4)
Reserved (Default AP response)	05h - FFh

Table 174. Decoding for APBIC

Destination	Identification Register (IDR)
HiFi4 DSP	5477_0002h
HiFi1 DSP	5477_0002h
EZH RISC-V Core	5477_0002h

From the interface of the debugger, serially feeding the “address” into a TAP controller via TDI, and once acknowledged, the associated data register is either streamed out on TDO or into on TDI. See the Arm Debug Interface Architecture Specification ADIv5.2 for more information about selecting and accessing Access Ports.

16.2.4 JTAGC and TLM

A TAP Linking Module (TLM) module is used to select the different JTAG TAPs inside the device.

- Port 0: JTAG (default port)
- Port 1: USB2.0 PHY

The selection between JTAG and SWD is performed by the SWJ-DP in the DAP. This is done by recognizing a sequence on the JTAG pins and therefore no additional switching of the JTAG scan chain is needed to support this Arm feature.

The JTAG command length is 4-bits. All JTAG TAPs operating in unison have the same Instruction Register (IR) width. This includes the Arm DAP’s TAP.

JTAGC and SWJ-DP share the same 5-pin JTAG interface from the PAD I/O pad of chip.

The JTAGC TAP follows the DAP TAP. If a the JTAGC TAP leaves the TAP Link Reset (TLR) state, then it is known that some activity is being performed on the JTAG port.

NOTE

IEEE 1149.7 interface is not supported.

16.2.4.1 JTAG IR Commands

Selection is based on the following table of decodes.

Table 175. Instruction Register

Instruction	Value	Module
EXTEST	0000b	JTAGC
SAMPLE	0001b	JTAGC
PRELOAD/SAMPLE	0001b	JTAGC
HIGHZ	0010b	JTAGC
CLAMP	0011b	JTAGC
IDCODE (JTAGC)	0100b	JTAGC
TEST_CTRL_REG	0111b	JTAGC
ABORT	1000b	Arm
DPACC	1010b	Arm
APACC	1011b	Arm
TEST_LEAKAGE	1100b	JTAGC
IDCODE (Arm)	1110b	Arm
BYPASS	1111b	JTAGC

16.2.4.2 JTAG and SWD

A special sequence on the JTAG state machine changes the DAP operation from the JTAG to SWD. This sequence is specified by Arm Ltd and fully recognized within the SWJ-DP's JTAG-DP.

SW-DP pins (TCK,TMS) are connected directly from and to chip pins.

When SW-DP is enabled, both JTAGC and JTAG-DP are disabled.

When SW-DP is disabled, the output to TDO from JTAGC and JTAG-DP are selected by a mux that is controlled by decode of the IR inside the JTAGC, and then connected to chip pins for debug and/or test purposes.

NOTE

By default, PIO0_0 must be externally pulled high to enable debug over SWD pins.

16.2.5 Debug over USB pins

This chip supports to debug over USB0 pins.

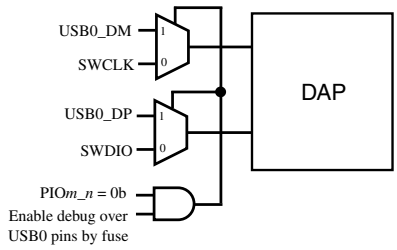


Figure 57. Debug over USB0 pins

In order to debug over USB0 pins, you need to:

1. Configure fuse to support debug over USB0 pins:

- Write 0b to BOOT_CFG8[DBG_OVER_USB_DIS] to enable debug over USB0 pin feature.
- Assign the port pin PIO*m_n* that will be used to enable debug over USB0 pins by BOOT_CFG8[DBG_OVER_USB_SEL_PORT] (*m*) and BOOT_CFG8[DBG_OVER_USB_SEL_GPIO] (*n*)

NOTE

PIO*m_n* must be an existent pin. PIO0_0 is the default pin to select debug over SWD or USB0. By default, PIO0_0 must be externally pulled high or low to enable debug over SWD or USB0. You can assign another existent port pin besides PIO0_0 to enable debug over SWD or USB0,

2. Configure the pin PIO*m_n* = 0b (low) to enable debug over USB0 pins.

NOTE

If BOOT_CFG8[DBG_OVER_USB_DIS] = 1b or PIO*m_n* = 1b (high), SWCLK and SWDIO are used as debug pins.

16.2.6 Trace

16.2.6.1 Trace Export

16.2.6.1.1 Embedded Trace Router (ETR)

The Embedded Trace Router converts ATB to AXI protocol so that trace data can be stored in chip addressable memory.

16.2.6.1.2 Arm CoreSight Trace Port interface (TPIU)

The TPIU is one of the targets for forwarded trace data from the Instruction Trace Macrocell (ITM). The TPIU is used to transfer trace information to the physical trace port of the device.

16.2.6.1.3 Serial Wire Output (SWO)

The device supports a serial wire output mechanism for trace. The output is muxed onto the TDO pin.

16.2.6.2 Trace Data Bus

The AMBA Trace Bus (ATB) is used to transport trace data from the trace source to the system memory or off-chip. ATB traces are routed in the chip through CoreSight SoC-400 ATB Funnel and ATB replicators.

16.2.7 Cross Triggering

Cross-triggering allows a condition recognized by one debug or trace block to be reacted to in another debug or trace block. Examples of this include multi-core halting, or having a core halt due to chip condition.

Cross-triggering is required between all major subsystems of the chip (those which have a core-like structure, or trace function STM, ETM, etc.) for debug purposes.

NOTE

SoC-400 CoreSight cross-triggering components are used, which require provided Cross Trigger Interface/Matrix (CTI/CTM) adapters to interface with compliant Arm cores and debug components.

The debug enable on the CTM and CTI blocks is connected to a chip wide debug enable. The effect is that selectively controlling cross-trigger only between two components is not supported. It is enabled or not-enabled globally.

The Cross-Triggering matrix include connections to:

- Cortex-M33 CPU cores
- HiFi4
- HiFi1

- EZH RISC-V
- ETR
- ETF
- Trigger connections to system interrupts

16.2.7.1 Embedded Cross Trigger (ECT)

Embedded Cross Trigger (ECT) comprises a set of CoreSight Cross Trigger Interface (CTI) and Cross Trigger Matrix (CTM) components connected together. The main function of the ECT is to pass debug events from one core to another. For example, the ECT can communicate debug state information from one core to another, so that program execution on both cores can be stopped at the same time if required.

The diagram below shows the structure of the ECT.

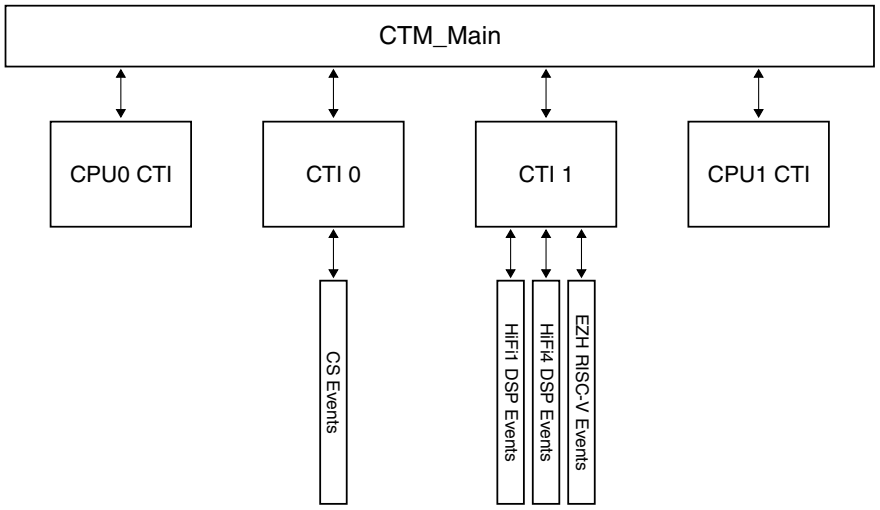


Figure 58. Embedded Cross Trigger Diagram

16.2.8 Timestamp

Timestamping on the device is performed with the use of several CoreSight components.

The Arm CoreSight Timestamp Generator is implemented in the system to provide a consistent view of time for the multiple blocks in the device. The 64-bit counter works in the Compute CPU0 CM33 clock domain and is the source of the time used by the following components:

- Cortex-M33 CPU0
- Cortex-M33 CPU1
- HiFi4
- HiFi1

16.2.9 JTAG IDCODE

The JTAG_IDCODE (JTAG ID) is a 32-bit identifier that identifies the chip silicon for the purpose of both debug and boundary scan access.

Table 176. JTAG_IDCODE

Device	JTAG_IDCODE
I.MXRT700	288C_C01Dh

Chapter 17
Audio Overview

17.1 Audio overview

The audio subsystem consists of an 8-channel MICFIL, which supports PDM input, hardware based voice activation detection (HWVAD), and decimation functions.. Some decimator outputs can be streamed to SAI module through a DMA operation. The subsystem also includes a Sigma-delta ADC (SDADC), which supports both line input mode and analog microphone input mode.

17.1.1 Block diagram

The audio block diagram is shown below:

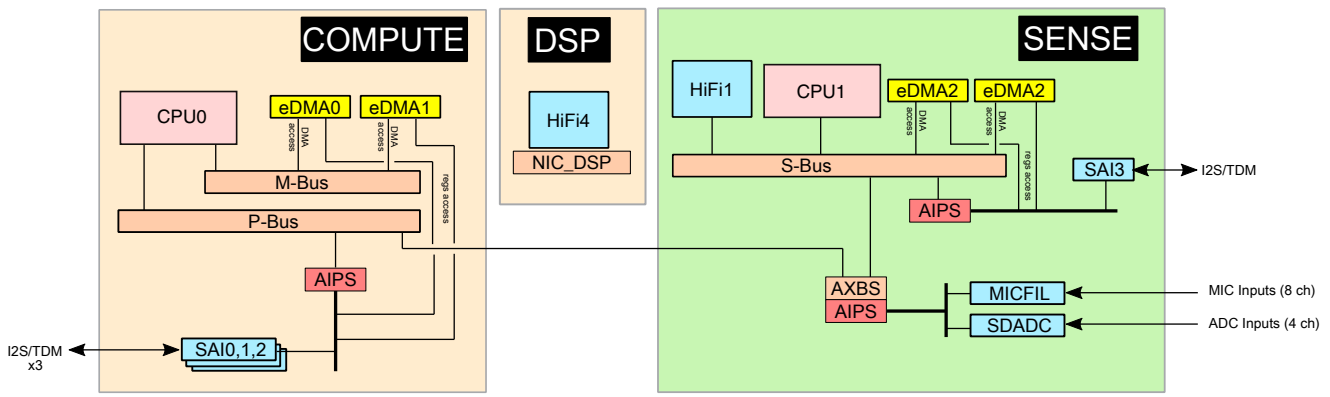


Figure 59. Audio block diagram

17.1.2 Modules

The following audio peripherals are part of this subsystem.

Table 177. Audio Modules

Modules	Instances	Description	Main Features
PDM Microphone Interface (MICFIL)	MICFIL	The PDM Microphone Interface (MICFIL) is used to receive audio from microphone. It supports 8-microphones (4 lanes) input and support for frame synchronization feature and Acoustic Overload Point (AOP) microphones for high quality audio applications. The MICFIL can support an ultrasonic channel on the same MICFIL input. There is a separate high frequency band pass filter that will route the ultrasonic channel data to a separate buffer.	• Decimation filters • FIFOs with interrupt and DMA capability • Hardware Voice Activity Detection (HWVAD) • Programmable PDM clock generator • MICFIL is in the Sense domain

Table continues on the next page...

Table 177. Audio Modules (continued)

Modules	Instances	Description	Main Features
Synchronous Audio Interface (SAI)	SAI0 SAI1 SAI2 SAI3	The SAI are synchronous serial interfaces used to transfer audio data.	• The microphone channel pair 0, 1 can be streamed to SAI0 through a DMA operation. • Transmitter and receiver with independent Bit Clock and Frame Sync • Asynchronous FIFO for each Transmit and Receive data line • SAI0 - SAI2 are in the Compute domain • SAI3 is in the Sense domain
Sigma Delta Analog-to-Digital Converter (SDADC)	SDADC	This ADC is a high dynamic range audio Analog-to-Digital Converter. It supports both line input mode and microphone input mode. It provides low latency decimation to enable Acoustic Noise Cancellation and Acoustic Echo Cancellation. It has a high input impedance to reduce its loading to input sources and to improve its robustness to interference. The DAC inside the SDADC can be used as a stand-alone DAC to provide a low-impedance voltage output.	• Low noise, high linearity ADC • Support single-ended and differential inputs • Support analog microphones: MEMS and JFET • SDADC is in the Sense domain
HiFi 1 DSP	HiFi1	This DSP resides in the Sense domain and is primarily intended for Always-On Applications, Bluetooth LE audio, and Sensor fusion.	• 2-slot machine with up to 64-bit load or storage bandwidth • Two-way SIMD ALU or MAC operations • Supports quad 16 x 16, dual 32 x 16-bit MACs, and single 32-bit MACs • HiFi1 is in the Sense domain
HiFi 4 DSP	HiFi4	This DSP is intended for efficient execution of audio and voice codecs, and pre- and post-processing modules.	• Support for four 32 x 32-bit MACs • Limited support for 72-bit accumulators • Support eight 32 x 16-bit MACs • Ability to issue two 64-bit loads per cycle. • Floating point unit included • HiFi4 is in the DSP domain
Enhanced Direct Memory Access (eDMA)	eDMA0 eDMA1 eDMA2 eDMA3	The eDMAs perform complex data transfers with minimal processor intervention. See Audio modules access to CPUs, DSPs and DMAs for specific information regarding audio module access to the eDMAs.	• eDMA0 and eDMA1 are 16-channel eDMAs. These reside in the Compute domain • eDMA2 and eDMA3 are 8-channel eDMAs. These reside in the Sense domain

17.2 Functional description

17.2.1 Audio modules access to CPUs, DSPs and DMAs

The following table shows which CPUs, DSPs and DMAs can access the audio modules.

Table 178. Audio modules access to CPUs, DSPs and DMAs

Module	CPU0 access?	HIFI4 access?	CPU1 access?	HIFI1 access?	eDMA0/ eDMA1 access?	eDMA2/ eDMA3 access?
MICFIL	yes	yes	yes	yes	yes	yes
SAI0-2	yes	yes	no	no	yes	no
SAI3	no	no	yes	yes	no	yes
SDADC	yes	yes	yes	yes	yes	yes

17.2.2 SAI module clocking

The following figure shows a detailed SAI clock multiplexing scheme.

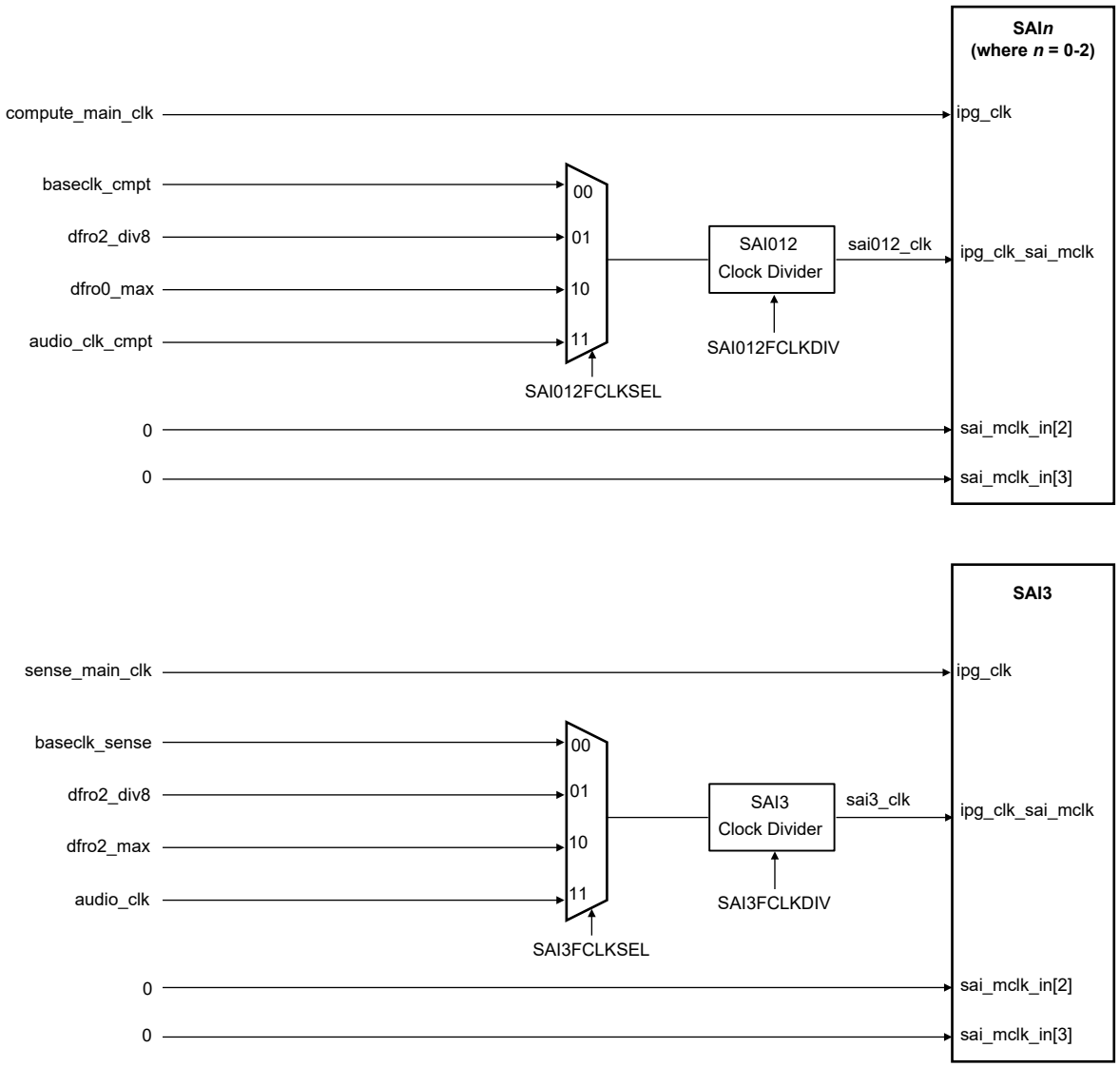


Figure 60. SAI clock multiplexing

NOTE

SAI012FCLKSEL, SAI012CLKDIV, SAI3FCLKSEL, and SAI3CLKDIV are CLKCTL registers.

The following figure shows muxing details for audio_clk and audio_clk_cmpt.

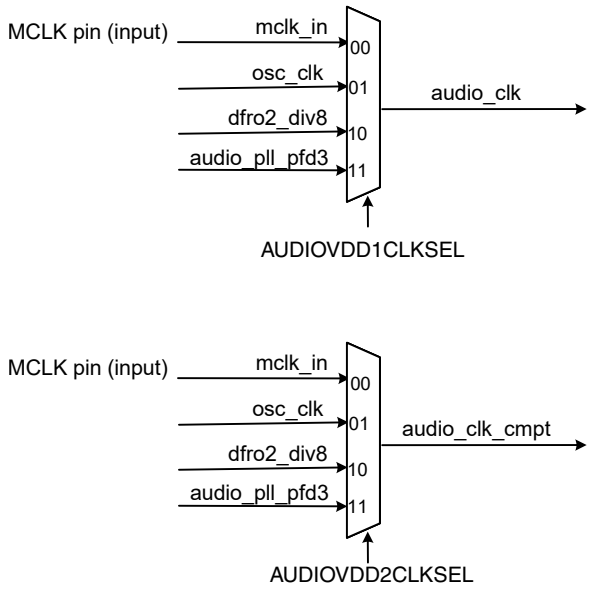


Figure 61. audio_clk and audio_clk_cmpt details

NOTE

AUDIOVDD1CLKSEL and AUDIOVDD2CLKSEL are CLKCTL registers.

See CLKCTL chapter for more clocking details. See SAI chapter for more SAI configuration information.

Chapter 18
Analog Overview

18.1 Analog Overview

This chapter covers the analog modules on this chip.

18.1.1 Block diagram

The analog block diagram is shown below:

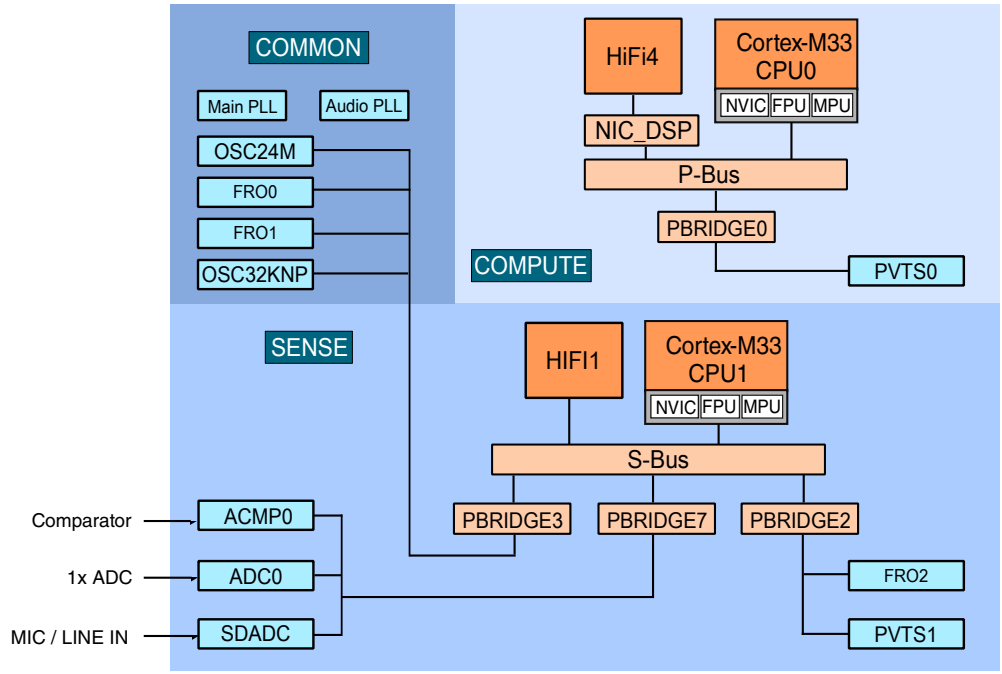


Figure 62. Analog block diagram

18.1.2 Modules

The following analog modules are part of this subsystem.

Table 179. Analog modules

Modules	Instances	Description	Main features
Analog-to-Digital Converter (ADC)	1	ADC is a dual successive approximation ADC designed for operation within an integrated microcontroller system-on-chip. ADC is in the VDD1_SENSE power domain.	• Linear successive approximation algorithm: differential operation with 16-bit or 13-bit resolution, single-ended operation with 16-bit or 12-bit resolution

Table continues on the next page...

Table 179. Analog modules (continued)

Modules	Instances	Description	Main features
			• One ADC supports for two simultaneous single ended conversions • Channel support for up to 64 analog input channels for conversion of external pins and from internal sources
Sigma Delta Analog-to-Digital Converter (SDADC)	1	SDADC is a low-power high-performance audio ADC system. It mainly targets battery- powered, hearable and wearable audio devices. It supports analog microphone and line inputs. Decimation filter is also part of this system. The decimation filter has low latency to enable acoustic noise and echo cancellation. SDADC is in the VDD1_SENSE power domain.	• Low noise, high linearity ADC • Low power modulator • Supports sampling frequencies of 16kHz, 32kHz, and 48kHz • Support single-ended and differential inputs • Support analog microphones: MEMS and JFET
Comparator (ACMP)	1	ACMP includes high-speed Comparator, DAC, and ANMUX modules. The high speed comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit operates across the full range of the supply voltage, known as rail-to-rail operation. The Analog MUX (ANMUX) module provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 8-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage. The DAC module is a 256-tap resistor ladder network that provides a selectable voltage	• CMP features: — Operational over the entire supply range — Rail to rail inputs — Programmable hysteresis control • DAC features: — 8-bit resolution — Selectable supply reference source — Configurable low power mode or high speed mode • ANMUX features: — Two 8-to-1 channel MUXes for selection of the source for comparator input

Table continues on the next page...

Table 179. Analog modules (continued)

Modules	Instances	Description	Main features
		reference for applications requiring a voltage reference. The 256-tap resistor ladder network divides the supply reference Vin into 256 voltage levels. A 8-bit digital register input selects the output voltage level, which varies from Vin to Vin/256. Vin can be selected from two voltage sources, Vin1 or Vin2. ACMP is in the VDD1_SENSE power domain.	— Operational over the entire supply range
Process, Voltage, and Temperature Sensor (PVTS)	2	PVTS delivers a digital output that indicates if junction temperature is above or below a given temperature threshold. PVTS0 is in the VDD2_COM power domain, and PVTS1 is in the VDD1_SENSE power domain.	• Programmable threshold value from -45 °C to 130 °C with better than 1.5 °C resolution • Configurable to continuously monitor a specific temperature threshold
Nano Power 32 kHz Oscillator (OSC32KNP)	1	OSC32KNP contains AIP and DIP. The AIP is an analog IP designed to provide 32.768 kHz clock to SoC with an external. The DIP is used to configure AIP and monitor the stability of clock. OSC32KNP is in the VDD1V8_AON power domain.	• Supports 32.768 kHz clock to SoC • Configurable clock accuracy • Capability to support clock in high power mode and low power mode
FRO Clock Generator (FRO_TUNER)	3	FRO_TUNER module provides a trimmable system clock. FRO_TUNER0 and FRO_TUNER1 are in the VDD2_COM power domain, and FRO_TUNER2 is in the VDD1_SENSE power domain. FRO_TUNER0 and FRO_TUNER1 are part of the memory slot of CLKCTL0, and FRO_TUNER2 is part of the memory slot of CLKCTL3.	• Trimmable FRO output • Programmable FRO post divider • Loss of trimmed lock Interrupt request

Table continues on the next page...

Table 179. Analog modules (continued)

Modules	Instances	Description	Main features
XTAL Oscillator (OSC24M)	1	OSC24M module provides the main crystal oscillator that can be used as an internal clock source. OSC24M is in the VDDN_COM power domain. OSC24M is part of the memory slot of CLKCTL2.	See CLKCTL section for information on OSC24M.
PLL	2	PLL contains a Main System PLL and an Audio PLL	See Clocking section for information on PLLs.

Chapter 19
Security Overview

19.1 Overview

This chapter provides an overview of the following chip security components, explaining the purpose and features of each of them.

More information can be found in the Security Reference Manual for this chip.

19.1.1 System diagram

The following shows a simplified security system diagram.

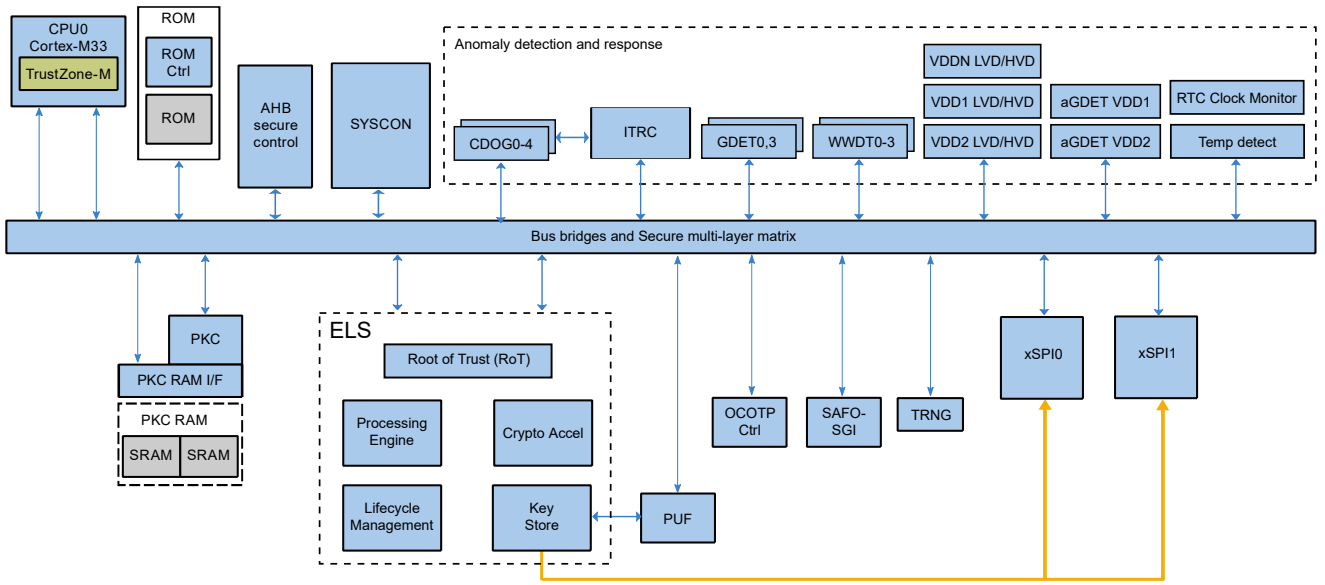


Figure 63. Security system diagram

NOTE

The High Voltage Detectors (HVD), Low Voltage Detectors (LVD), Analog Glitch Detectors (aGDET), and Temperature Detector are part of the Power Management Controller (PMC). See the PMC chapter for more information on these detectors.

19.1.2 Modules

The following modules are security components of the chip:

Table 180. Security subsystems and modules

Modules	Description
Secure Isolation components	
Arm TrustZone	Arm's TrustZone for Armv8-M architecture included as part of the Cortex-M33 complexes
Secure bus control	NXP's Secure AHB Bus Controller

Table continues on the next page...

Table 180. Security subsystems and modules (continued)

Modules	Description
Cryptographic Accelerators	
ELS	EdgeLock Secure Subsystem
PKC	Public Key Crypto Co-processor
SAFO_SGI	Symmetric Cryptographic Accelerator
Key Management	
ELS	Part of ELS provides this function
SRAM-PUF	Physical Unclonable Function to create device unique Root of Trust (RoT) key
Security Configuration Storage	
OCOTP Control	OTP controller to store security configuration
Anomaly Detection and Response	
aGDET	Two Analog glitch sensors - one for each security-sensitive voltage rail
GDET	Two Digital Glitch Detectors connected to voltage rails
ITRC	Intrusion and Tamper Response controller to configure response sequence to detected security events
LVD/HVD	Low-Voltage (LVD) and High-voltage (HVD) Detectors per voltage rail
RTC Clock Monitor	Clock frequency monitor for 32KHz RTC crystal oscillator and main oscillator
CDOG	Five Code Watchdog controllers (3x Compute domain, 2x Sense domain)
Boot ROM	
Boot ROM Security Features	• Secure boot providing ECDSA signature verification as per NIST P-256 and P-384 curves • Secure update using SB3.1 file format. Supports firmware update mechanism with authenticity (ECDSA P-256/P-384 signed) and confidentiality (AES-CBC encrypted and SHA256 chained) protection • Secure Debug: Certificate based debug authentication mechanism using ECDSA P-256 and P-384 keys • Secure Provisioning • Secure Identity solutions
Secure Storage	
ELS key store	Support for OTP-based device unique key as well as SRAM-PUF device unique key **Note:** ELS key store generated from SRAM-PUF based device unique key supported only when external non-volatile memory is available
IPED	Inline Prince Encryption and Decryption engine (IPED) for off-chip flash with option to select regular counter mode (CTR) or Galois Counter mode (GCM) per region for code storage and data

19.2 Functional description

19.2.1 Root of Trust (RoT)

The Root of Trust (RoT) consists of truly immutable hardware logic, including analog and digital logic, read-only memory and one-time programmable memory. Immutable RoT is essential for guaranteeing any security feature, including Secure Boot, Secure Debug, Life-cycle Management, and a number of others. In this device, Immutable RoT is embedded in the Boot ROM (immutable bootloader). It uses the device hardware cryptographic functionality for its function.

19.2.2 Life-cycle management

During its lifespan, a typical device finds itself in various places around the world. It is manufactured in a semiconductor factory, tested and packaged in silicon manufacturer facilities, sold to various distributors, sold further to the Original Equipment Manufacturer (OEM), assembled, tested and provisioned by their Contract Manufacturers and, finally, delivered to the end customer. In the case of failure, a device is returned to OEM or even back to the silicon manufacturer for further failure analysis.

Device life-cycle state is used to reflect the actual state of the device, which is further used to instruct the device on how exactly to protect the assets a device hosts during specific time. For example, when a device is being tested at a silicon manufacturer facility and no OEM or end-customer assets have been provisioned on it, then access to the device, in terms of debug or test, is less restrictive than when a device is with the end-customer.

Life-cycle state is monotonic, meaning it can only be increased. Transition between different life-cycle states is an irreversible process. Immutable RoT is in charge of life-cycle management and it enforces device access policies accordingly.

19.2.3 Secure boot

Secure Boot ensures authenticity, integrity and confidentiality of the device bootloader, firmware, and other software during the boot process and ensures that the intended secure life-cycle state is reached. ECDSA P-256 with SHA-256 or ECDSA P-384 with SHA-384 are there to guarantee authenticity and integrity of the firmware image. Immutable RoT is in charge of enforcing Secure Boot and it does so according to the policies defined by the life-cycle state.

19.2.4 Secure update

Secure Update is the process used to securely update the firmware image in the field. On this chip, the ROM supports secure update using SB3.1 file format. The SB3.1 file is encrypted usingAES-128 or AES-256 and signed using ECDSA P-256 or ECDSA P-384.

Secure Update provides authenticity and confidentiality of the new image. It can also include image rollback protection.

19.2.5 Secure debug

Secure Debug is the process used to securely access debug, following the policy defined by the life-cycle state.

See the Security Reference Manual for more information.

19.2.6 IP protection

IP Protection is a set of mechanisms used to protect confidentiality and integrity of valuable code and data stored on the device.

PRINCE-based memory encryption can be used to provide confidentiality for code/data stored in external flash. Up to 16 independent memory regions/context are supported for XSPI0 and XSPI1.

19.2.7 Secure isolation

The chip has multiple functional domains and power domains with bus architecture partitioned for a balance of optimal performance and power efficiency based on target applications. The chip uses the following components to provide secure isolation:

- ELS with dedicated processing engine and cryptographic accelerators providing fully isolated security functions

- Additional isolation is provided using ARM TrustZone for Armv8-M architectures
  - Memory Protection Unit (MPU) inside Cortex-M33 core. Both CPU0 and CPU1 have 8 MPU regions
  - Security Attribute Unit (SAU) inside Cortex-M33 core. Both CPU0 and CPU1 have 8 SAU regions
- Secure AHB Bus Controller (AHBSC) module provides register to define which domains can access shared SRAM and peripherals present in common and sense domains. See the Security Reference Manual for more information

19.2.8 Secure attestation

Secure Attestation is a set of mechanisms used to provide evidence to a remote party on the device's genuine identity, its software and firmware versions, as well as its integrity and life-cycle state. Runtime Fingerprint (RTF) is the NXP-proprietary attestation mechanism, which measures the device's state during boot-time and run-time as well.

19.2.9 Secure storage

ELS Key store with the following two key hierarchies are supported on this chip:

- SRAM-PUF generated Device Unique Key (DUK)
- OTP-based DUK

NOTE

SRAM-PUF generated DUK is only supported in the case where external non-volatile memory is available.

In the case that the ELS key store is generated from SRAM-PUF-based device unique key, Inline Prince Encryption and Decryption engine (IPED) is used for off-chip flash with option to select regular counter mode (CTR) or Galois Counter mode (GCM) per region for code storage and data.

In the case of the OTP-based DUK hierarchy, the DUK is derived from OTP seed and ELS commands. This key is then used to derive memory encryption (PRINCE) keys. For more information, see the Security Reference Manual for this chip.

19.2.10 Trust provisioning

Trust provisioning is a process used for creation of initial Device Identity keys. Its major objective is to provide a cryptographic proof of the device's origin and to offer a set of tools to OEM for secure provisioning of their own assets. A device-unique private-public key pair is created on every device, the public portion of which is collected and signed by NXP. That signed public key is installed back onto every device in a form of a device-unique certificate, which serves as the actual proof of the device's origin. The corresponding private key, together with other pre-installed key, is then used for authentication and secure connection to the device, enabling secure provisioning of OEM assets.

19.2.11 Secure key management

Secure key management is a process of securing valuable keys and various key material which are essential in maintaining security of the end-user, OEM, and NXP assets. The process strongly relies on the Immutable RoT consisting of ELS Key store, hardware logic, and ROM. A device-unique master key is provided and only used for further key derivation. Part of the derivation data is supplied by the Immutable RoT, accurately reflecting the device's state, making sure different keys are derived in different states. For example, different life-cycle state will often yield a different derived key. Similarly, when the debug port is open a different key will be derived than when the debug port is closed. All the platform keys reside within a security subsystem, hidden from the application core at all times.

See Secure storage and the chip's Security Reference Manual for more information.

19.2.12 Anomaly Detection and Reaction

Anomaly Detection and Reaction describes the processes or algorithms that analyze the device input and output such as sensor data, as well as the software integrity and application operation for abnormal events and, if required, trigger and execute a user-defined security application (e.g. logging the anomaly, issuing a message to the cloud backend, resetting the device, and changing the life-cycle state).

Chapter 20
IO PAD Controller (IOPCTL)

20.1 Chip-specific IOPCTL information

Table 181. Reference links to related information

Topic	Related module	Reference
Full description	IOPCTL	IOPCTL
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Power management	PMC	Power overview PMC
Signal multiplexing		External signals and pin multiplexing
General-purpose input and output	GPIO	General-purpose input and output

20.1.1 Module instances

This device has 3 instances of the IOPCTL module: IOPCTL0, IOPCTL1 and IOPCTL2.

Each IOPCTL supports to configure each pad (bond wired to pin) including function selection (IO muxing) for the corresponding pad. The function 0 (0000b) of IOPCTL.PIO*x_y*[FSEL] is for GPIO.

Table 182. IOPCTL instances

	IOPCTL0	IOPCTL1	IOPCTL2
Control pins in	GPIO0-GPIO3	GPIO8-GPIO10, PMIC_I2C	GPIO4-GPIO7
Control pins[1]	PIO0_0-31, PIO1_0-19, PIO2_0-15, PIO3_0-14	PIO8_0-1, PIO9_0-2, PIO10_0-17, PMIC_I2C_SDA/SCL	PIO4_0-20, PIO5_0-20, PIO6_0-12, PIO7_0-25
Located in	VDD2_COM	VDD1_SENSE	VDDN_COM
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
ipg_clk_s clock source	COMPUTE_MAIN_CLK	SENSE_MAIN_CLK_1	COMMON_VDDN_CLK

1. Not all the pins are available for each package, see Pin assignments for more details.

20.2 Overview

IOPCTL provides IO pad control functions. Each port pin PIOm_n has one IOPCTL register assigned to control the characteristics of the pin.

20.2.1 Block diagram

Figure 64 is the block diagram for PIO0, PIO1 and PIO3-PIO7.

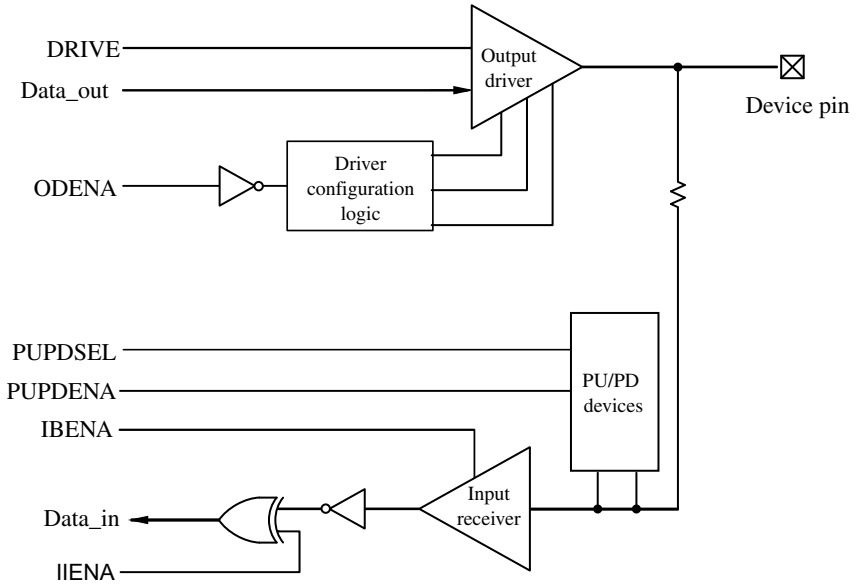


Figure 64. IO cell block diagram 1

Figure 65 is the block diagram for PIO2, PIO8-PIO10, and PMIC_I2C. PMIC_I2C has no DRIVE.

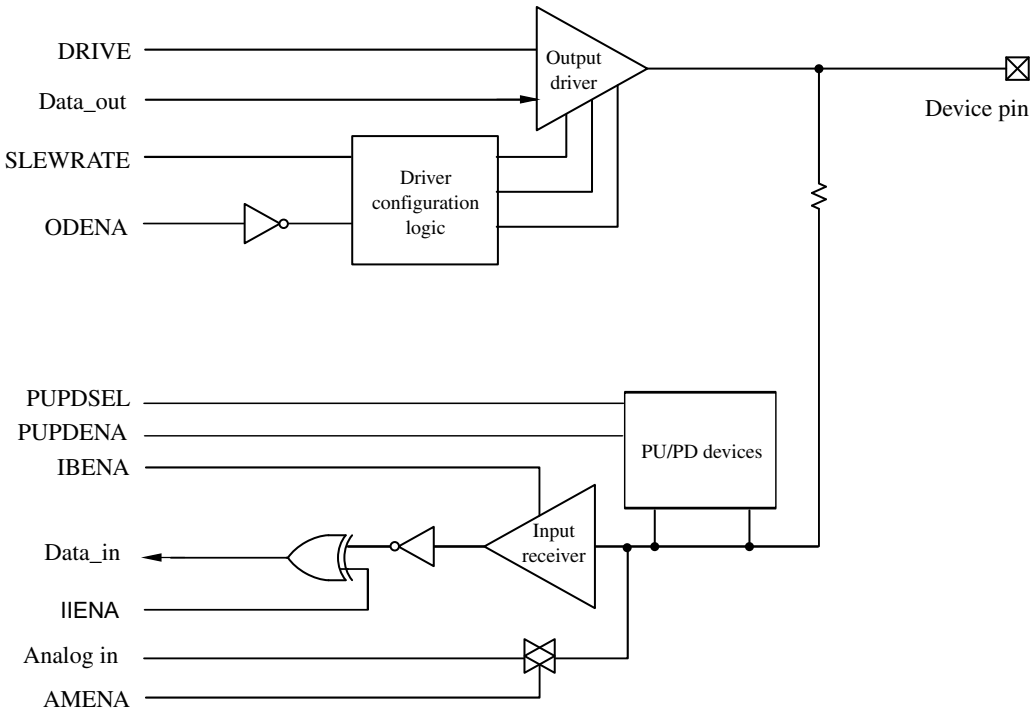


Figure 65. IO cell block diagram 2

Figure 66 shows the Data_out and Data_in in the above two figures.

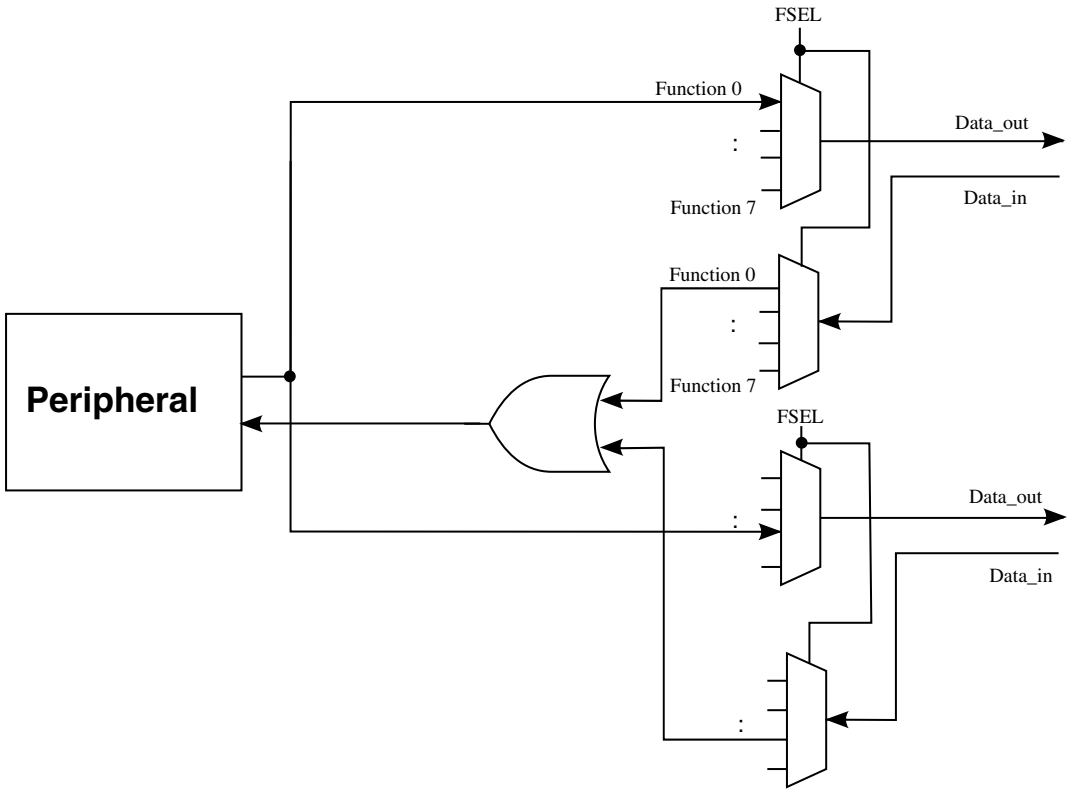


Figure 66. Function selector block diagram

20.2.2 Features

- Function selection from among the available functions for each pin.
- Pull-up resistor, pull-down resistor, or neither.
- Input buffer enable/disable.
- Slew rate.
- Drive strength.
- Analog multiplexor enable for pins that include an analog function.
- Simulated open-drain mode.
- Inverted input function.

20.3 Functional description

20.3.1 Clocking

Table 183. IOPCTL clock

Type of clock	Description
Module clock (ipg_clk_s)	Functions the module and controls the access to the registers.

See the chip-specific information for more details.

20.4 Signals

See External Signals and Pin Multiplexing for details on the available port pins and functions supported.

20.5 Initialization

This module has no initialization considerations.

20.6 Memory map and register definition

This section includes the IOPCTL module memory map and detailed descriptions of all registers.

The IOPCTL registers control the functions of device pins. Each GPIO pin has a dedicated control register to select its function and characteristics. Each pin has a unique set of functional capabilities. Not all pin characteristics are selectable on all pins. For instance, pins that have an I^2C function can be configured for different I^2C-bus modes, while pins that have an analog alternate function have an analog mode can be selected.

Since a particular peripheral function may be allowed on more than one pin, it is possible to configure more than one pin to perform the same function. If a peripheral output function is configured to appear on more than one pin, it will in fact be routed to those pins. If a peripheral input function is defined as coming from more than one source, the values will be logically combined, possibly resulting in incorrect peripheral operation. Therefore care should be taken to avoid this situation.

The IOPCTL registers support only the 32-bit accesses.

20.6.1 IOPCTL0 register descriptions

20.6.1.1 IOPCTL0 memory map

IOPCTL0 base address: 4000_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IOPCTL Configuration (PIO0_0)	32	RW	0000_0000h
4h	IOPCTL Configuration (PIO0_1)	32	RW	0000_0000h
8h	IOPCTL Configuration (PIO0_2)	32	RW	0000_0000h
Ch	IOPCTL Configuration (PIO0_3)	32	RW	0000_0000h
10h	IOPCTL Configuration (PIO0_4)	32	RW	0000_0000h
14h	IOPCTL Configuration (PIO0_5)	32	RW	0000_0000h
18h	IOPCTL Configuration (PIO0_6)	32	RW	0000_0000h
1Ch	IOPCTL Configuration (PIO0_7)	32	RW	0000_0000h
20h	IOPCTL Configuration (PIO0_8)	32	RW	0000_0000h
24h	IOPCTL Configuration (PIO0_9)	32	RW	0000_0000h
28h	IOPCTL Configuration (PIO0_10)	32	RW	0000_0000h
2Ch	IOPCTL Configuration (PIO0_11)	32	RW	0000_0000h
30h	IOPCTL Configuration (PIO0_12)	32	RW	0000_0000h
34h	IOPCTL Configuration (PIO0_13)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
38h	IOPCTL Configuration (PIO0_14)	32	RW	0000_0000h
3Ch	IOPCTL Configuration (PIO0_15)	32	RW	0000_0000h
40h	IOPCTL Configuration (PIO0_16)	32	RW	0000_0000h
44h	IOPCTL Configuration (PIO0_17)	32	RW	0000_0000h
48h	IOPCTL Configuration (PIO0_18)	32	RW	0000_0000h
4Ch	IOPCTL Configuration (PIO0_19)	32	RW	0000_0000h
50h	IOPCTL Configuration (PIO0_20)	32	RW	0000_0000h
54h	IOPCTL Configuration (PIO0_21)	32	RW	0000_0000h
58h	IOPCTL Configuration (PIO0_22)	32	RW	0000_0000h
5Ch	IOPCTL Configuration (PIO0_23)	32	RW	0000_0000h
60h	IOPCTL Configuration (PIO0_24)	32	RW	0000_0000h
64h	IOPCTL Configuration (PIO0_25)	32	RW	0000_0000h
68h	IOPCTL Configuration (PIO0_26)	32	RW	0000_0000h
6Ch	IOPCTL Configuration (PIO0_27)	32	RW	0000_0000h
70h	IOPCTL Configuration (PIO0_28)	32	RW	0000_0000h
74h	IOPCTL Configuration (PIO0_29)	32	RW	0000_0000h
78h	IOPCTL Configuration (PIO0_30)	32	RW	0000_0000h
7Ch	IOPCTL Configuration (PIO0_31)	32	RW	0000_0000h
80h	IOPCTL Configuration (PIO1_0)	32	RW	0000_0000h
84h	IOPCTL Configuration (PIO1_1)	32	RW	0000_0000h
88h	IOPCTL Configuration (PIO1_2)	32	RW	0000_0000h
8Ch	IOPCTL Configuration (PIO1_3)	32	RW	0000_0000h
90h	IOPCTL Configuration (PIO1_4)	32	RW	0000_0000h
94h	IOPCTL Configuration (PIO1_5)	32	RW	0000_0000h
98h	IOPCTL Configuration (PIO1_6)	32	RW	0000_0000h
9Ch	IOPCTL Configuration (PIO1_7)	32	RW	0000_0000h
A0h	IOPCTL Configuration (PIO1_8)	32	RW	0000_0000h
A4h	IOPCTL Configuration (PIO1_9)	32	RW	0000_0000h
A8h	IOPCTL Configuration (PIO1_10)	32	RW	0000_0000h
ACh	IOPCTL Configuration (PIO1_11)	32	RW	0000_0000h
B0h	IOPCTL Configuration (PIO1_12)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
B4h	IOPCTL Configuration (PIO1_13)	32	RW	0000_0000h
B8h	IOPCTL Configuration (PIO1_14)	32	RW	0000_0000h
BCh	IOPCTL Configuration (PIO1_15)	32	RW	0000_0000h
C0h	IOPCTL Configuration (PIO1_16)	32	RW	0000_0000h
C4h	IOPCTL Configuration (PIO1_17)	32	RW	0000_0000h
C8h	IOPCTL Configuration (PIO1_18)	32	RW	0000_0000h
CCh	IOPCTL Configuration (PIO1_19)	32	RW	0000_0000h
100h	IOPCTL Configuration (PIO2_0)	32	RW	0000_0000h
104h	IOPCTL Configuration (PIO2_1)	32	RW	0000_0000h
108h	IOPCTL Configuration (PIO2_2)	32	RW	0000_0000h
10Ch	IOPCTL Configuration (PIO2_3)	32	RW	0000_0000h
110h	IOPCTL Configuration (PIO2_4)	32	RW	0000_0000h
114h	IOPCTL Configuration (PIO2_5)	32	RW	0000_0000h
118h	IOPCTL Configuration (PIO2_6)	32	RW	0000_0000h
11Ch	IOPCTL Configuration (PIO2_7)	32	RW	0000_0000h
120h	IOPCTL Configuration (PIO2_8)	32	RW	0000_0000h
124h	IOPCTL Configuration (PIO2_9)	32	RW	0000_0000h
128h	IOPCTL Configuration (PIO2_10)	32	RW	0000_0000h
12Ch	IOPCTL Configuration (PIO2_11)	32	RW	0000_0000h
130h	IOPCTL Configuration (PIO2_12)	32	RW	0000_0000h
134h	IOPCTL Configuration (PIO2_13)	32	RW	0000_0000h
138h	IOPCTL Configuration (PIO2_14)	32	RW	0000_0000h
13Ch	IOPCTL Configuration (PIO2_15)	32	RW	0000_0000h
180h	IOPCTL Configuration (PIO3_0)	32	RW	0000_0000h
184h	IOPCTL Configuration (PIO3_1)	32	RW	0000_0000h
188h	IOPCTL Configuration (PIO3_2)	32	RW	0000_0000h
18Ch	IOPCTL Configuration (PIO3_3)	32	RW	0000_0000h
190h	IOPCTL Configuration (PIO3_4)	32	RW	0000_0000h
194h	IOPCTL Configuration (PIO3_5)	32	RW	0000_0000h
198h	IOPCTL Configuration (PIO3_6)	32	RW	0000_0000h
19Ch	IOPCTL Configuration (PIO3_7)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
1A0h	IOPCTL Configuration (PIO3_8)	32	RW	0000_0000h
1A4h	IOPCTL Configuration (PIO3_9)	32	RW	0000_0000h
1A8h	IOPCTL Configuration (PIO3_10)	32	RW	0000_0000h
1ACh	IOPCTL Configuration (PIO3_11)	32	RW	0000_0000h
1B0h	IOPCTL Configuration (PIO3_12)	32	RW	0000_0000h
1B4h	IOPCTL Configuration (PIO3_13)	32	RW	0000_0000h
1B8h	IOPCTL Configuration (PIO3_14)	32	RW	0000_0000h

20.6.1.2 IOPCTL Configuration (PIO0_0 - PIO3_14)

Offset

Registers in this array exist only for the following combinations of index values.

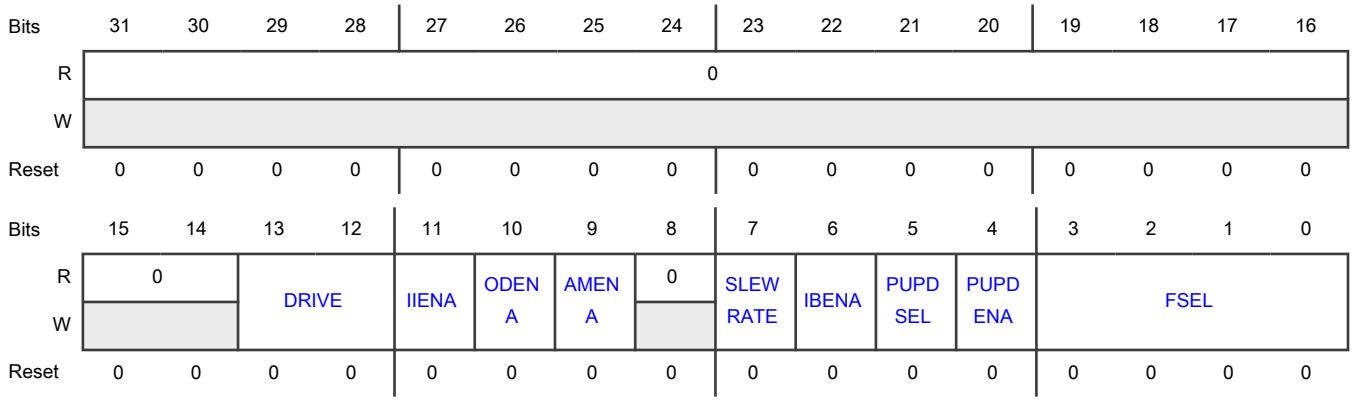
Index *y*	Index *x*
0	0–31
1	0–19
2	0–15
3	0–14

Register	Offset
PIOy_x	0h + (y × 80h) + (x × 4h)

Function

Each port pin PIO has one IOPCTL register assigned to control the characteristics of the pin.

Diagram



Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-14
—</td><td>Reserved</td></tr>
<tr><td>13-12
DRIVE</td><td>Drive
For PIO0, PIO1 and PIO3, this field selects transmitter current drive.
• 00b - 100 Ω
• 01b - 66 Ω
• 10b - 50 Ω
• 11b - 33 Ω
For PIO2, the “Full output drive” setting has twice the high and low drive capability of the “Normal output drive” setting. The drive strength depends on the I/O supply voltage for the specific pin, which can be found in the chip datasheet. See device datasheet for details.
• x0b - Normal output drive
• x1b - Full output drive, twice the drive of normal mode</td></tr>
<tr><td>11
IIENA</td><td>Input Invert Enable
Enables input invert.
0b - Disables
1b - Enables</td></tr>
<tr><td>10
ODENA</td><td>Open-drain Mode Enable
Enables open-drain mode.
0b - Disables for normal push-pull output
1b - Enables for simulated open-drain output (high drive disabled)</td></tr>
<tr><td>9
AMENA</td><td>Analog Mux Enable
Enables analog multiplexor.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>IOPCTL0</td><td>PIO2_0–PIO2_15</td><td>PIO0_0–PIO1_19
PIO3_0–PIO3_14</td></tr></table>
0b - Disables for digital pin function
1b - Enables for analog pin function</td></tr>
<tr><td>8</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>7
SLEWRATE</td><td>Slew Rate Control
Enables the output slew rate control to limit the output rate change and maximum toggle frequency. This effect is voltage and load dependent, see datasheet for details.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>IOPCTL0</td><td>PIO2_0–PIO2_15</td><td>PIO0_0–PIO1_19
PIO3_0–PIO3_14</td></tr></table>
0b - Disables, in standard mode.
1b - Enables, in slow mode.</td></tr>
<tr><td>6
IBENA</td><td>Input Buffer Enable
Enables the input buffer for digital functions. It can be disabled when not needed and must be disabled when the pin is used for an analog function.
Input buffer is not needed for a GPIO that is used as an output and is never read back by software. Peripheral functions that are generally outputs do need the input buffer enabled because the peripheral uses it to sense the situation on the pin as part of its function. If there is any doubt, it is best to leave the input buffer enabled
0b - Disables
1b - Enables</td></tr>
<tr><td>5
PUPDSEL</td><td>Pullup or Pulldown Selector
Enables the internal pull-up or pull-down resistor when PIO[PUPDENA]=1.
0b - Enables the internal pull-down resistor.
1b - Enables the internal pull-up resistor.</td></tr>
<tr><td>4
PUPDENA</td><td>Pullup / Pulldown Enable
Enables the internal pull resistors on each port pin.
0b - Disable
1b - Enable</td></tr>
<tr><td>3-0
FSEL</td><td>Function Selector (Digital Function)
Selects pin function according to the Pin assignments.
NOTE
Do not select the reserved value.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000b - Function 0 0001b - Function 1 0010b - Function 2 0011b - Function 3 0100b - Function 4 0101b - Function 5 0110b - Function 6 0111b - Function 7 All other values are reserved.

20.6.2 IOPCTL1 register descriptions

20.6.2.1 IOPCTL1 memory map

IOPCTL1 base address: 4006_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IOPCTL Configuration (PIO8_0)	32	RW	0000_0055h
4h	IOPCTL Configuration (PIO8_1)	32	RW	0000_0075h
8h	IOPCTL Configuration (PIO8_2)	32	RW	0000_0075h
Ch	IOPCTL Configuration (PIO8_3)	32	RW	0000_0005h
10h	IOPCTL Configuration (PIO8_4)	32	RW	0000_0075h
14h	IOPCTL Configuration (PIO8_5)	32	RW	0000_0000h
18h	IOPCTL Configuration (PIO8_6)	32	RW	0000_0000h
1Ch	IOPCTL Configuration (PIO8_7)	32	RW	0000_0000h
20h	IOPCTL Configuration (PIO8_8)	32	RW	0000_0000h
24h	IOPCTL Configuration (PIO8_9)	32	RW	0000_0000h
28h	IOPCTL Configuration (PIO8_10)	32	RW	0000_0000h
2Ch	IOPCTL Configuration (PIO8_11)	32	RW	0000_0000h
30h	IOPCTL Configuration (PIO8_12)	32	RW	0000_0000h
34h	IOPCTL Configuration (PIO8_13)	32	RW	0000_0000h
38h	IOPCTL Configuration (PIO8_14)	32	RW	0000_0000h
3Ch	IOPCTL Configuration (PIO8_15)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
40h	IOPCTL Configuration (PIO8_16)	32	RW	0000_0000h
44h	IOPCTL Configuration (PIO8_17)	32	RW	0000_0000h
48h	IOPCTL Configuration (PIO8_18)	32	RW	0000_0000h
4Ch	IOPCTL Configuration (PIO8_19)	32	RW	0000_0000h
50h	IOPCTL Configuration (PIO8_20)	32	RW	0000_0000h
54h	IOPCTL Configuration (PIO8_21)	32	RW	0000_0000h
58h	IOPCTL Configuration (PIO8_22)	32	RW	0000_0000h
5Ch	IOPCTL Configuration (PIO8_23)	32	RW	0000_0000h
60h	IOPCTL Configuration (PIO8_24)	32	RW	0000_0000h
64h	IOPCTL Configuration (PIO8_25)	32	RW	0000_0000h
68h	IOPCTL Configuration (PIO8_26)	32	RW	0000_0000h
6Ch	IOPCTL Configuration (PIO8_27)	32	RW	0000_0000h
70h	IOPCTL Configuration (PIO8_28)	32	RW	0000_0000h
74h	IOPCTL Configuration (PIO8_29)	32	RW	0000_0000h
78h	IOPCTL Configuration (PIO8_30)	32	RW	0000_0000h
7Ch	IOPCTL Configuration (PIO8_31)	32	RW	0000_0000h
80h	IOPCTL Configuration (PIO9_0)	32	RW	0000_0000h
84h	IOPCTL Configuration (PIO9_1)	32	RW	0000_0000h
88h	IOPCTL Configuration (PIO9_2)	32	RW	0000_0000h
100h	IOPCTL Configuration (PIO10_0)	32	RW	0000_0000h
104h	IOPCTL Configuration (PIO10_1)	32	RW	0000_0000h
108h	IOPCTL Configuration (PIO10_2)	32	RW	0000_0000h
10Ch	IOPCTL Configuration (PIO10_3)	32	RW	0000_0000h
110h	IOPCTL Configuration (PIO10_4)	32	RW	0000_0000h
114h	IOPCTL Configuration (PIO10_5)	32	RW	0000_0000h
118h	IOPCTL Configuration (PIO10_6)	32	RW	0000_0000h
11Ch	IOPCTL Configuration (PIO10_7)	32	RW	0000_0000h
120h	IOPCTL Configuration (PIO10_8)	32	RW	0000_0000h
124h	IOPCTL Configuration (PIO10_9)	32	RW	0000_0000h
128h	IOPCTL Configuration (PIO10_10)	32	RW	0000_0000h
12Ch	IOPCTL Configuration (PIO10_11)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
130h	IOPCTL Configuration (PIO10_12)	32	RW	0000_0000h
134h	IOPCTL Configuration (PIO10_13)	32	RW	0000_0000h
138h	IOPCTL Configuration (PIO10_14)	32	RW	0000_0000h
13Ch	IOPCTL Configuration (PIO10_15)	32	RW	0000_0000h
140h	IOPCTL Configuration (PIO10_16)	32	RW	0000_0000h
144h	IOPCTL Configuration (PIO10_17)	32	RW	0000_0000h
180h	PMIC_I2C_SDA (PMIC_I2C_SDA)	32	RW	0000_0000h
184h	PMIC_I2C_SCL (PMIC_I2C_SCL)	32	RW	0000_0000h

20.6.2.2 IOPCTL Configuration (PIO8_0 - PIO10_17)

Offset

Registers in this array exist only for the following combinations of index values.

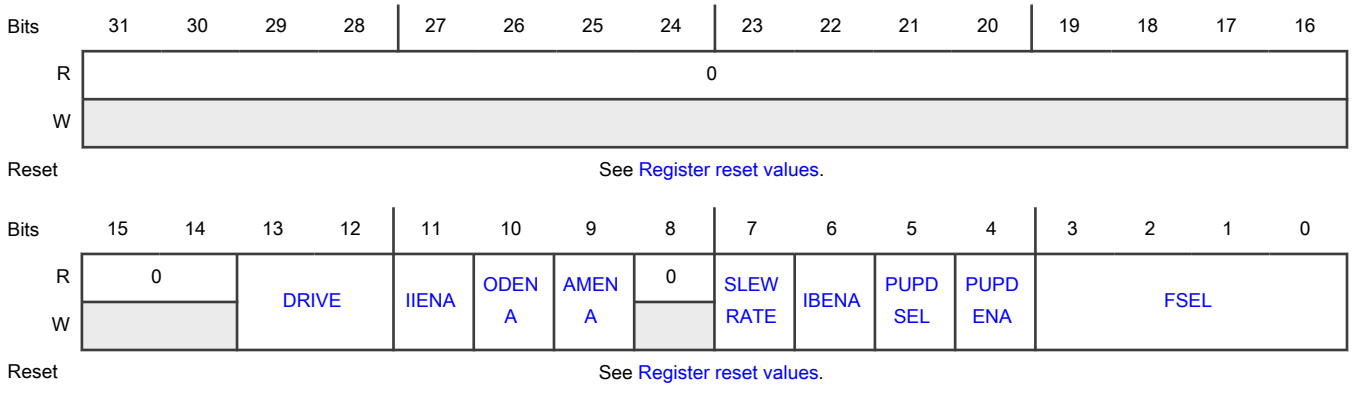
Index *y*	Index *x*
8	0–31
9	0–2
10	0–17

Register	Offset
PIOy_x	-400h + (y × 80h) + (x × 4h)

Function

Each port pin PIO has one IOPCTL register assigned to control the characteristics of the pin.

Diagram



Register reset values

Register	Reset value
PIO8_0	0000_0055h
PIO8_1–PIO8_2	0000_0075h
PIO8_3	0000_0005h
PIO8_4	0000_0075h
PIO8_5–PIO10_17	0000_0000h
PIO9_3–PIO9_31	—

Fields

Field	Function
31-14 —	Reserved
13-12 DRIVE	Drive Selector The “Full output drive” setting has twice the high and low drive capability of the “Normal output drive” setting. The drive strength depends on the I/O supply voltage for the specific pin, which can be found in the device datasheet. See device datasheet for details. x0b - Normal output drive x1b - Full output drive, twice the drive of normal mode.
11 IIENA	Input Invert Enable Enables input function invert 0b - Disables. Input function is not inverted 1b - Enables, input is function inverted
10 ODENA	Open-drain Mode Enable Enables open-drain mode. 0b - Disables for Normal push-pull output 1b - Enables for open-drain output (high drive is disabled)
9 AMENA	Analog Mux Enable Enables analog multiplexor. 0b - Disables for digital pin function 1b - Enables for analog pin function
8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 SLEWRATE	Slew Rate Control Enables the output slew rate control to limit the output rate change and maximum toggle frequency. This effect is voltage and load dependent, see datasheet for details. 0b - Disables. Standard mode. 1b - Enables. Slow mode.
6 IBENA	Input Buffer Enable Enables the input buffer for digital functions. It can be disabled when not needed and must be disabled when the pin is used for an analog function. Input buffer is not needed for a GPIO that is used as an output and is never read back by software. Peripheral functions that are generally outputs do need the input buffer enabled because the peripheral uses it to sense the situation on the pin as part of its function. If there is any doubt, it is best to leave the input buffer enabled 0b - Disables 1b - Enables
5 PUPDSEL	Pullup or Pulldown Selector Enables the internal pull-up or pull-down resistor when PIO[PUPDENA]=1. 0b - Enables the internal pull-down resistor. 1b - Enables the internal pull-up resistor.
4 PUPDENA	Pullup / Pulldown Enable Enables the internal pull resistors on each port pin. 0b - Disable 1b - Enable
3-0 FSEL	Function Selector (Digital Function) Selects pin function according to the Pin assignments. **NOTE** Do not select the reserved value. 0000b - Function 0 0001b - Function 1 0010b - Function 2 0011b - Function 3 0100b - Function 4 0101b - Function 5 0110b - Function 6 0111b - Function 7

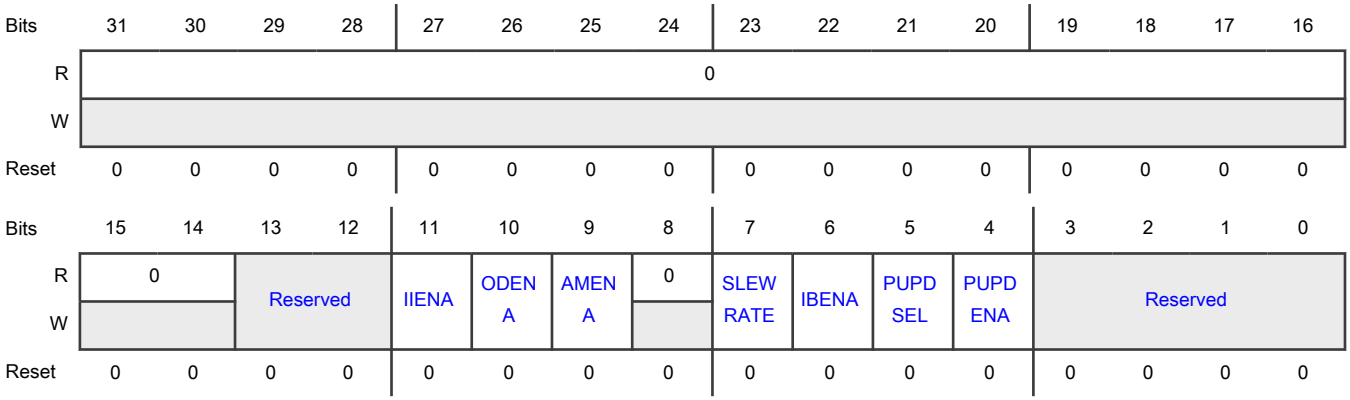
20.6.2.3 PMIC_I2C_SDA (PMIC_I2C_SDA)

Offset

Register	Offset
PMIC_I2C_SDA	180h

Function
Pin Control for PMIC_I2C_SDA.

Diagram



Fields

Field	Function
31-14 —	Reserved
13-12 —	Reserved
11 IIENA	Input Invert Enable Enables input invert. 0b - Disables 1b - Enables
10 ODENA	Open-drain Mode Enable Enables open-drain mode. 0b - Disables for normal push-pull output 1b - Enables for simulated open-drain output (high drive disabled)
9	Analog Mux Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
AMENA	Enables analog multiplexor. 0b - Disables for digital pin function 1b - Enables for analog pin function
8 —	Reserved
7 SLEWRATE	Slew Rate Control Enables the output slew rate control to limit the output rate change and maximum toggle frequency. This effect is voltage and load dependent, see datasheet for details. 0b - Disables, in standard mode. 1b - Enables, in slow mode.
6 IBENA	Input Buffer Enable Enables the input buffer for digital functions. It can be disabled when not needed and must be disabled when the pin is used for an analog function. Input buffer is not needed for a GPIO that is used as an output and is never read back by software. Peripheral functions that are generally outputs do need the input buffer enabled because the peripheral uses it to sense the situation on the pin as part of its function. If there is any doubt, it is best to leave the input buffer enabled 0b - Disables 1b - Enables
5 PUPDSEL	Pullup or Pulldown Selector Enables the internal pull-up or pull-down resistor when PMIC_I2C_SDA[PUPDENA]=1. 0b - Enables the internal pull-down resistor. 1b - Enables the internal pull-up resistor.
4 PUPDENA	Pullup / Pulldown Enable Enables internal pull resistor on the pin. 0b - Disables 1b - Enables
3-0 —	Reserved

20.6.2.4 PMIC_I2C_SCL (PMIC_I2C_SCL)

Offset

Register	Offset
PMIC_I2C_SCL	184h

Function
Pin Control for PMIC_I2C_SCL.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		Reserved		IIENA	ODENA	AMENA	0	SLEW RATE	IBENA	PUPD SEL	PUPD ENA	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-14 —	Reserved
13-12 —	Reserved
11 IIENA	Input Invert Enable Enables input invert. 0b - Disables 1b - Enables
10 ODENA	Open-drain Mode Enable Enables open-drain mode. 0b - Disables for normal push-pull output 1b - Enables for simulated open-drain output (high drive disabled)
9	Analog Mux Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
AMENA	Enables analog multiplexor. 0b - Disables for digital pin function 1b - Enables for analog pin function
8 —	Reserved
7 SLEWRATE	Slew Rate Control Enables the output slew rate control to limit the output rate change and maximum toggle frequency. This effect is voltage and load dependent, see datasheet for details. 0b - Disables. Standard mode. 1b - Enables. Slow mode.
6 IBENA	Input Buffer Enable Enables the input buffer for digital functions. It can be disabled when not needed and must be disabled when the pin is used for an analog function. Input buffer is not needed for a GPIO that is used as an output and is never read back by software. Peripheral functions that are generally outputs do need the input buffer enabled because the peripheral uses it to sense the situation on the pin as part of its function. If there is any doubt, it is best to leave the input buffer enabled 0b - Disables 1b - Enables
5 PUPDSEL	Pullup or Pulldown Selector Enables the internal pull-up or pull-down resistor when PMIC_I2C_SDA[PUPDENA]=1. 0b - Enables the internal pull-down resistor. 1b - Enables the internal pull-up resistor.
4 PUPDENA	Pullup / Pulldown Enable Enables internal pull resistor on the pin. 0b - Disables 1b - Enables
3-0 —	Reserved

20.6.3 IOPCTL2 register descriptions

20.6.3.1 IOPCTL2 memory map

IOPCTL2 base address: 400A_5000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IOPCTL Configuration (PIO4_0)	32	RW	0000_0000h
4h	IOPCTL Configuration (PIO4_1)	32	RW	0000_0000h
8h	IOPCTL Configuration (PIO4_2)	32	RW	0000_0000h
Ch	IOPCTL Configuration (PIO4_3)	32	RW	0000_0000h
10h	IOPCTL Configuration (PIO4_4)	32	RW	0000_0000h
14h	IOPCTL Configuration (PIO4_5)	32	RW	0000_0000h
18h	IOPCTL Configuration (PIO4_6)	32	RW	0000_0000h
1Ch	IOPCTL Configuration (PIO4_7)	32	RW	0000_0000h
20h	IOPCTL Configuration (PIO4_8)	32	RW	0000_0000h
24h	IOPCTL Configuration (PIO4_9)	32	RW	0000_0000h
28h	IOPCTL Configuration (PIO4_10)	32	RW	0000_0000h
2Ch	IOPCTL Configuration (PIO4_11)	32	RW	0000_0030h
30h	IOPCTL Configuration (PIO4_12)	32	RW	0000_0000h
34h	IOPCTL Configuration (PIO4_13)	32	RW	0000_0000h
38h	IOPCTL Configuration (PIO4_14)	32	RW	0000_0000h
3Ch	IOPCTL Configuration (PIO4_15)	32	RW	0000_0000h
40h	IOPCTL Configuration (PIO4_16)	32	RW	0000_0000h
44h	IOPCTL Configuration (PIO4_17)	32	RW	0000_0000h
48h	IOPCTL Configuration (PIO4_18)	32	RW	0000_0000h
4Ch	IOPCTL Configuration (PIO4_19)	32	RW	0000_0000h
50h	IOPCTL Configuration (PIO4_20)	32	RW	0000_0000h
80h	IOPCTL Configuration (PIO5_0)	32	RW	0000_0030h
84h	IOPCTL Configuration (PIO5_1)	32	RW	0000_0000h
88h	IOPCTL Configuration (PIO5_2)	32	RW	0000_0000h
8Ch	IOPCTL Configuration (PIO5_3)	32	RW	0000_0000h
90h	IOPCTL Configuration (PIO5_4)	32	RW	0000_0000h
94h	IOPCTL Configuration (PIO5_5)	32	RW	0000_0000h
98h	IOPCTL Configuration (PIO5_6)	32	RW	0000_0000h
9Ch	IOPCTL Configuration (PIO5_7)	32	RW	0000_0000h
A0h	IOPCTL Configuration (PIO5_8)	32	RW	0000_0000h
A4h	IOPCTL Configuration (PIO5_9)	32	RW	0000_0000h
A8h	IOPCTL Configuration (PIO5_10)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
ACh	IOPCTL Configuration (PIO5_11)	32	RW	0000_0030h
B0h	IOPCTL Configuration (PIO5_12)	32	RW	0000_0000h
B4h	IOPCTL Configuration (PIO5_13)	32	RW	0000_0000h
B8h	IOPCTL Configuration (PIO5_14)	32	RW	0000_0000h
BCh	IOPCTL Configuration (PIO5_15)	32	RW	0000_0000h
C0h	IOPCTL Configuration (PIO5_16)	32	RW	0000_0000h
C4h	IOPCTL Configuration (PIO5_17)	32	RW	0000_0000h
C8h	IOPCTL Configuration (PIO5_18)	32	RW	0000_0000h
CCh	IOPCTL Configuration (PIO5_19)	32	RW	0000_0000h
D0h	IOPCTL Configuration (PIO5_20)	32	RW	0000_0000h
100h	IOPCTL Configuration (PIO6_0)	32	RW	0000_0000h
104h	IOPCTL Configuration (PIO6_1)	32	RW	0000_0000h
108h	IOPCTL Configuration (PIO6_2)	32	RW	0000_0030h
10Ch	IOPCTL Configuration (PIO6_3)	32	RW	0000_0000h
110h	IOPCTL Configuration (PIO6_4)	32	RW	0000_0000h
114h	IOPCTL Configuration (PIO6_5)	32	RW	0000_0000h
118h	IOPCTL Configuration (PIO6_6)	32	RW	0000_0000h
11Ch	IOPCTL Configuration (PIO6_7)	32	RW	0000_0000h
120h	IOPCTL Configuration (PIO6_8)	32	RW	0000_0000h
124h	IOPCTL Configuration (PIO6_9)	32	RW	0000_0000h
128h	IOPCTL Configuration (PIO6_10)	32	RW	0000_0000h
12Ch	IOPCTL Configuration (PIO6_11)	32	RW	0000_0000h
130h	IOPCTL Configuration (PIO6_12)	32	RW	0000_0030h
180h	IOPCTL Configuration (PIO7_0)	32	RW	0000_0000h
184h	IOPCTL Configuration (PIO7_1)	32	RW	0000_0000h
188h	IOPCTL Configuration (PIO7_2)	32	RW	0000_0000h
18Ch	IOPCTL Configuration (PIO7_3)	32	RW	0000_0000h
190h	IOPCTL Configuration (PIO7_4)	32	RW	0000_0000h
194h	IOPCTL Configuration (PIO7_5)	32	RW	0000_0000h
198h	IOPCTL Configuration (PIO7_6)	32	RW	0000_0000h
19Ch	IOPCTL Configuration (PIO7_7)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
1A0h	IOPCTL Configuration (PIO7_8)	32	RW	0000_0000h
1A4h	IOPCTL Configuration (PIO7_9)	32	RW	0000_0000h
1A8h	IOPCTL Configuration (PIO7_10)	32	RW	0000_0000h
1ACh	IOPCTL Configuration (PIO7_11)	32	RW	0000_0000h
1B0h	IOPCTL Configuration (PIO7_12)	32	RW	0000_0000h
1B4h	IOPCTL Configuration (PIO7_13)	32	RW	0000_0000h
1B8h	IOPCTL Configuration (PIO7_14)	32	RW	0000_0000h
1BCh	IOPCTL Configuration (PIO7_15)	32	RW	0000_0000h
1C0h	IOPCTL Configuration (PIO7_16)	32	RW	0000_0000h
1C4h	IOPCTL Configuration (PIO7_17)	32	RW	0000_0000h
1C8h	IOPCTL Configuration (PIO7_18)	32	RW	0000_0000h
1CCh	IOPCTL Configuration (PIO7_19)	32	RW	0000_0000h
1D0h	IOPCTL Configuration (PIO7_20)	32	RW	0000_0000h
1D4h	IOPCTL Configuration (PIO7_21)	32	RW	0000_0000h
1D8h	IOPCTL Configuration (PIO7_22)	32	RW	0000_0000h
1DCh	IOPCTL Configuration (PIO7_23)	32	RW	0000_0000h
1E0h	IOPCTL Configuration (PIO7_24)	32	RW	0000_0000h
1E4h	IOPCTL Configuration (PIO7_25)	32	RW	0000_0000h

20.6.3.2 IOPCTL Configuration (PIO4_0 - PIO7_25)

Offset

Registers in this array exist only for the following combinations of index values.

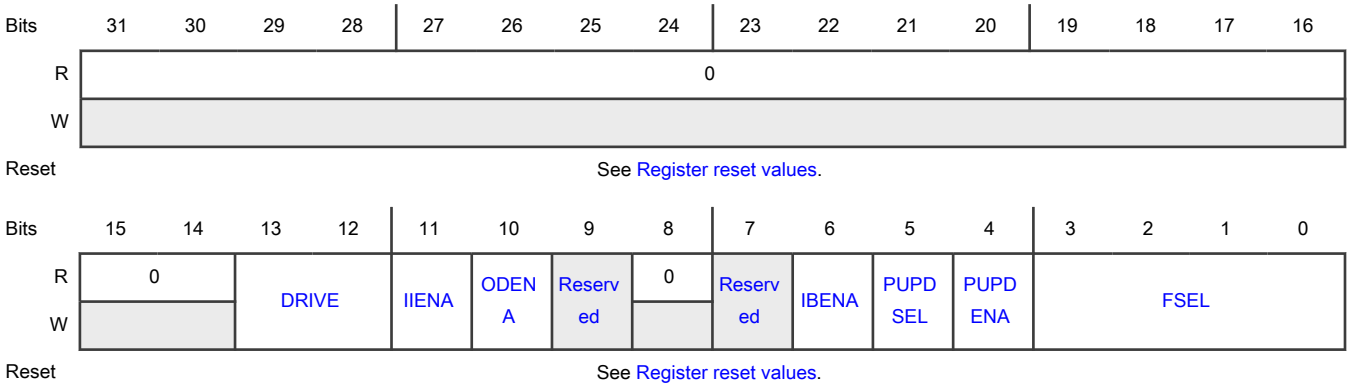
Index *y*	Index *x*
4–5	0–20
6	0–12
7	0–25

Register	Offset
PIOy_x	-200h + (y × 80h) + (x × 4h)

Function

Each port pin PIO has one IOPCTL register assigned to control the characteristics of the pin.

Diagram



Register reset values

Register	Reset value
PIO4_0–PIO4_10	0000_0000h
PIO4_11	0000_0030h
PIO4_12–PIO4_20	0000_0000h
PIO5_0	0000_0030h
PIO5_1–PIO5_10	0000_0000h
PIO5_11	0000_0030h
PIO5_12–PIO6_1	0000_0000h
PIO6_2	0000_0030h
PIO6_3–PIO6_11	0000_0000h
PIO6_12	0000_0030h
PIO6_13–PIO6_20	—
PIO7_0–PIO7_25	0000_0000h

Fields

Field	Function
31-14 —	Reserved
13-12 DRIVE	Drive Selects transmitter current drive.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - 100 Ω 01b - 66 Ω 10b - 50 Ω 11b - 33 Ω
11 IIENA	Input Invert Enable Enables input invert. 0b - Disables 1b - Enables
10 ODENA	Open-drain Mode Enable Enables open-drain mode. 0b - Disables for normal push-pull output 1b - Enables for simulated open-drain output (high drive disabled)
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 IBENA	Input Buffer Enable Enables the input buffer for digital functions. It can be disabled when not needed and must be disabled when the pin is used for an analog function. Input buffer is not needed for a GPIO that is used as an output and is never read back by software. Peripheral functions that are generally outputs do need the input buffer enabled because the peripheral uses it to sense the situation on the pin as part of its function. If there is any doubt, it is best to leave the input buffer enabled 0b - Disables 1b - Enables
5 PUPDSEL	Pullup or Pulldown Selector Enables the internal pull-up or pull-down resistor when PIO[PUPDENA]=1. 0b - Enables the internal pull-down resistor. 1b - Enables the internal pull-up resistor.
4 PUPDENA	Pullup / Pulldown Enable Enables the internal pull resistors on each port pin.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
3-0 FSEL	Function Selector (Digital Function) Selects pin function according to the Pin assignments. **NOTE** Do not select the reserved value. 0000b - Function 0 0001b - Function 1 0010b - Function 2 0011b - Function 3 0100b - Function 4 0101b - Function 5 0110b - Function 6 0111b - Function 7 All other values are reserved.

Chapter 21
Pin Interrupt and Pattern Match (PINT)

21.1 Chip-specific PINT information

Table 184. Reference links to related information

Topic	Related module	Reference
Full description	PINT	PINT
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Input multiplexing	INPUTMUX	See PINT0_TRIG0 - PINT0_TRIG7 registers in INPUTMUX0 and PINT1_TRIG0 - PINT1_TRIG3 registers in INPUTMUX1.

21.1.1 Module instances

This device has two instances of the PINT module, PINT0 and PINT1.

PINT0 interrupt 0-7 are used to interrupt CPU0 and PINT1 interrupt 0-3 are for CPU1 interrupt. The edge and level interrupts are supported even as a wake-up source of the sleep and deep-sleep modes.

Table 185. PINT instances

	PINT0	PINT1
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Num. of interrupt sources	8	4
Interrupt input configured by	INPUTMUX0	INPUTMUX1
Interrupt input pins	PIO0_0-31, PIO1_0-19	PIO8_0-31, PIO10_0-17

21.1.2 PMCFG[CFGy] configuration for LPUART

When an LP_FLEXCOMM is used as LPUART, and the related LP_FLEXCOMMx_TRIG[TRIGIN] in INPUTMUX is configured as 1100b to select PINT Boolean pattern-match output as input trigger , the related PMCFG[CFGy] must not be configured as 111b (Event).

21.1.3 Initialization

- Select up to eight external interrupt pins from all available digital port pins on ports 0 and 1 in INPUTMUX. The pin selection process is the same for pin interrupts and the pattern match engine. The two features are mutually exclusive.
- Enable the clock to the PINT0 module via CLKCTL0_PSCCTL3[PINT0] and enable the clock to the PINT1 module via CLKCTL1_PSCCTL1[PINT1].

- Clear the PINT0 peripheral reset via RSTCTL0_PRSTCTL4[PINT0] or RSTCTL0_PRSTCTL4_CLR[PINT0] and clear the PINT1 peripheral reset via RSTCTL1_PRSTCTL0[PINT1] or RSTCTL1_PRSTCTL0_CLR[PINT1].

21.1.3.1 Pin interrupts initialization

In addition to the initialization steps described above, for Pin Interrupts initialization do the following:

- Each selected pin interrupt is assigned to one interrupt in the INPUTMUX. Note that 8 are available for PINT0 and 4 are available for PINT1.

21.1.3.2 Pattern Match Initialization

In addition to the initialization steps described above, for Pattern match engine initialization do the following:

- Each bit slice of the pattern match engine is assigned to one interrupt (see INPUTMUX). Note that 8 are available for PINT0 and 4 are available for PINT1.

21.1.3.3 Configure pins as pin interrupts or as inputs to the pattern match engine

1. Determine the pins that you want to serve as pin interrupts or pattern match inputs. See the data sheet for determining the GPIO port pin number associated with the package pin.
2. For each pin selected, program the GPIO port pin number from port 0 into one of the eight PINT0_TRIG registers and port 1 into one of the four PINT1_TRIG registers in the INPUTMUX module.

   NOTE

   The port pin number serves to identify the pin to the PINT_TRIG register. Any function, including GPIO, can be assigned to this pin via IOPCTL.

3. Enable each pin interrupt in the PINT.

Once the pin interrupts or pattern match inputs are configured, the pin interrupt detection levels or the pattern match boolean expression can be set up.

NOTE

The inputs to the Pin interrupt select registers bypass the IOPCTL function selection. They do not have to be selected as GPIO in IOPCTL. Make sure that no analog function is selected on pins that are input to the pin interrupts.

21.2 Overview

The Pin Interrupt and Pattern Match (PINT) IP provides interrupt functionality for a selected set of GPIO pins. Per-pin edge- or level-sensitive interrupts can be generated. This IP also includes a pattern match engine that uses the selected GPIO pins as inputs to a Boolean expression. When the expression is satisfied, an interrupt is generated.

21.2.1 Block diagram

From all available GPIO pins, select up to 8 pins in the INPUTMUX module to serve as external interrupt pins (see the "INPUTMUX" chapter for more details). Each external interrupt pins connect to eight individual interrupts in the NVIC. It is possible to create external interrupt pins based either on rising or falling edges or on the input level.

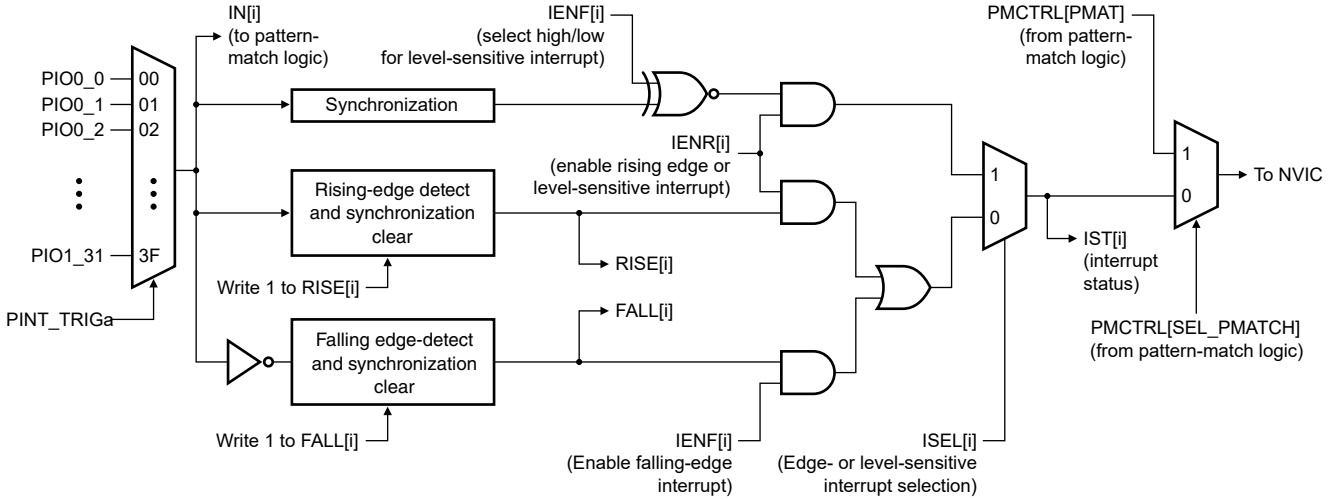


Figure 67. Pin Interrupt diagram

21.2.2 Features

- Pin interrupts
  - Select up to 8 pins from all GPIO pins on ports 0 and 1 as edge- or level-sensitive interrupt requests. Each request creates a separate interrupt in the NVIC.
  - Edge-sensitive interrupt pins interrupt on rising or falling edges or both.
  - Level-sensitive interrupt pins are high- or low-active.
- Pattern-match engine
  - Select up to 8 pins from all digital pins on ports 0 and 1 to contribute to a Boolean expression. The Boolean expression consists of specified levels and/or transitions on various combinations of these pins.
  - Each bit slice minterm (product term) consisting of the specified Boolean expression can generate its own, dedicated interrupt request.
  - You can program any occurrence of a pattern match to generate a Receive Event (RXEV) notification to the CPU.
  - Use pattern match, in conjunction with software, to create complex state machines based on pin inputs.

21.3 Functional description

21.3.1 Pattern-match engine

With pattern matching, you can construct complex Boolean expressions using the same set of 8 GPIO pins you selected for interrupts. Implement each term in the Boolean expression as one slice of the pattern-match engine. A slice consists of an input selector and detection logic that continuously monitors the selected input and creates a HIGH output (true). If the input qualifies as detected, several terms combine to a minterm. Assertion of a pin interrupt occurs when the minterm evaluates as true.

The detection logic of each slice can detect the following events on the selected input:

- Edge with memory (sticky): A rising edge, a falling edge, or a rising or falling edge that is detected at any time after the edge-detection mechanism has been cleared. The input qualifies as detected (the detection logic output remains high) until the pattern-match engine detect logic is cleared again.
- Event (nonsticky): Every time an edge (rising or falling) is detected, the detection logic output goes HIGH. After one clock cycle, writing 1 to this bit enables the detection logic to detect another edge.
- Level: A high or low level on the selected input.

The following figure shows the details of the edge-detection logic for each slice.

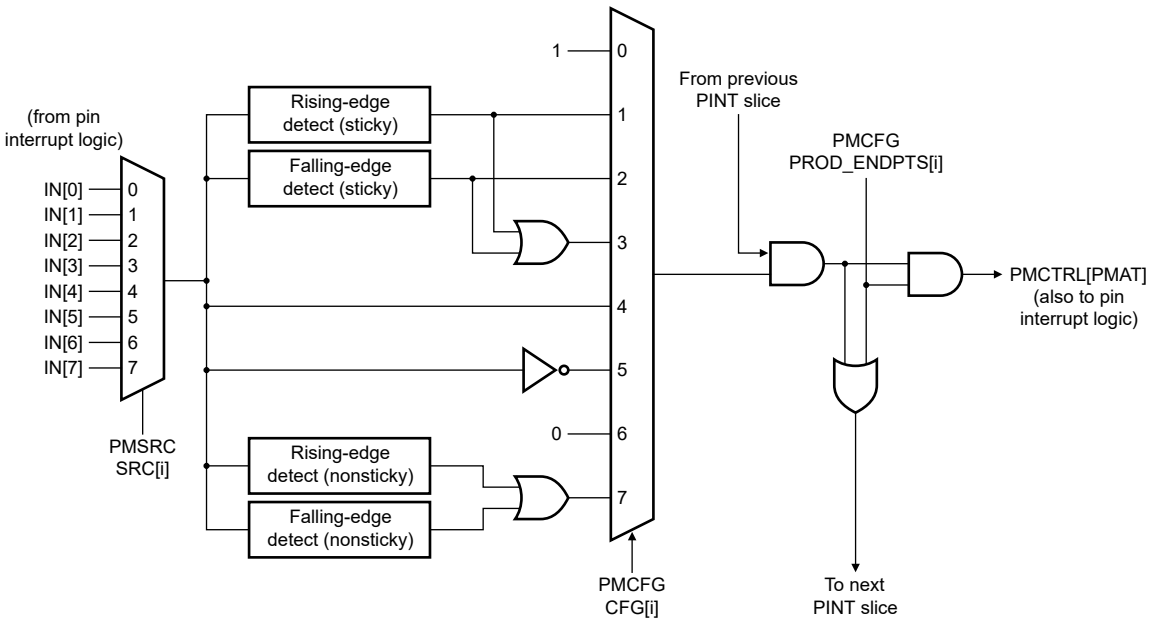


Figure 68. Pattern-match bit slice

Combined sticky and nonsticky events form a pin interrupt whenever a rising or falling edge occurs after a qualifying edge event.

Combining a level detect with an event detect can create a time window during which rising or falling edges will create a pin interrupt. See Pattern-match engine edge-detect examples for details.

The following figure shows connections between the pins and the pattern-match engine. The INPUTMUX block selects all pins that are inputs to the pattern-match engine and can be GPIO port pins or other pin functions depending on the configuration.

NOTE

The pattern-match feature requires clocks in order to operate. It cannot generate an interrupt or wake the chip during reduced power modes below sleep mode.

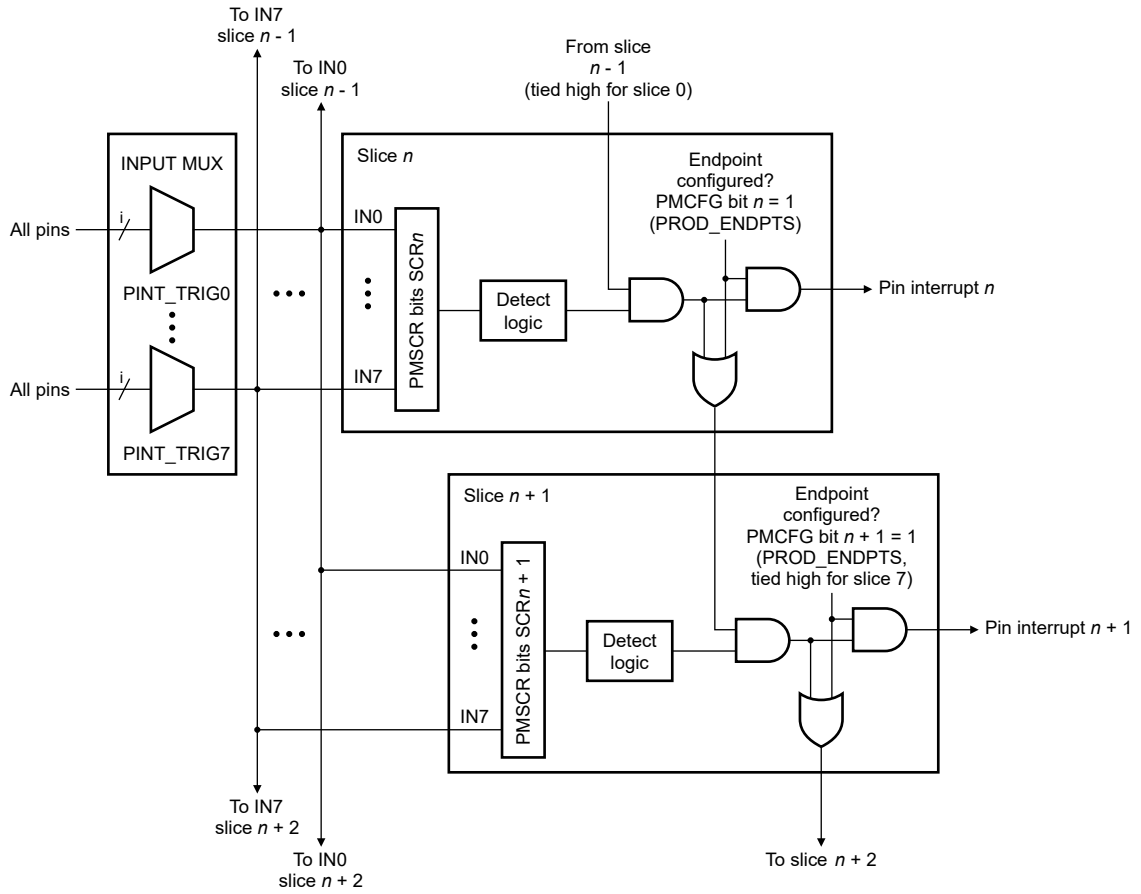


Figure 69. Pattern-match engine connections

The pattern-match logic continuously monitors the 8 inputs and generates interrupts when any one or more minterms (product terms) of the specified Boolean expression match. Each individual minterm generates a separate interrupt request.

It is also possible to enable the pattern-match module to generate RXEV output if any minterm in the Boolean expression matches.

The pattern-match function uses the same 8 interrupt request lines as the pin interrupts. Therefore, in terms of interrupt generation, these two features are mutually exclusive. Control bit generates interrupt requests based on standard pin interrupts or pattern matches. When selecting the pin interrupts, enable the RXEV request to the CPU for pattern matches.

NOTE

Do not use pattern matching to wake the part from deep-sleep mode. Pin interrupts must be selected in order to use the GPIO for wake-up.

8 bit-slice elements make up the pattern-match module. Each bit slice represents a single component in a minterm (product term) within the Boolean expression. Every time the last bit slice for a particular minterm matches, an interrupt request is asserted.

Use the pattern-match capability to create complex software state machines. Each minterm (and its corresponding individual interrupt) represents a different transition event to a new state. Then establish the new set of conditions (that is a new Boolean expression) to make a transition out of the current state.

21.3.2 Clocking

The PINT module has a single clock input, the APB PCLK.

This clock is used for its APB interface, as well as all other peripheral functionality. It must be enabled by the system for register accesses and pattern matching.

21.3.3 Interrupts

The PINT module has 8 interrupt channels, which can be configured by its registers. Each channel has its own interrupt output. Depending on how this IP is integrated, the interrupt outputs can be ORed together to use a single system-level interrupt.

21.4 External signals

PINT_TRIGa determines the inputs to the pin interrupt and pattern-match and engines. See the "INPUTMUX" chapter for more details.

21.5 Initialization

For initialization steps, see chip-specific section.

21.6 Application information

Pattern-Match Interrupt Bit-Slice Source (PMSRC) and Pattern-Match Interrupt Bit Slice Configuration (PMCFG) specify the following expression:

```
IN0 AND NOT IN1 AND IN3 rising edge OR IN1 AND IN2 OR IN0 AND NOT IN3 AND NOT IN4
```

Each term in the Boolean expression, IN0, NOT IN1, IN3 rising edge, and so on, represents one bit slice of the pattern-match engine.

- In the first AND function IN0 AND NOT IN1 AND IN3 rising edge, bit slice 0 monitors for a high level on input IN0, bit slice 1 monitors for a low level on input IN1, and bit slice 2 monitors for a rising edge on input IN3. Detecting this combination of features (all three terms are true) asserts the interrupt associated with bit slice 2.
- In the second AND function IN1 AND IN2, bit slice 3 monitors input IN1 for a high level and bit slice 4 monitors input IN2 for a high level. Detecting this combination asserts the interrupt associated with bit slice 4.
- In the third AND function IN0 AND NOT IN3 AND NOT IN4, bit slice 5 monitors input IN0 for a high level, bit slice 6 monitors input IN3 for a low level, and bit slice 7 monitors input IN4 for a low level. Detecting this combination asserts the interrupt associated with bit slice 7.
- The ORed result of all three AND functions asserts the RXEV request to the CPU. That is, any of the three terms that are true will assert the output.

21.6.1 Pattern-match engine example

Suppose you want to match the Boolean pattern

(IN1) + (IN1 * IN2) + (~IN2 * ~IN3 * IN6fe) + (IN5 * IN7ev)

with:

IN6fe = (sticky) falling edge on input 6

IN7ev = (nonsticky) event (rising or falling edge) on input 7

Each individual term in the above expression is controlled by one bit slice. To specify this expression, program the pattern-match bit slice source and configuration register fields as follows:

- Pattern-Match Interrupt Bit-Slice Source (PMSRC)
  - Because bit slice 5 is used to detect a sticky event on input 6, you can write 1 to SRC5 to clear any preexisting edge detects on bit slice 5.
  - SRC0: 001 - select input 1 for bit slice 0
  - SRC1: 001 - select input 1 for bit slice 1
  - SRC2: 010 - select input 2 for bit slice 2

- SRC3: 010 - select input 2 for bit slice 3
- SRC4: 011 - select input 3 for bit slice 4
- SRC5: 110 - select input 6 for bit slice 5
- SRC6: 101 - select input 5 for bit slice 6
- SRC7: 111 - select input 7 for bit slice 7

- Pattern-Match Interrupt Bit Slice Configuration (PMCFG)
  - PROD_ENDPTS0 = 1
  - PROD_ENDPTS2 = 1
  - PROD_ENDPTS5 = 1
  - All other slices are not product term endpoints and their PMCFG[PROD_ENDPTSn] are 0. Slice 7 is always a product term endpoint and does not have a register bit associated with it.
  - PROD_ENDPTS= 0100101 - bit slices 0, 2, 5, and 7 are product-term endpoints. (Bit slice 7 is an endpoint by default - no associated register bit).
  - CFG0: 000 - high level on the selected input (input 1) for bit slice 0
  - CFG1: 000 - high level on the selected input (input 1) for bit slice 1
  - CFG2: 000 - high level on the selected input (input 2) for bit slice 2
  - CFG3: 101 - low level on the selected input (input 2) for bit slice 3
  - CFG4: 101 - low level on the selected input (input 3) for bit slice 4
  - CFG5: 010 - (sticky) falling edge on the selected input (input 6) for bit slice 5
  - CFG6: 000 - high level on the selected input (input 5) for bit slice 6
  - CFG7: 111 - event (any edge, nonsticky) on the selected input (input 7) for bit slice 7
- Pattern-Match Interrupt Control (PMCTRL)
  - Bit0: Writing this bit selects pattern matches to generate the pin interrupts in place of the standard pin interrupt mechanism.

    For this example, detecting a match on the first product term asserts pin interrupt 0 (which, in this case, is just a high level on input 1).

    Pin interrupt 2 is asserted in response to a match on the second product term.

    Pin interrupt 5 is asserted when there is a match on the third product term.

    Pin interrupt 7 is asserted on a match on the last term.
  - Bit1: Writing this bit causes the RXEV signal to the CPU getting asserted whenever a match occurs on any of the product terms in the expression. Otherwise, the RXEV line does not get used.
  - Bit31:24: At any given time, bits 0, 2, 5, and 7 may be high if the corresponding product terms are currently matching.
  - The remaining bits are low.

21.6.2 Pattern-match engine edge-detect examples

The following figures show only pattern-match functionality in which accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

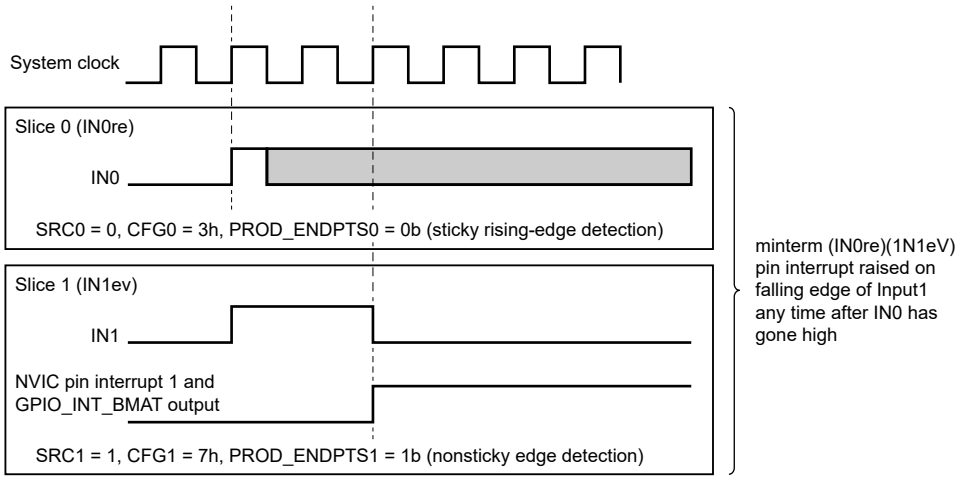


Figure 70. Pattern-match engine examples: sticky edge detect

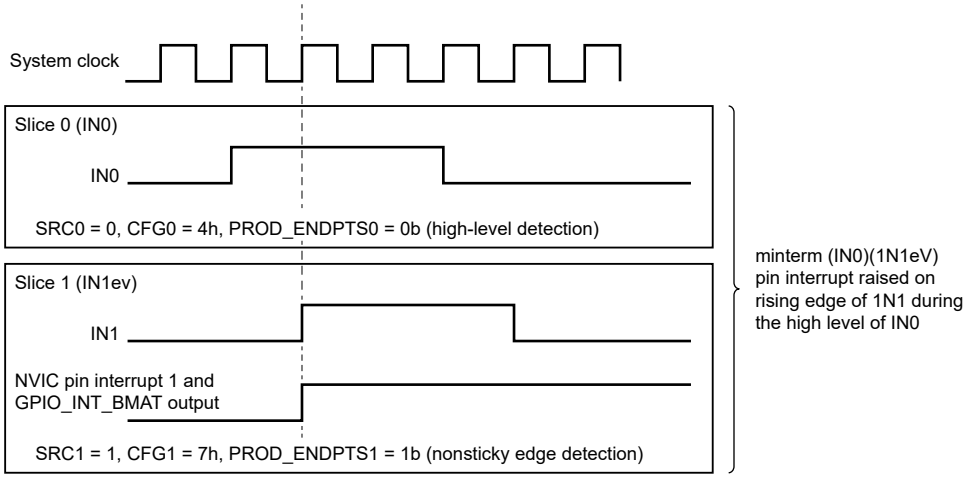


Figure 71. Pattern-match engine examples: Windowed nonsticky edge-detect evaluates as true

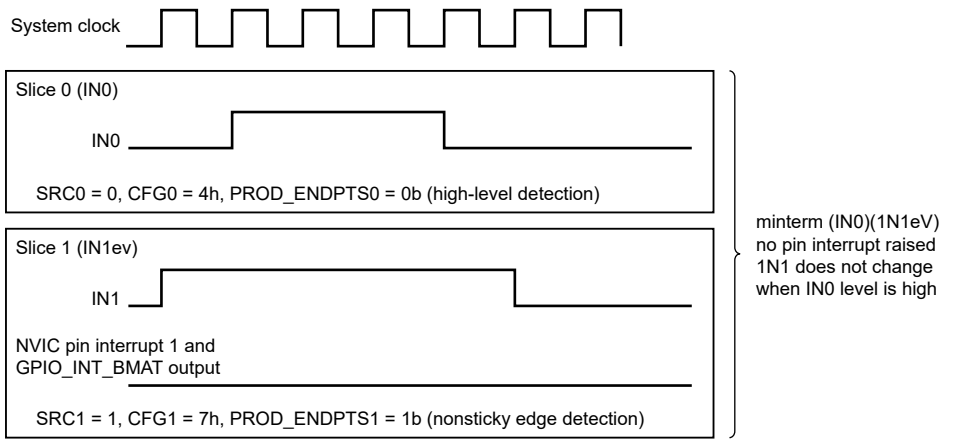


Figure 72. Pattern-match engine examples: Windowed nonsticky edge-detect evaluates as false

21.7 Memory map and register definition

This section includes the PINT memory map and detailed descriptions of all registers.

The chip includes up to 8 pins identified as interrupt sources by Pin Interrupt Select (PINT_TRIGa). The registers have 8 fields that correspond to the pins called out by PINT_TRIG.

Table 186. Pin interrupt registers for edge- and level-sensitive pins

Name	Edge-sensitive function	Level-sensitive function
IENR	Enables rising-edge interrupts.	Enables level interrupts.
SIENR	Write to enable rising-edge interrupts.	Write to enable level interrupts.
CIENR	Write to disable rising-edge interrupts.	Write to disable level interrupts.
IENF	Enables falling-edge interrupts.	Selects active level.
SIENF	Write to enable falling-edge interrupts.	Write to select high active.
CIENF	Write to disable falling-edge interrupts.	Write to select low active.

21.7.1 PINT register descriptions

21.7.1.1 PINT memory map

PINT0 base address: 4002_5000h

PINT1 base address: 4004_5000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Pin Interrupt Mode (ISEL)	32	RW	See section
4h	Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR)	32	RW	See section
8h	Pin Interrupt Level or Rising-Edge Interrupt Set (SIENR)	32	W	See section
Ch	Pin Interrupt Level (Rising-Edge Interrupt) Clear (CIENR)	32	RW	See section
10h	Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF)	32	RW	See section
14h	Pin Interrupt Active Level or Falling-Edge Interrupt Set (SIENF)	32	W	See section
18h	Pin Interrupt Active Level or Falling-Edge Interrupt Clear (CIENF)	32	W	See section
1Ch	Pin Interrupt Rising Edge (RISE)	32	RW	See section
20h	Pin Interrupt Falling Edge (FALL)	32	RW	See section
24h	Pin Interrupt Status (IST)	32	RW	See section
28h	Pattern-Match Interrupt Control (PMCTRL)	32	RW	See section
2Ch	Pattern-Match Interrupt Bit-Slice Source (PMSRC)	32	RW	See section
30h	Pattern-Match Interrupt Bit Slice Configuration (PMCFG)	32	RW	See section

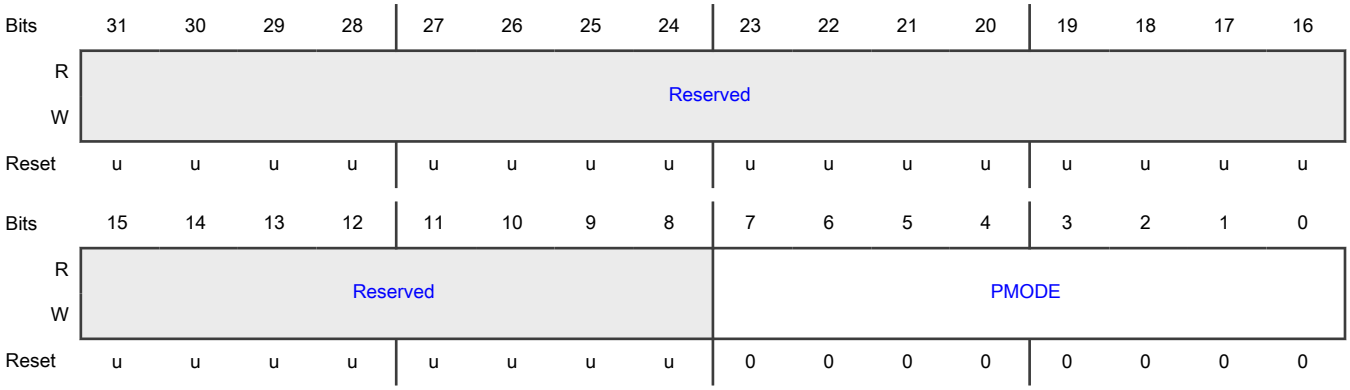
21.7.1.2 Pin Interrupt Mode (ISEL)

Offset

Register	Offset
ISEL	0h

Function
Determines whether an interrupt is edge- or level-sensitive.

Diagram



Fields

Field	Function
31-8 —	Read value is undefined; write only 0.
7-0 PMODE	Interrupt mode Selects the interrupt mode for each pin interrupt. Bit *n* configures the pin interrupt selected in PINT_TRIG in INPUTMUX. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	ISEL	—
PINT1	ISEL[3–0]	ISEL[7–4]

NOTE
The descriptions of the field settings vary by module instance.

Table continued from the previous page...

Field	Function

Instance	Field value and description
PINT0	0000_0000b - In bit n configures the interrupt to be edge-sensitive 0000_0001b - In bit n configures the interrupt to be level-sensitive
PINT1	0000b - In bit n configures the interrupt to be edge-sensitive 0001b - In bit n configures the interrupt to be level-sensitive

21.7.1.3 Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR)

Offset

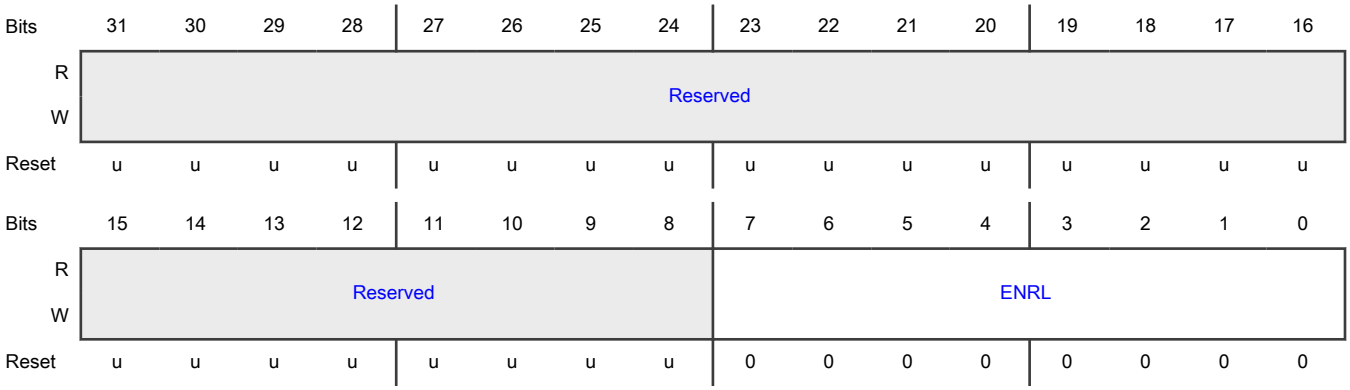
Register	Offset
IENR	4h

Function

Enables the interrupt by each field in Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR), depending on the ISEL[PMODE] configured in Pin Interrupt Mode (ISEL).

- If the PMODE is edge sensitive (PMODE = 0), the rising-edge interrupt is enabled.
- If the PMODE is level sensitive (PMODE = 1), the level interrupt is enabled. Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF) configures the active level (HIGH or LOW) for this interrupt.

Diagram



Fields

Field	Function
31-8 —	Read value is undefined; write only 0.
7-0 ENRL	Enables Interrupt Enables the rising edge or level interrupt for each pin interrupt corresponding to the bit index. Bit *n* configures the pin interrupt selected in PINT_TRIGa. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	IENR	—
PINT1	IENR[3–0]	IENR[7–4]

NOTE
The descriptions of the field settings vary by module instance.

Instance	Field value and description
PINT0	0000_0000b - In bit n disables the corresponding interrupt 0000_0001b - In bit n enables the corresponding interrupt
PINT1	0000b - In bit n disables the corresponding interrupt 0001b - In bit n enables the corresponding interrupt

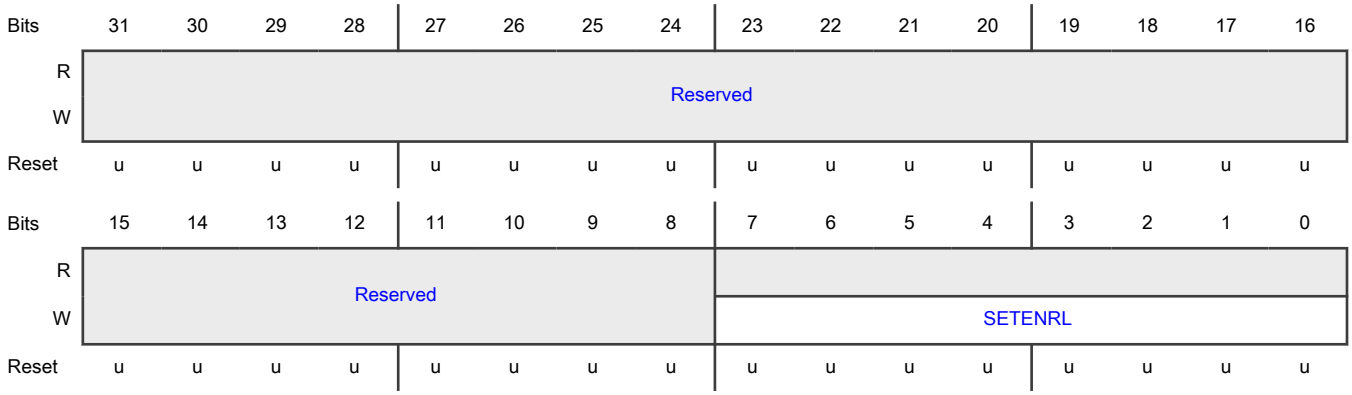
21.7.1.4 Pin Interrupt Level or Rising-Edge Interrupt Set (SIENR)

Offset

Register	Offset
SIENR	8h

Function
Configures the corresponding field in Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR) by each field in Pin Interrupt Level or Rising-Edge Interrupt Set (SIENR), depending on the ISEL[PMODE] configured in Pin Interrupt Mode (ISEL).

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SETENRL	Configures IENR Configures IENR by writing 1s to this address. Bit *n* configures the corresponding bit in IENR. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	SIENR	—
PINT1	SIENR[3–0]	SIENR[7–4]

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
PINT0	0000_0000b - No operation for interrupt n 0000_0001b - Enable rising edge or level interrupt for interrupt n
PINT1	0000b - No operation for interrupt n 0001b - Enable rising edge or level interrupt for interrupt n

21.7.1.5 Pin Interrupt Level (Rising-Edge Interrupt) Clear (CIENR)

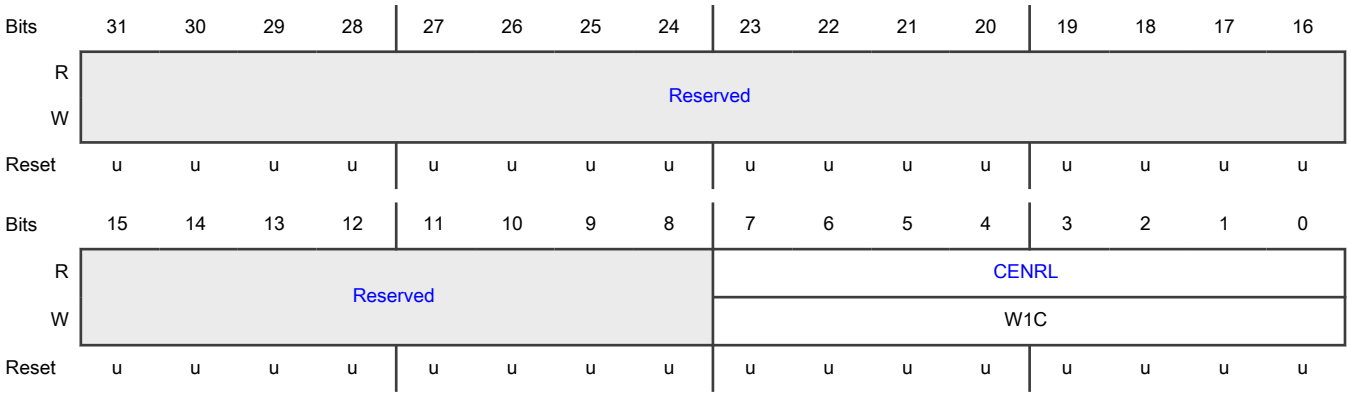
Offset

Register	Offset
CIENR	Ch

Function

Writing 0 to each bit in Pin Interrupt Level (Rising-Edge Interrupt) Clear (CIENR) adjusts the corresponding bit in Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR) depending on the ISEL[PMODE] configured in Pin Interrupt Mode (ISEL). Writing 0 to this address configures bits in Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR), effectively disabling interrupts. Writing 0 to bit *a* configures bit *n* in Pin Interrupt Level or Rising-Edge Interrupt Enable (IENR).

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 CENRL	Clear bits in IENR **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	CIENR	—
PINT1	CIENR[3–0]	CIENR[7–4]

NOTE

The descriptions of the field settings vary by module instance.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>PINT0</td><td>0000_0000b - No operation 0000_0001b - Disable rising edge or level interrupt</td></tr><tr><td>PINT1</td><td>0000b - No operation 0001b - Disable rising edge or level interrupt</td></tr></table>

21.7.1.6 Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF)

Offset

Register	Offset
IENF	10h

Function

Enables the falling-edge interrupt or configures the level sensitivity depending on the ISEL[PMODE] configured in Pin Interrupt Mode (ISEL).

- If the PMODE is edge sensitive (PMODE = 0), the falling-edge interrupt is enabled.
- If the PMODE is level sensitive (PMODE = 1), the active level of the level interrupt (HIGH or LOW) is configured.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ENAF							
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8	Reserved

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>7-0
ENAF</td><td>Enables Interrupt
Enables the falling edge or configures the active-level interrupt for each pin interrupt. Bit a configures the pin interrupt selected in PINT_TRIGa.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>PINT0</td><td>IENF</td><td>—</td></tr>
<tr><td>PINT1</td><td>IENF[3–0]</td><td>IENF[7–4]</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>PINT0</td><td>0000_0000b - Disable (set active interrupt level LOW)
0000_0001b - Enable (set active interrupt level HIGH)</td></tr>
<tr><td>PINT1</td><td>0000b - Disable (set active interrupt level LOW)
0001b - Enable (set active interrupt level HIGH)</td></tr>
</table>
</td></tr>
</table>

21.7.1.7 Pin Interrupt Active Level or Falling-Edge Interrupt Set (SIENF)

Offset

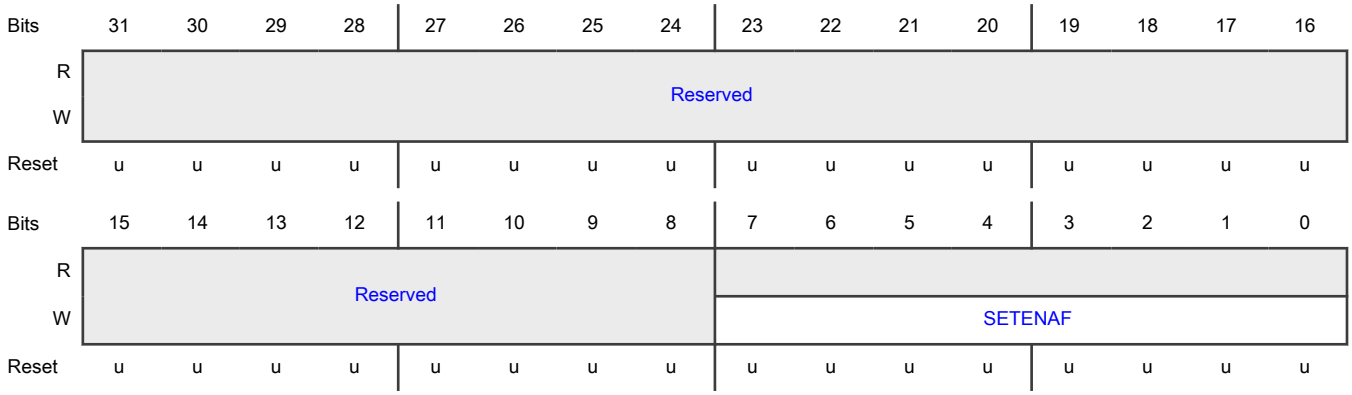
Register	Offset
SIENF	14h

Function

Sets the corresponding bit in Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF) (by each bit of SIENF), depending on the PMODE configured in Pin Interrupt Mode (ISEL).

- If the PMODE is edge sensitive (PMODE = 0), the falling-edge interrupt becomes 1.
- If the PMODE is level sensitive (PMODE = 1), the HIGH-active interrupt is selected.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SETENAF	Write 1 to this address to clear Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF) to disable interrupts. Bit *a* sets bit *n* in IENF. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	SIENF	—
PINT1	SIENF[3–0]	SIENF[7–4]

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
PINT0	0000_0000b - Writes 0 to IENF. No operation 0000_0001b - Select HIGH-active interrupt or enable falling-edge interrupt
PINT1	0000b - Writes 0 to IENF. No operation 0001b - Select HIGH-active interrupt or enable falling-edge interrupt

21.7.1.8 Pin Interrupt Active Level or Falling-Edge Interrupt Clear (CIENF)

Offset

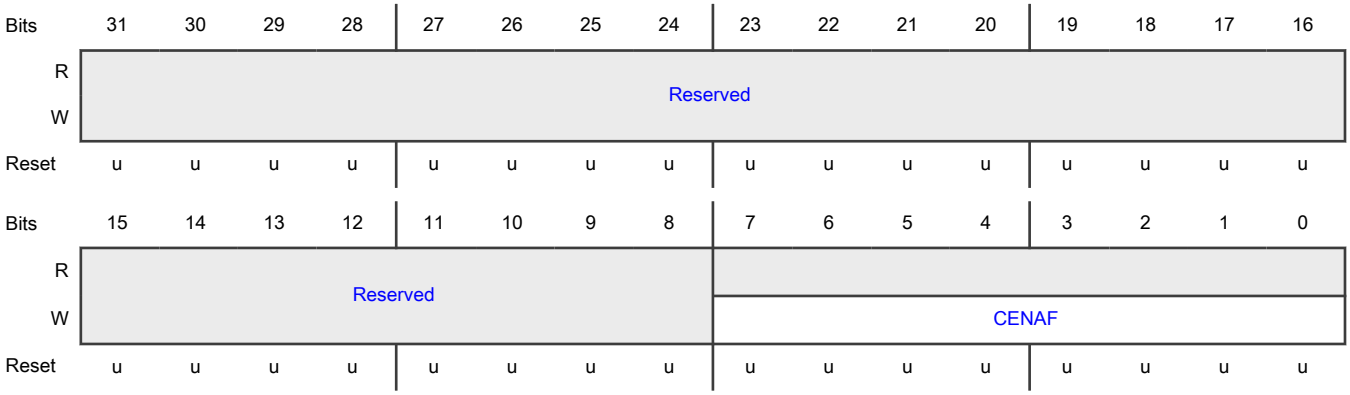
Register	Offset
CIENF	18h

Function

Sets the corresponding bit in Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF) (by each bit of Pin Interrupt Active Level or Falling-Edge Interrupt Clear (CIENF), depending on the ISEL[PMODE] configured in Pin Interrupt Mode (ISEL).

- If the PMODE is edge sensitive (PMODE = 0), the falling-edge interrupt is cleared.
- If the PMODE is level sensitive (PMODE = 1), the LOW-active interrupt is selected.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 CENAF	Writes 0 to IENF Specifies that if you change Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF) to 0 by writing 1 to it,this will disable interrupts. Writing 0 to bit *a* configures bit *n* in Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF). **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	CIENF	—
PINT1	CIENF[3–0]	CIENF[7–4]

Table continued from the previous page...

Field	Function
	Instance / **Field supported in** / **Field not supported in** **NOTE** The descriptions of the field settings vary by module instance.

Instance	Field value and description
PINT0	0000_0000b - No operation 0000_0001b - LOW-active interrupt selected or falling-edge interrupt disabled
PINT1	0000b - No operation 0001b - LOW-active interrupt selected or falling-edge interrupt disabled

21.7.1.9 Pin Interrupt Rising Edge (RISE)

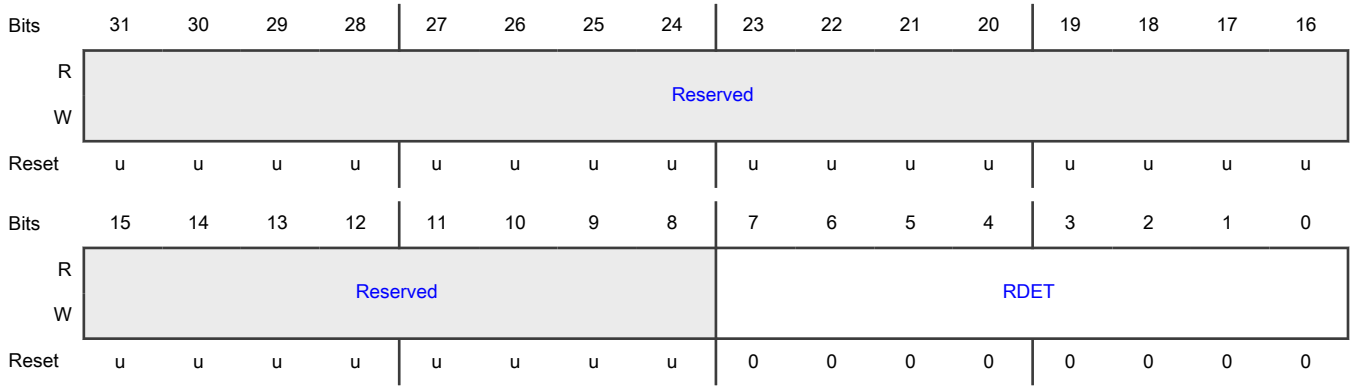
Offset

Register	Offset
RISE	1Ch

Function

Contains 1s for pin interrupts selected in PINT_TRIGa (see the "INPUTMUX" chapter for more details) on which a rising edge has been detected. Writing 1s to this register clears rising-edge detection. Ones in this register assert an interrupt request for pins that are enabled for rising-edge interrupts. All edges are detected for all pins selected by PINT_TRIG, regardless of whether they are interrupt-enabled.

Diagram



Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-8
—</td><td>Reserved</td></tr>
<tr><td>7-0
RDET</td><td>Rising-Edge Detect
Bit a detects the rising edge of the pin selected in PINT_TRIGa.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>PINT0</td><td>RISE</td><td>—</td></tr>
<tr><td>PINT1</td><td>RISE[3–0]</td><td>RISE[7–4]</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>PINT0</td><td>0000_0000b - Read 0- No rising edge (since Reset or you wrote a 1 to this field last time), Write 0- No operation
0000_0001b - Read 1- Rising edge (since Reset or you wrote a 1 to this field last time), Write 1- Clear rising-edge detection for this pin</td></tr>
<tr><td>PINT1</td><td>0000b - Read 0- No rising edge (since Reset or you wrote a 1 to this field last time), Write 0- No operation
0001b - Read 1- Rising edge (since Reset or you wrote a 1 to this field last time), Write 1- Clear rising-edge detection for this pin</td></tr>
</table></td></tr>
</table>

21.7.1.10 Pin Interrupt Falling Edge (FALL)

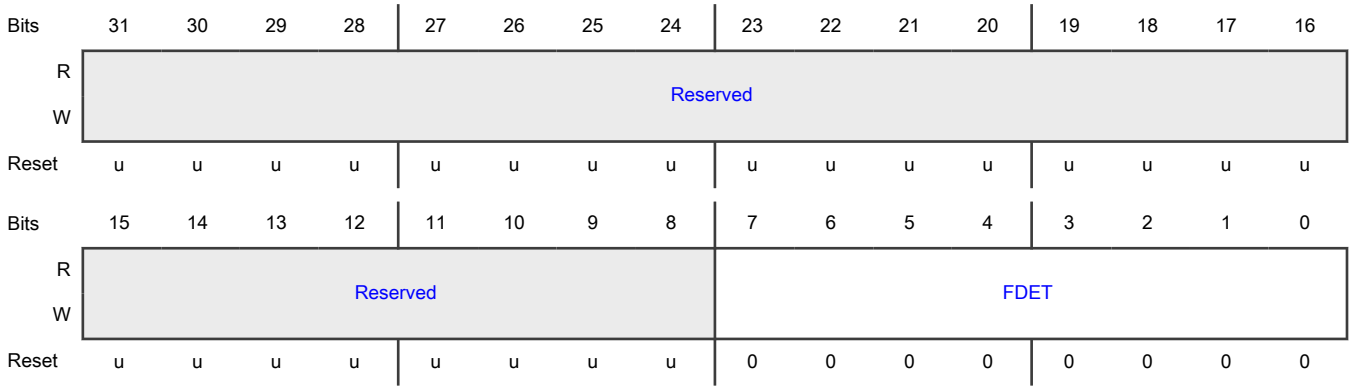
Offset

Register	Offset
FALL	20h

Function

Contains 1s for pin interrupts selected in PINT_TRIGa on which a falling edge has been detected. Writing 1s to this register clears falling-edge detection. The values (1s) in this register assert interrupt requests for pins with falling-edge interrupt capabilities. PINT_TRIGa detects all edges for all pins, regardless of whether they are interrupt-enabled.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 FDET	Falling-Edge Detect Bit *a* detects the falling edge of the pin selected in PINT_TRIGa. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	FALL	—
PINT1	FALL[3–0]	FALL[7–4]

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
PINT0	0000_0000b - Read 0- No falling edge (since Reset or you wrote a 1 to this field last time), Write 0- No operation 0000_0001b - Read 1- Falling edge (since Reset or you wrote a 1 to this field last time), Write 1- Clear falling-edge detection for this bit
PINT1	0000b - Read 0- No falling edge (since Reset or you wrote a 1 to this field last time), Write 0- No operation 0001b - Read 1- Falling edge (since Reset or you wrote a 1 to this field last time), Write 1- Clear falling-edge detection for this bit

21.7.1.11 Pin Interrupt Status (IST)

Offset

Register	Offset
IST	24h

Function

Returns 1s for pin interrupts that are currently requesting an interrupt. For edge-sensitive pins in Interrupt Select, writing 1s to this register clears both rising- and falling-edge detection for the pin. For level-sensitive pins, writing 1s inverts the corresponding bit in the active-level register, thus switching the active level on the pin.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PSTAT							
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 PSTAT	Pin Interrupt Status Bit *a* returns the status, clears the edge interrupt, or inverts the active level of the pin selected in PINT_TRIGa. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PINT0	IST	—
PINT1	IST[3–0]	IST[7–4]

NOTE

The descriptions of the field settings vary by module instance.

Table continued from the previous page...

Field	Function	
	Instance	Field value and description
	PINT0	0000_0000b - Read 0- Interrupt is not requested, Write 0- No operation 0000_0001b - Read 1- Interrupt is requested, Write 1 (edge-sensitive)- clear rising- and falling-edge detection for this pin, Write 1 (level-sensitive)- switch the active level for this pin in Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF)
	PINT1	0000b - Read 0- Interrupt is not requested, Write 0- No operation 0001b - Read 1- Interrupt is requested, Write 1 (edge-sensitive)- clear rising- and falling-edge detection for this pin, Write 1 (level-sensitive)- switch the active level for this pin in Pin Interrupt Active Level or Falling-Edge Interrupt Enable (IENF)

21.7.1.12 Pattern-Match Interrupt Control (PMCTRL)

Offset

Register	Offset
PMCTRL	28h

Function

Contains a field to select pattern-match interrupt generation (as opposed to pin interrupts which share the same interrupt request lines), and another to enable the RXEV output to the CPU. Read the current state of the pattern match in this register. If the pattern-match feature is not used (either for interrupt generation or for RXEV assertion), write 0 to PMCTRL[SEL_PMATCH] and PMCTRL[ENA_RXEV] of this register to conserve power.

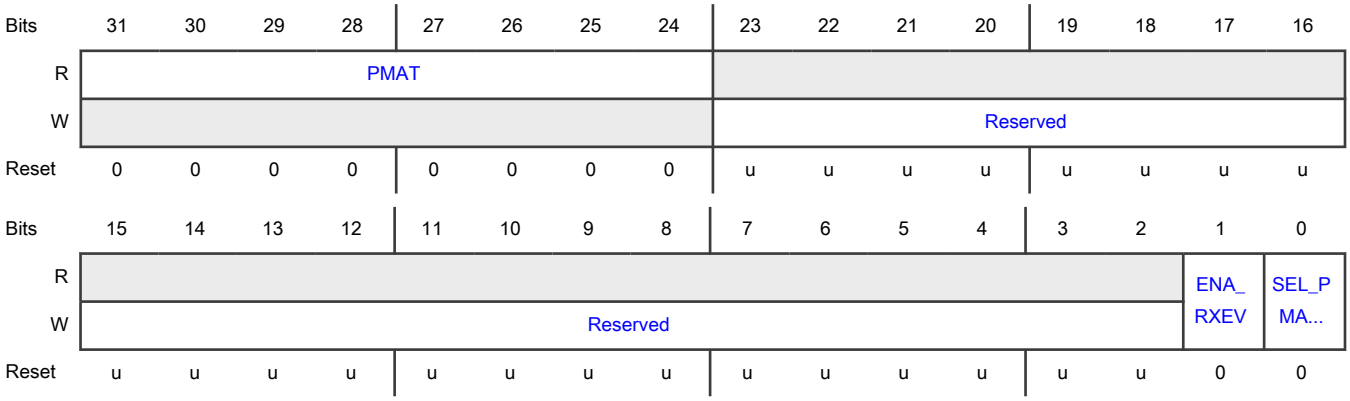
NOTE

Set up the pattern-match configuration in Pattern-Match Interrupt Bit-Slice Source (PMSRC) and Pattern-Match Interrupt Bit Slice Configuration (PMCFG) before writing to this register to enable (or reenable) the pattern-match functionality. This prevents spurious interrupts from occurring during the feature"s activation.

NOTE

The pattern-match feature requires clocks to operate and thus not generate an interrupt or wake the chip during reduced power modes below sleep mode.

Diagram



Fields

Field	Function
31-24 PMAT	Pattern Matches Displays the current state of pattern matches. 0000_0001b - The corresponding product term is matched by the current state of the appropriate inputs
23-2 —	Reserved, will read 0"s
1 ENA_RXEV	Enables the RXEV output to the CPU and/or to a GPIO output, when the specified Boolean expression evaluates to true. If this value is 0b, RXEV output to the CPU is disabled. If this value is 1b, RXEV output to the CPU is enabled. 0b - Disabled 1b - Enabled
0 SEL_PMATCH	Specifies whether the pin interrupts are controlled by the pin interrupt function or by the pattern-match function. If this value is 0b, interrupts are driven in response to the standard pin interrupt function. If this value is 1b, interrupts are driven in response to pattern matches. 0b - Pin interrupt 1b - Pattern match

21.7.1.13 Pattern-Match Interrupt Bit-Slice Source (PMSRC)

Offset

Register	Offset
PMSRC	2Ch

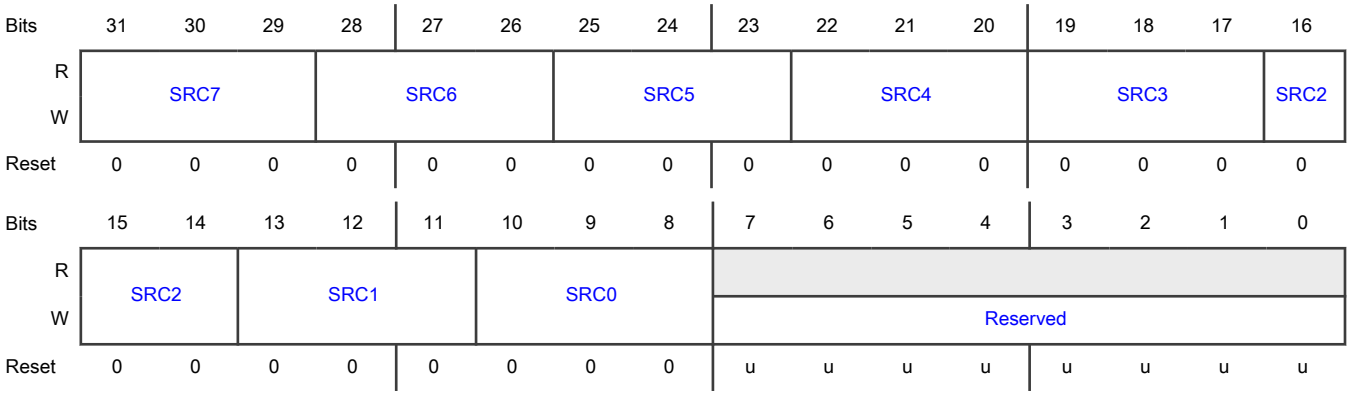
Function

Specifies the input source for each of the pattern-match bit slices. Each of the possible 8 inputs is selected in the pin interrupt select registers in INPUTMUX. Input 0 corresponds to the pin selected in PINT_TRIG0, input 1 corresponds to the pin selected in PINT_TRIG1, and so on.

NOTE

Writing any value to either Pattern-Match Interrupt Bit Slice Configuration (PMCFG) or Pattern-Match Interrupt Bit-Slice Source (PMSRC), or disabling the pattern-match feature by clearing both the PMCTRL[SEL_PMATCH] and PMCTRL[ENA_RXEV] bits in Pattern-Match Interrupt Control (PMCTRL) to zeros will erase all edge-detect history.

Diagram



Fields

Field	Function
31-29: SRC7 28-26: SRC6 25-23: SRC5 22-20: SRC4 19-17: SRC3 16-14: SRC2 13-11: SRC1 10-8: SRC0	Selects the input source for bit slice n 000b - Input 0 (selects the pin identified in PINT_TRIG0) 001b - Input 1 (selects the pin identified in PINT_TRIG1) 010b - Input 2 (selects the pin identified in PINT_TRIG2) 011b - Input 3 (selects the pin identified in PINT_TRIG3) 100b - Input 4 (selects the pin identified in PINT_TRIG4) 101b - Input 5 (selects the pin identified in PINT_TRIG5) 110b - Input 6 (selects the pin identified in PINT_TRIG6) 111b - Input 7 (selects the pin identified in PINT_TRIG7)
7-0 —	Reserved, will read 0's

21.7.1.14 Pattern-Match Interrupt Bit Slice Configuration (PMCFG)

Offset

Register	Offset
PMCFG	30h

Function

Configures the detect logic and contains bits to select from among eight alternative conditions for each bit slice that cause that bit slice to contribute to a pattern match. The seven LSBs of this register specify which bit slices are the end-points of product terms in the Boolean expression (where OR terms are to be inserted in the expression).

Two types of edge detection on each input are possible:

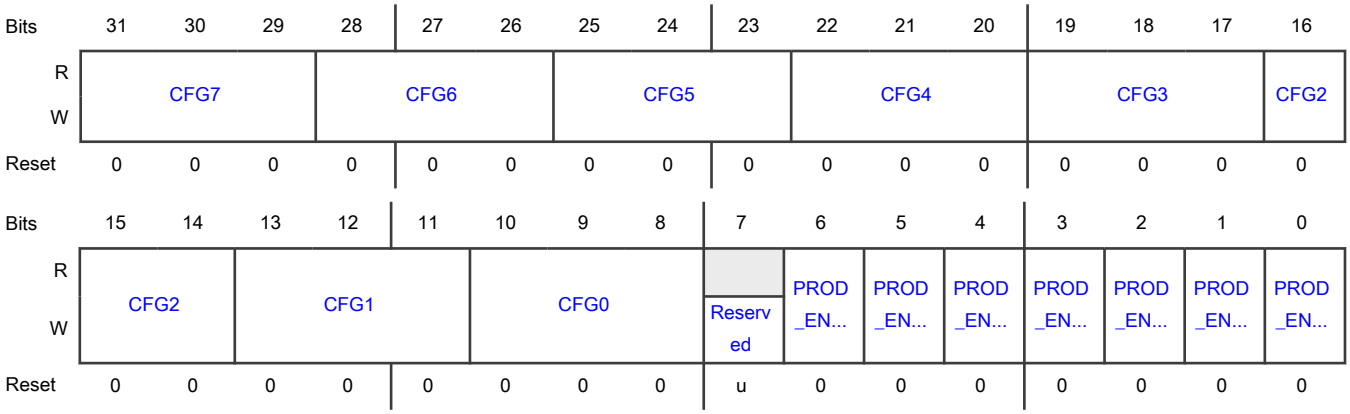
- Sticky: A rising edge, a falling edge, or a rising or falling edge that is detected at any time after the edge-detection mechanism becomes 0. The input qualifies as detected (the detect logic output remains HIGH) until the pattern-match engine detect logic becomes 0 again.
- Nonsticky: In the event of an edge (rise or fall), the detect logic output for this pin goes HIGH. One clock cycle clears this bit, allowing the edge detector to detect a second edge.

To clear the pattern-match engine detect logic, write any value to either Pattern-Match Interrupt Bit Slice Configuration (PMCFG) or Pattern-Match Interrupt Bit-Slice Source (PMSRC), or disable the pattern-match feature by changing both PMCTRL[SEL_PMATCH] and PMCTRL[ENA_RXEV] to 0 by writing 1 to Pattern-Match Interrupt Control (PMCTRL). This will erase all edge-detect history.

To select whether a slice marks the final component in a minterm of the Boolean expression, write a 1 in the corresponding PROD_ENDPTSa bit. Setting a term as the final component has two effects:

1. Anytime a match to that product term occurs, this leads to an assertion of interrupt request associated with this bit slice.
2. The next bit slice will start a new, independent product term in the Boolean expression. The Boolean expression will include an OR following the element controlled by this bit slice.

Diagram



Fields

Field	Function
31-29: CFG7 28-26: CFG6 25-23: CFG5 22-20: CFG4 19-17: CFG3 16-14: CFG2 13-11: CFG1	Match Configuration Specifies the match contribution condition for bit slice n: • If the value is 000b, this bit slice always contributes to a product term match. • If the value is 001b, edge detection for this bit slice is clear when there has been a rising edge on the specified input since last clearing. In order to clear this match condition, you must write to Pattern-Match Interrupt Bit Slice Configuration (PMCFG) or Pattern-Match Interrupt Bit-Slice Source (PMSRC).

Table continues on the next page...

Table continued from the previous page...

Field	Function
10-8: CFG0	• If the value is 010b, edge detection for this bit slice is clear when there has been a falling edge on the specified input since last clearing. In order to clear this match condition, you must write to Pattern-Match Interrupt Bit Slice Configuration (PMCFG) or Pattern-Match Interrupt Bit-Slice Source (PMSRC). • If the value is 011b, edge detection for this bit slice is clear when there has been either a rising or a falling edge on the specified input since last clearing. In order to clear this match condition, you must write to Pattern-Match Interrupt Bit Slice Configuration (PMCFG) or Pattern-Match Interrupt Bit-Slice Source (PMSRC). • If the value is 100b, match for this bit slice occurs when there is a high level on the input specified for this bit slice in Pattern-Match Interrupt Bit-Slice Source (PMSRC). • If the value is 101b, match occurs when there is a low level on the specified input. • If the value is 110b, it never contributes to a match and must be used to disable any unused bit slices. • If the value is 111b, a match occurs when the specified input first detects a rising or falling edge. This is a nonsticky version of value 0 x 3. Write 0 to this field after one clock cycle. 000b - Constant HIGH 001b - Sticky rising edge 010b - Sticky falling edge 011b - Sticky rising or falling edge 100b - High level 101b - Low level 110b - Constant 0 111b - Event (Nonsticky rising or falling edge)
7 ——	Bit slice 7 is automatically considered a product endpoint.
6-0 PROD_ENDPTSn	Determines whether slice n is an endpoint. Slice n is not an endpoint. Slice n is the endpoint of a product term (minterm). Pin interrupt n in the NVIC is raised if the minterm evaluates as true. 0b - No effect 1b - Endpoint

Chapter 22
General-Purpose Input/Output (GPIO)

22.1 Chip-specific GPIO information

Table 187. Reference links to related information

Topic	Related module	Reference
Full description	GPIO	GPIO
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Power management	PMC	Power overview PMC
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

22.1.1 Module instances

This device has 11 instances of the GPIO module: GPIO0 to GPIO10.

Table 188. GPIO instances

	GPIO0-GPIO7	GPIO8-GPIO10
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, eDMA2, eDMA3

22.1.2 GPIO pins and interrupts

Each GPIO instance supports GPIO configuration of 32 pins. Each GPIO function is connected to the pin via IOPCTL. Not all the GPIO pins are available, see Pin assignments for GPIO pins available for each package.

GPIO0-7 are used to interrupt CPU0 and GPIO8-10 are for CPU1 interrupt. The edge and level interrupts are supported even as a wake-up source of the sleep and deep-sleep modes. Each GPIO interrupt can be configured to channel 0 or 1.

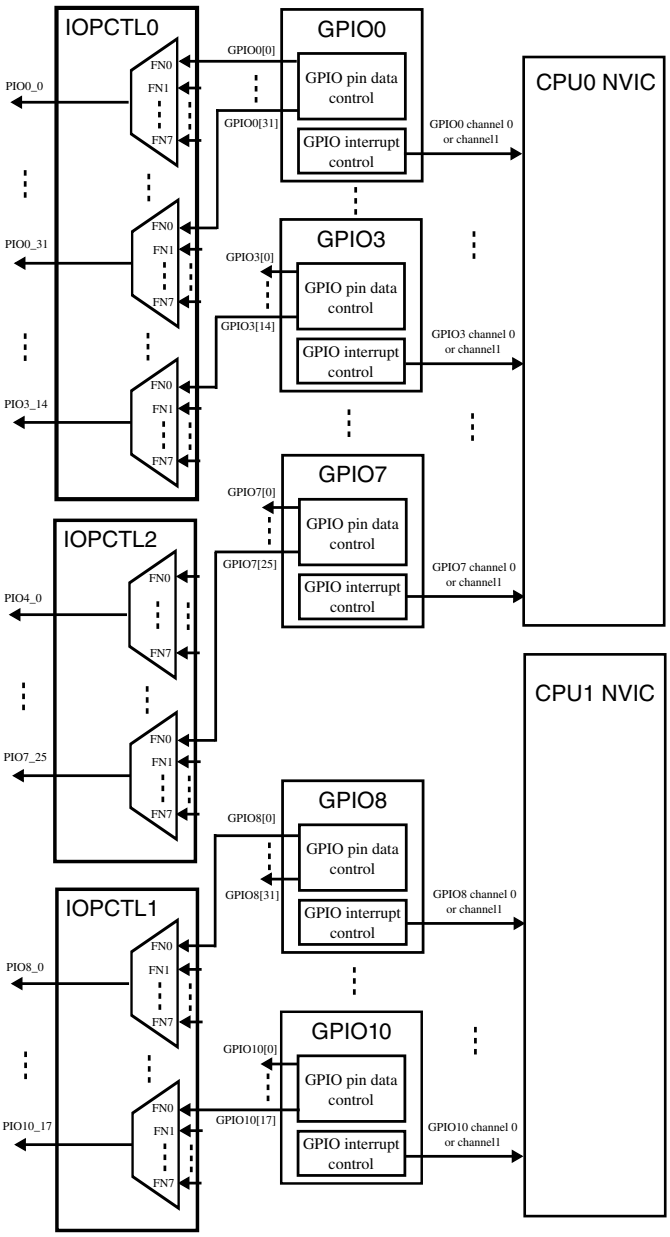


Figure 73. GPIO pins and interrupts

22.1.3 Security considerations

The GPIO module implements Access protection that can be used to configure secure/non-secure and privileged/non-privileged access on a per pin basis. The interrupt, DMA, or trigger functionality for each pin can also be configured for secure/non-secure and privileged/non-privileged access.

Each GPIO module is also instantiated to use two module slots (for example, GPIO0 and GPIO0_alias). At the Secure AHB controller, one slot can be configured for secure access while the other is configured for non-secure access. This allows each GPIO module to be accessed by both secure and non-secure masters using the appropriate slot. While the Secure AHB controller's MISC_CTRL_REG[DISABLE_STRICT] option is configured for strict mode. Each GPIO module's pin and interrupt, DMA, or trigger secure/non-secure access protections are then used to restrict access according to the level used by that particular GPIO slot.

22.2 Overview

GPIO communicates to the processor core via a zero wait-state interface for maximum pin performance.

22.2.1 Block diagram

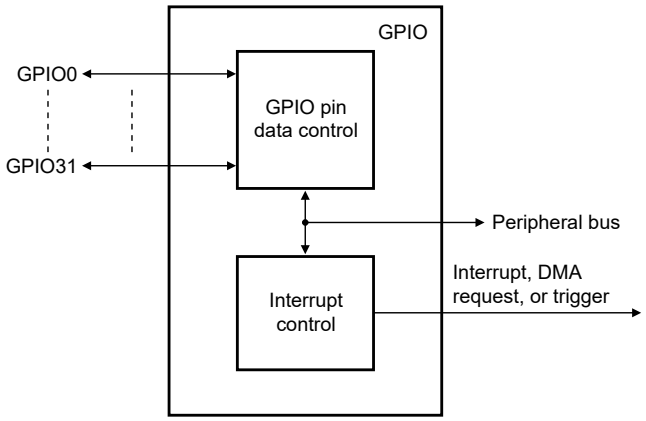


Figure 74. Block diagram

22.2.2 Features

- Port Data Input (PDIR) displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt modules for that pin are enabled.
- Port Data Output (PDOR), with corresponding set, clear, and toggle registers, controls the output data of each pin when the pin is configured for the GPIO function.
- Port Data Direction (PDDR) controls the direction of each pin when the pin is configured for the GPIO function.
- Port Input Disable (PIDR) controls disabling of the input for each general-purpose pin.
- Pin interrupts:
  - Interrupt flags and enable registers for each pin are functional in all digital pin muxing modes.
  - Support for interrupt, peripheral trigger, or DMA request is configured for each pin.
  - Support for edge-sensitive (rising, falling, or both) or level-sensitive (low, high) interrupts is configured for each pin.
  - Asynchronous wake-up in Low-Power mode.
  - GPIO generates a total of 2 interrupts, 2 output triggers, and 2 DMA requests.
  - Each pin can be used to generate a single interrupt, output trigger, or DMA request.
- Protection registers:
  - Each pin is configured for secure or nonsecure and privileged or nonprivileged access.
  - Each interrupt, trigger, and DMA request domain is configured for secure or nonsecure and privileged or nonprivileged access.

22.3 Functional description

22.3.1 Low-Power mode

You can configure GPIO to exit Low-Power mode via an asynchronous wake-up signal if an enabled interrupt is detected.

22.3.2 Debug mode

GPIO remains functional in Debug mode.

22.3.3 General-purpose input

The logic state of each pin is available via Port Data Input (PDIR) if:

- The corresponding field in Port Input Disable (PIDR) is 0.
- The pin is configured for a digital function.

22.3.4 General-purpose output

The logic state of each pin is controlled via Port Data Output (PDOR) and Port Data Direction (PDDR), provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input or output.

Table 189. General-purpose output

If	Then
A pin is configured for the GPIO function and the corresponding PDDR field is 0	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding PDDR field is 1	The pin is configured as an output and the logic state of the pin is equal to the corresponding PDOR field.

For efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers allow one or more outputs within one port to be set, cleared, or toggled from a single register write, eliminating the need for a read-modify-write operation to prevent changing pins accidentally.

22.3.5 Clocking

GPIO receives a single clock, which is used for register access and synchronization with external pin inputs. There are no special considerations.

22.3.6 Reset

GPIO receives a single reset, which resets the peripheral. There are no special considerations.

22.3.7 External interrupts

The external interrupt capability of GPIO is available in all digital pin muxing modes.

GPIO generates a total of 2 interrupts, 2 output triggers, and 2 DMA requests. You can configure each pin individually for interrupt, output trigger, or DMA requests:

- Each output implements a separate interrupt status flag (ISF) register for that domain.
- Each output generates a single interrupt.
- Each output generates a single DMA request.
- Each output generates a single peripheral trigger output.

You can configure each pin individually for any of the external interrupt modes shown in the following table.

Table 190. Available pin configurations for external interrupts

Signal conditions	Software polling using flags	Peripheral triggers	Interrupts	DMA requests
Rising-edge	Yes	—	Yes	Yes
Falling-edge	Yes	—	Yes	Yes
Rising- and falling-edge	Yes	—	Yes	Yes
High-level	—	Yes	Yes	—
Low-level	—	Yes	Yes	—

The interrupt status flag is set when the configured edge or level is detected on the pin. Unless GPIO is in Low-Power mode, the input is first synchronized to the system clock to detect the configured level or edge transition.

GPIO generates a pin interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that output. The interrupt negates after the interrupt status flags for all enabled interrupts are cleared by writing a logic 1 to either ISFR0[ISF*n*] or ICR0[ISF].

GPIO generates a DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that output. The DMA request negates after the DMA transfer is completed because that clears the interrupt status flags for all enabled DMA requests.

In Low-Power mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wake-up signal to exit Low-Power mode.

GPIO generates a peripheral trigger output that asserts if any pin configured for the active-high trigger is logic 1, or any pin triggered for the active-low trigger is logic 0. The peripheral trigger output asynchronously updates from the value on the configured pins.

22.3.8 Global interrupt control

The two global interrupt control registers (Global Interrupt Control Low (GICLR) and Global Interrupt Control High (GICHR)) allow a single register write to update the upper 16 bits of Interrupt Control a (ICR0 - ICR31) for up to 16 pins, all with the same value.

These global interrupt control registers allow you to quickly configure multiple pins within the same port with the same interrupt configuration.

The global interrupt control registers are write-only registers and always read 0.

22.3.9 DMA

GPIO generates 2 requests. For each pin, you can configure DMA requests to assert for either rising, falling, or both edge detection. See External interrupts for more information.

22.3.10 Access protection

GPIO implements access protection registers (PCNS and PCNP) for each pin, as shown in Table 191.

Table 191. PCNS and PCNP register access protection implementation

Access	Description
Secure	You can write to or read the pins configured for secure access only in the Secure state.
Nonsecure	You can write to or read the pins configured for nonsecure access only in the Nonsecure state.

Table continues on the next page...

Table 191. PCNS and PCNP register access protection implementation (continued)

Access	Description
Privilege	You can write to the pins configured for privilege access only in the Privilege state.
Nonprivilege	You can write to the pins configured for nonprivilege access in both the Privilege and Nonprivilege states.

GPIO implements access protection registers (ICNS and ICNP) for each interrupt, trigger output, and DMA request as shown in Table 192.

Table 192. ICNS and ICNP register access protection implementation

Access	Description
Secure	You can configure the outputs programmed for secure access only in the Secure state.
Nonsecure	You can configure the outputs programmed for nonsecure access only in the Nonsecure state.
Privilege	You can configure the outputs programmed for privilege access only in the Privilege state.
Nonprivilege	You can configure the outputs programmed for nonprivilege access in both the Privilege and Nonprivilege states.

You can write to the access protection registers (PCNS, PCNP, ICNS, and ICNP) only in the Secure-Privilege state; you can optionally lock these registers until the next reset.

Configuring a pin interrupt, trigger output, or DMA request requires the following access permissions:

- Access permission to configure the pin.
- Access permission to either configure to the desired interrupt, trigger output, or DMA request, or writing 0 to ICR*n*[IRQC].
- Access permission to either configure with the existing interrupt, trigger output, or DMA request, or if the current value of ICR*n*[IRQC] is 0.
- The interrupt configuration is not locked.

22.4 External signals

Table 193. External signals

Signal	Description		Direction
GPIO31–GPIO0	General-purpose input/output		I/O
	State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.	
	Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.	

NOTE

Not all pins within each GPIO are implemented on each chip. See the "Signal Multiplexing" chapter for the number of GPIO pins within each port available on this chip.

22.5 Initialization

To initialize GPIO:

1. Initialize the GPIO pins for the output function:
    a. Configure the output logic value for each pin by using Port Data Output (PDOR).
    b. Configure the direction for each pin by using Port Data Direction (PDDR).
2. Initialize the interrupt function by writing to Interrupt Control a (ICR0 - ICR31) for the corresponding pins and desired configuration. If the pin is previously used for a different function, first write 0100_0000h to Interrupt Control a (ICR0 - ICR31) to disable the previous function and clear the flag.

22.6 Application information

GPIO includes the following applications:

- Reading the state of a single pin by performing a byte read of Pin Data (P0DR - P31DR).
- Updating the state of a single pin by performing a byte write to Pin Data (P0DR - P31DR).
- Reading the state of multiple pins by reading Port Data Input (PDIR).
- Updating the state of multiple pins by using the following ways:
    — Writing to Port Data Output (PDOR).
    — Writing to Port Set Output (PSOR) to write 1 to Port Data Output (PDOR).
    — Writing to Port Clear Output (PCOR) to write 0 to Port Data Output (PDOR).
    — Writing to Port Toggle Output (PTOR) to toggle Port Data Output (PDOR).

22.7 Memory map and register definition

The GPIO registers support 8-bit, 16-bit, or 32-bit accesses. Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's register appears with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. See the "Signal Multiplexing" chapter for the exact control bits of each port.

22.7.1 GPIO register descriptions

22.7.1.1 GPIO memory map

GPIO0 base address: 4010_0000h

GPIO0_ALIAS base address: 4010_1000h

GPIO1 base address: 4010_2000h

GPIO1_ALIAS base address: 4010_3000h

GPIO2 base address: 4010_4000h

GPIO2_ALIAS base address: 4010_5000h

GPIO3 base address: 4010_6000h

GPIO3_ALIAS base address: 4010_7000h

GPIO4 base address: 4010_8000h

GPIO4_ALIAS base address: 4010_9000h

GPIO5 base address: 4010_A000h

GPIO5_ALIAS base address: 4010_B000h

GPIO6 base address: 4010_C000h

GPIO6_ALIAS base address: 4010_D000h

GPIO7 base address: 4010_E000h

GPIO7_ALIAS base address: 4010_F000h

GPIO8 base address: 4032_0000h

GPIO8_ALIAS base address: 4032_1000h

GPIO9 base address: 4032_2000h

GPIO9_ALIAS base address: 4032_3000h

GPIO10 base address: 4032_4000h

GPIO10_ALIAS base address: 4032_5000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0201_0001h
4h	Parameter (PARAM)	32	R	0000_0002h
Ch	Lock (LOCK)	32	RW	0000_0000h
10h	Pin Control Nonsecure (PCNS)	32	RW	0000_0000h
14h	Interrupt Control Nonsecure (ICNS)	32	RW	0000_0000h
18h	Pin Control Nonprivilege (PCNP)	32	RW	0000_0000h
1Ch	Interrupt Control Nonprivilege (ICNP)	32	RW	0000_0000h
40h	Port Data Output (PDOR)	32	RW	0000_0000h
44h	Port Set Output (PSOR)	32	RW	0000_0000h
48h	Port Clear Output (PCOR)	32	RW	0000_0000h
4Ch	Port Toggle Output (PTOR)	32	RW	0000_0000h
50h	Port Data Input (PDIR)	32	R	0000_0000h
54h	Port Data Direction (PDDR)	32	RW	0000_0000h
58h	Port Input Disable (PIDR)	32	RW	0000_0000h
60h - 7Fh	Pin Data (P0DR - P31DR)	8	RW	00h
80h - FCh	Interrupt Control a (ICR0 - ICR31)	32	RW	0000_0000h
100h	Global Interrupt Control Low (GICLR)	32	RW	0000_0000h
104h	Global Interrupt Control High (GICHR)	32	RW	0000_0000h
120h - 124h	Interrupt Status Flag (ISFR0 - ISFR1)	32	RW	0000_0000h

22.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function
Indicates the version ID number of each GPIO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Basic implementation 0000_0000_0000_0001b - Protection registers implemented

22.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the interrupt number of each GPIO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												IRQNUM			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-4 —	Reserved
3-0 IRQNUM	Interrupt Number Indicates the number of interrupt, trigger, or DMA request domains.

22.7.1.4 Lock (LOCK)

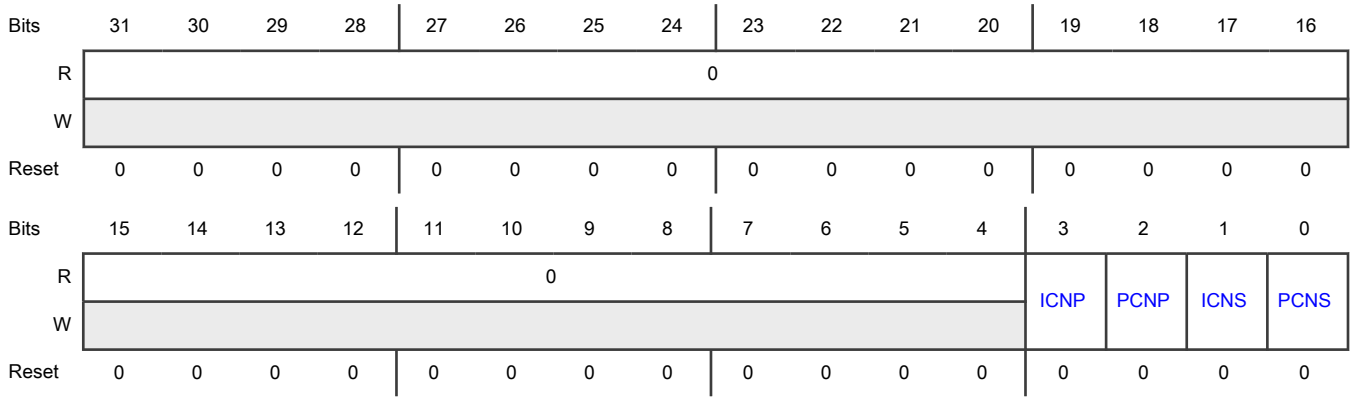
Offset

Register	Offset
LOCK	Ch

Function

Locks the nonsecure and nonprivilege access protection registers. You can write to this register only in the Secure-Privilege state.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 ICNP	Lock ICNP Locks Interrupt Control Nonprivilege (ICNP). If this field is 0, you can write to ICNP in the Secure-Privilege state. If this field is 1, you cannot write to ICNP until the next reset. 0b - Writable in Secure-Privilege state 1b - Not writable until the next reset
2 PCNP	Lock PCNP Locks Pin Control Nonprivilege (PCNP). If this field is 0, you can write to PCNP in the Secure-Privilege state. If this field is 1, you cannot write to PCNP until the next reset. 0b - Writable in Secure-Privilege state 1b - Not writable until the next reset
1 ICNS	Lock ICNS Locks Interrupt Control Nonsecure (ICNS). If this field is 0, you can write to ICNS in the Secure-Privilege state. If this field is 1, you cannot write to ICNS until the next reset. 0b - Writable in Secure-Privilege state 1b - Not writable until the next reset
0 PCNS	Lock PCNS Locks Pin Control Nonsecure (PCNS). If this field is 0, you can write to PCNS in the Secure-Privilege state. If this field is 1, you cannot write to PCNS until the next reset. 0b - Writable in Secure-Privilege state 1b - Not writable until the next reset

22.7.1.5 Pin Control Nonsecure (PCNS)

Offset

Register	Offset
PCNS	10h

Function

Configures secure or nonsecure access protection for each pin. You can write to this register only in the Secure-Privilege state if it is not locked (LOCK[PCNS] = 0).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE31	NSE30	NSE29	NSE28	NSE27	NSE26	NSE25	NSE24	NSE23	NSE22	NSE21	NSE20	NSE19	NSE18	NSE17	NSE16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE15	NSE14	NSE13	NSE12	NSE11	NSE10	NSE9	NSE8	NSE7	NSE6	NSE5	NSE4	NSE3	NSE2	NSE1	NSE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 NSEn	Nonsecure Enable Configures secure or nonsecure access protection for each pin. If this field is 0, the pin is configured for secure access. You can read or write to the corresponding pin's registers and fields only in the Secure state. When you access the corresponding pin's registers in the Nonsecure state, all fields in the registers related to that pin are read zero (with writes ignored). If this field is 1, the pin is configured for nonsecure access. You can read or write to the corresponding pin's registers and fields only in the Nonsecure state. When you access the corresponding pin's registers in the Secure state, all fields in the registers related to that pin are read zero (with writes ignored). 0b - Secure access 1b - Nonsecure access

22.7.1.6 Interrupt Control Nonsecure (ICNS)

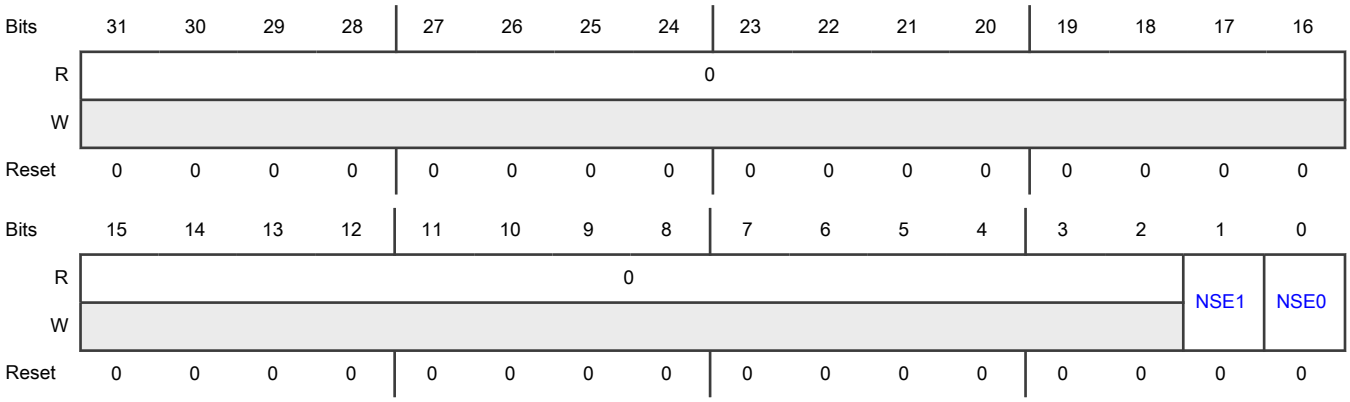
Offset

Register	Offset
ICNS	14h

Function

Configures secure and nonsecure access protection for each interrupt, output trigger, or DMA request. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[ICNS] = 0).

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 NSEn	Nonsecure Enable Configures secure or nonsecure access protection for each interrupt, output trigger, or DMA request. If this field is 0, the interrupt, output trigger, or DMA request is configured for secure access. You can configure a pin to use the corresponding interrupt, output trigger, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, output trigger, or DMA request only in the Secure state. If this field is 1, the interrupt, output trigger, or DMA request is configured for nonsecure access. You can configure a pin to use the corresponding interrupt, output trigger, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, output trigger, or DMA request only in the Nonsecure state. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0b - Secure access 1b - Nonsecure access

22.7.1.7 Pin Control Nonprivilege (PCNP)

Offset

Register	Offset
PCNP	18h

Function

Configures access protection on each pin either for privilege or for both privilege and nonprivilege accesses. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[PCNP] = 0).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NPE31	NPE30	NPE29	NPE28	NPE27	NPE26	NPE25	NPE24	NPE23	NPE22	NPE21	NPE20	NPE19	NPE18	NPE17	NPE16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NPE15	NPE14	NPE13	NPE12	NPE11	NPE10	NPE9	NPE8	NPE7	NPE6	NPE5	NPE4	NPE3	NPE2	NPE1	NPE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 NPEn	Nonprivilege Enable Configures privilege or nonprivilege access protection for each pin. If this field is 0, the pin is configured for privilege access. Write access to the corresponding pin's registers and fields is allowed only in the Privilege state. When you access the corresponding pin's registers and fields in the Nonprivilege state, all fields related to that pin in this GPIO are readable (with writes ignored). If this field is 1, the pin is configured for nonprivilege access; read or write access to the corresponding pin's registers is allowed in both Privilege and Nonprivilege states. 0b - Privilege access 1b - Nonprivilege access

22.7.1.8 Interrupt Control Nonprivilege (ICNP)

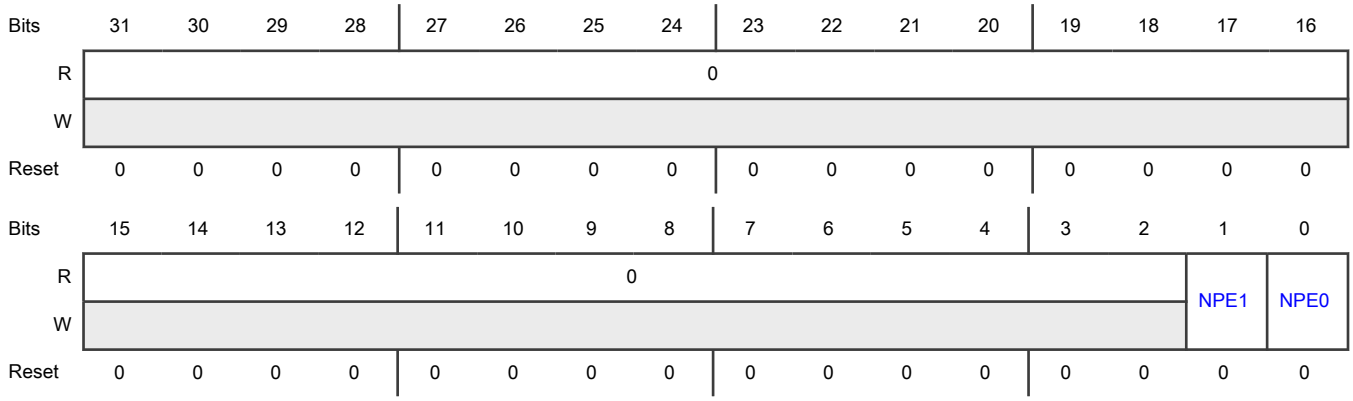
Offset

Register	Offset
ICNP	1Ch

Function

Configures privilege and nonprivilege access protection for each interrupt, trigger output, or DMA request. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[ICNP] = 0).

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 NPEn	Nonprivilege Enable Configures privilege or nonprivilege access protection for each interrupt, trigger output, or DMA request. If this field is 0, the pin is configured for privilege access. You can configure a pin to use the corresponding interrupt, trigger output, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, trigger output, or DMA request only in the Privilege state. If this field is 1, the pin is configured for nonprivilege access. In either Privilege or Nonprivilege state, you can configure a pin to use the corresponding interrupt, trigger output, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, trigger output, or DMA request. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0b - Privilege access 1b - Nonprivilege access

22.7.1.9 Port Data Output (PDOR)

Offset

Register	Offset
PDOR	40h

Function

Configures the logic levels that are driven on each general-purpose output pin.

NOTE

Do not modify the pin configuration registers associated with pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDO31	PDO30	PDO29	PDO28	PDO27	PDO26	PDO25	PDO24	PDO23	PDO22	PDO21	PDO20	PDO19	PDO18	PDO17	PDO16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO15	PDO14	PDO13	PDO12	PDO11	PDO10	PDO9	PDO8	PDO7	PDO6	PDO5	PDO4	PDO3	PDO2	PDO1	PDO0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PDOn	Port Data Output Configures the logic level on the pin if it is configured for general-purpose output. If this field is 0, logic level 0 is driven on the pin, if the pin is configured for general-purpose output. If this field is 1, logic level 1 is driven on the pin, if the pin is configured for general-purpose output. **NOTE** Reading the fields for unbonded pins returns an undefined value. 0b - Logic level 0 1b - Logic level 1

22.7.1.10 Port Set Output (PSOR)

Offset

Register	Offset
PSOR	44h

Function

Updates the corresponding fields of Port Data Output (PDOR) to become 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 31	PTSO 30	PTSO 29	PTSO 28	PTSO 27	PTSO 26	PTSO 25	PTSO 24	PTSO 23	PTSO 22	PTSO 21	PTSO 20	PTSO 19	PTSO 18	PTSO 17	PTSO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 15	PTSO 14	PTSO 13	PTSO 12	PTSO 11	PTSO 10	PTSO 9	PTSO 8	PTSO 7	PTSO 6	PTSO 5	PTSO 4	PTSO 3	PTSO 2	PTSO 1	PTSO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTSOn	Port Set Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 1. 0b - No change 1b - Corresponding field in PDOR becomes 1

22.7.1.11 Port Clear Output (PCOR)

Offset

Register	Offset
PCOR	48h

Function

Updates the corresponding fields of Port Data Output (PDOR) to become 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 31	PTCO 30	PTCO 29	PTCO 28	PTCO 27	PTCO 26	PTCO 25	PTCO 24	PTCO 23	PTCO 22	PTCO 21	PTCO 20	PTCO 19	PTCO 18	PTCO 17	PTCO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 15	PTCO 14	PTCO 13	PTCO 12	PTCO 11	PTCO 10	PTCO 9	PTCO 8	PTCO 7	PTCO 6	PTCO 5	PTCO 4	PTCO 3	PTCO 2	PTCO 1	PTCO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTCOn	Port Clear Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 0. 0b - No change 1b - Corresponding field in PDOR becomes 0

22.7.1.12 Port Toggle Output (PTOR)

Offset

Register	Offset
PTOR	4Ch

Function

Updates the corresponding fields of Port Data Output (PDOR) to set to the inverse of their current logic states.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 31	PTTO 30	PTTO 29	PTTO 28	PTTO 27	PTTO 26	PTTO 25	PTTO 24	PTTO 23	PTTO 22	PTTO 21	PTTO 20	PTTO 19	PTTO 18	PTTO 17	PTTO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 15	PTTO 14	PTTO 13	PTTO 12	PTTO 11	PTTO 10	PTTO 9	PTTO 8	PTTO 7	PTTO 6	PTTO 5	PTTO 4	PTTO 3	PTTO 2	PTTO 1	PTTO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTTOn	Port Toggle Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field is set to the inverse of its current logic state. 0b - No change 1b - Set to the inverse of its current logic state

22.7.1.13 Port Data Input (PDIR)

Offset

Register	Offset
PDIR	50h

Function

Captures the logic levels of each general-purpose input pin.

NOTE

Do not modify the pin configuration registers associated with the pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI31	PDI30	PDI29	PDI28	PDI27	PDI26	PDI25	PDI24	PDI23	PDI22	PDI21	PDI20	PDI19	PDI18	PDI17	PDI16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI15	PDI14	PDI13	PDI12	PDI11	PDI10	PDI9	PDI8	PDI7	PDI6	PDI5	PDI4	PDI3	PDI2	PDI1	PDI0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PDIn	Port Data Input Indicates the logic level of the pin that is configured for use by a digital function. If this field is 0, the pin logic level is logic 0 or is not configured or implemented for use by a digital function. If this field is 1, the pin logic level is logic 1. 0b - Logic 0 1b - Logic 1

22.7.1.14 Port Data Direction (PDDR)

Offset

Register	Offset
PDDR	54h

Function

Configures the individual port pins for input or output.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDD31	PDD30	PDD29	PDD28	PDD27	PDD26	PDD25	PDD24	PDD23	PDD22	PDD21	PDD20	PDD19	PDD18	PDD17	PDD16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD15	PDD14	PDD13	PDD12	PDD11	PDD10	PDD9	PDD8	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PDDn	Port Data Direction Configures individual port pins for input or output. If this field is 0, the pin is configured as general-purpose input for the GPIO function. If this field is 1, the pin is configured as general-purpose output for the GPIO function. 0b - Input 1b - Output

22.7.1.15 Port Input Disable (PIDR)

Offset

Register	Offset
PIDR	58h

Function

Disables the input for each general-purpose pin, which prevents the value from being reported in Port Data Input (PDIR).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PIDn	Port Input Disable Disables a pin for general-purpose input. If this field is 0, the pin is configured for general-purpose input, provided the pin is configured for a digital function. If this field is 1, the pin is disabled for general-purpose input. 0b - Configured for general-purpose input 1b - Disabled for general-purpose input

22.7.1.16 Pin Data (P0DR - P31DR)

Offset

For a = 0 to 31:

Register	Offset
PaDR	60h + (a × 1h)

Function

Configures the data feature of a pin. Pins that are unimplemented or not configured for a digital function read zero.

NOTE

You must not modify Pin Data (P0DR - P31DR) associated with the pins that are not available in your selected package. These unbonded pins are, by default, set to the Disable state for lowest power consumption.

Diagram

Bits	7	6	5	4	3	2	1	0
R	0							PD
W								
Reset	0	0	0	0	0	0	0	0

Fields

Field	Function
7-1 —	Reserved
0 PD	Pin Data (I/O) Specifies the pin logic level. This field updates the corresponding field in Port Data Output (PDOR); reading this field returns the value in the corresponding field of Port Data Input (PDIR). If this field is 0, the pin logic level is logic zero or not configured for use by a digital function. If this field is 1, the pin logic level is logic one. 0b - Logic zero 1b - Logic one

22.7.1.17 Interrupt Control a (ICR0 - ICR31)

Offset

For a = 0 to 31:

Register	Offset
ICRa	80h + (a × 4h)

Function

Configures interrupt features on each pin.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	LK	0		IRQS	IRQC			
W								W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-25 —	Reserved
24 ISF	Interrupt Status Flag Indicates whether the configured interrupt is detected. The pin interrupt configuration is valid in all digital pin muxing modes. The fields in Interrupt Status Flag (ISFR0 - ISFR1) have the same function. ISF can be cleared with either register field. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
23 LK	Lock

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Locks ICR[23:0], and you cannot update the locked field until the next system reset. If this field is 0, interrupt configuration by ICR[23:0] is not locked and can be updated. If this field is 1, interrupt configuration by ICR[23:0] is locked and cannot be updated until the next system reset. 0b - Not locked 1b - Locked
22-21 —	Reserved
20 IRQS	Interrupt Select Configures the selected interrupt, trigger output, or DMA request. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0b - Interrupt, trigger output, or DMA request 0 1b - Interrupt, trigger output, or DMA request 1
19-16 IRQC	Interrupt Configuration Specifies the ISF and DMA request configuration. The pin interrupt configuration is valid in all digital pin muxing modes. When changing the interrupt configuration, it is recommended to first disable ISF and then write the new configuration. The corresponding pin is configured to generate interrupt, trigger, or DMA request. **NOTE** See the GPIO chip-specific information to determine the GPIO instances that support interrupt, DMA request, or trigger capabilities. 0000b - ISF is disabled 0001b - ISF and DMA request on rising edge 0010b - ISF and DMA request on falling edge 0011b - ISF and DMA request on either edge 0100b - Reserved 0101b - ISF sets on rising edge 0110b - ISF sets on falling edge 0111b - ISF sets on either edge 1000b - ISF and interrupt when logic 0 1001b - ISF and interrupt on rising edge 1010b - ISF and interrupt on falling edge 1011b - ISF and Interrupt on either edge

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1100b - ISF and interrupt when logic 1 1101b - Enable active-high trigger output; ISF on rising edge (pin state is ORed with other enabled triggers to generate the output trigger for use by other peripherals) 1110b - Enable active-low trigger output; ISF on falling edge (pin state is inverted and ORed with other enabled triggers to generate the output trigger for use by other peripherals) 1111b - Reserved
15-0 —	Reserved

22.7.1.18 Global Interrupt Control Low (GICLR)

Offset

Register	Offset
GICLR	100h

Function

Updates any combination of the lower 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 15	GIWE 14	GIWE 13	GIWE 12	GIWE 11	GIWE 10	GIWE 9	GIWE 8	GIWE 7	GIWE 6	GIWE 5	GIWE 4	GIWE 3	GIWE 2	GIWE 1	GIWE 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.
15-0 GIWEn	Global Interrupt Write Enable Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0b - Not updated 1b - Updated

22.7.1.19 Global Interrupt Control High (GICHR)

Offset

Register	Offset
GICHR	104h

Function

Updates any combination of the upper 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 31	GIWE 30	GIWE 29	GIWE 28	GIWE 27	GIWE 26	GIWE 25	GIWE 24	GIWE 23	GIWE 22	GIWE 21	GIWE 20	GIWE 19	GIWE 18	GIWE 17	GIWE 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-0 GIWEn	Global Interrupt Write Enable Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0b - Not updated. 1b - Updated

22.7.1.20 Interrupt Status Flag (ISFR0 - ISFR1)

Offset

Register	Offset
ISFR0	120h
ISFR1	124h

Function

Indicates whether the related configured interrupt is detected on each pin. The pin interrupt configuration is valid in all digital pin muxing modes. The ISF for each pin is also visible in the corresponding Interrupt Control register, and each flag can be cleared in either location.

There is a separate ISF register for each interrupt, trigger, or DMA request domain. Each status flag is only visible in the register that corresponds to that flag's domain, as configured in ICR*n*[IRQS].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISF31	ISF30	ISF29	ISF28	ISF27	ISF26	ISF25	ISF24	ISF23	ISF22	ISF21	ISF20	ISF19	ISF18	ISF17	ISF16
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF15	ISF14	ISF13	ISF12	ISF11	ISF10	ISF9	ISF8	ISF7	ISF6	ISF5	ISF4	ISF3	ISF2	ISF1	ISF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ISFn	Interrupt Status Flag

Table continues on the next page...

Field	Function
	Indicates the detection of the configured interrupt on each pin of the same number. If this field is 0, the configured interrupt is not detected on the pin of the same number. If this field is 1, the configured interrupt is detected on the pin of the same number. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag

Chapter 23
Performance Monitor (CMX_PERFMON)

23.1 Chip-specific CMX_PERFMON information

Table 194. Reference links to related information

Topic	Related module	Reference
Full description	CMX_PERFMON	CMX_PERFMON
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

23.1.1 Module instances

This device has four instances of the CMX_PERFMON module, CMX_PERFMON0 to CMX_PERFMON3. These fours instances monitor:

Table 195. CMX_PERFMON instances

	CMX_PERFMON0	CMX_PERFMON1	CMX_PERFMON2	CMX_PERFMON3
Located in	VDD2_COMP	VDD2_COMP	VDD2_COMP	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
Monitors	XCACHE1 (code cache)	XCACHE0 (system cache)	CACHE64_CTRL0	CACHE64_CTRL1

23.1.2 PMECTR counters

The following table describes the countable events each CMX_PERFMON instance can monitor by configuring PMCR[SELEVT*n*] (*n* = 1-3) .

Table 196. Countable events CMX_PERFMON monitors

PMCR[SELEVT*n*] (*n* = 1-3)	Mnemonic	Descriptons	
100_0000b	cac_if	AHB nonTCM ("cache") transfers	Cached instruction fetches
100_0100b	cac_if_stalls		Stalled cached instruction fetches
100_0001b	cac_opr		Cached operand reads
100_0101b	cac_opr_stalls		Stalled cached operand reads
100_0010b	cac_opw		Cached operand writes

Table continues on the next page...

Table 196. Countable events CMX_PERFMON monitors (continued)

PMCR[SELEVT*n*] (*n* = 1-3)	Mnemonic	Descriptons	
100_0110b	cac_opw_stalls		Stalled cached operand writes
100_0011b	cac_opx		Cached operand reads and writes
100_0111b	cac_opx_stalls		Stalled cached operand reads and writes
100_1000b	cac_wt_opw		Cache writethrough operand writes
100_1100b	cac_wt_opw_stalls		Stalled writethrough operand writes
101_0001b	cac_opr_miss		Operand read miss
101_0010b	cac_opw_miss		Operand write miss
101_0011b	cac_opx_miss		Operand read and write miss
101_1000b	cac_wb_push		Cache writeback line pushes
101_1100b	cac_push_stalls		Stalled cache writeback line pushes

23.2 Overview

CMX_PERFMON contains counters that you can configure to count events used to calculate the performance of a CPU, cache, or memory. You can configure CMX_PERFMON to select the events to be counted for each counter. An additional instruction counter is available to count CPU instruction fetches.

Block diagram provides an overview of CMX_PERFMON, which includes three event counters and an instruction counter.

23.2.1 Block diagram

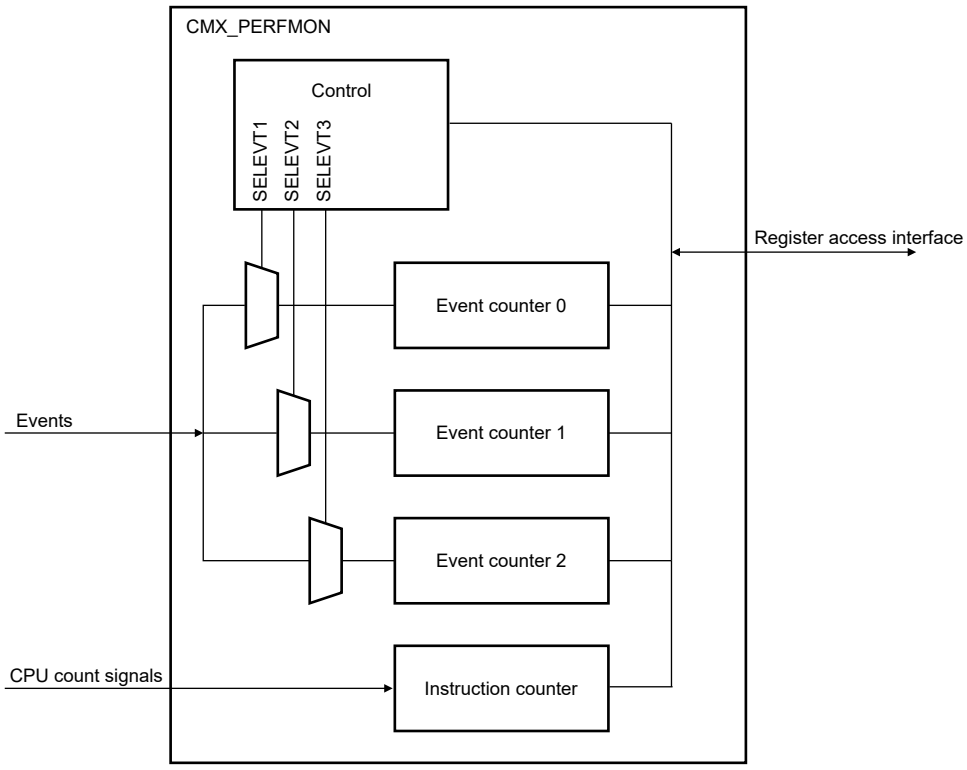


Figure 75. Block diagram

23.2.2 Features

- Three 40-bit event counters
- One 40-bit instruction counter
- Programmable event select
- Countable events that include cache, branch fetches, and stalls
- Memory mapped performance monitor that counts instructions, cache, branch, IPS, and TCM stats
- Count filter based on privilege level

23.3 Functional description

23.3.1 Operations

See the chip-specific CMX_PERFMON information for details about the countable events for the performance counters per CMX_PERFMON instance on this chip.

CMX_PERFMON has 3 programmable counters to count CPU events. The event to be counted is configured using the PMCR0 register.

Each counter is a 40-bit register that tracks the number of occurrences of the selected event.

The counters can be programmed to take one of the 3 states:

- At idle state, the counters are in idle and do not perform any operation.

- At local start state, the counters are actively listening to CPU and cache and start incrementing when the selected events are observed.
- At local stop state, the counters stop listening to all components. The 3 event counter can be independently cleared by writing to PMCR0[RECTR*n*], where *n* denotes the counter instance.

23.3.2 Operating modes

The following table shows:

- The operating modes that you can specify for CMX_PERFMON.
- The PMCR0[CMODE] value that you must specify to count events in each mode.

Table 197. Operating modes

Mode	Description	CMODE value (binary)
Privileged	If supported, only counts events in Privilege mode.	11
User	If supported, only counts events in User mode.	10
Both Privileged and User	Counts events in both Privilege and User mode.	00

Use PMCR0[SSC] to start and stop the counters.

23.3.3 Clocking

Clock name	Direction	Description
CPU clock (hclk)	Input	Configures the counter logic.
Bus clock (ipg_clk)	Input	Controls the access to the registers.

23.3.4 Interrupts

This module has no interrupts.

23.4 External signals

CMX_PERFMON has no external signals.

23.5 Initialization

To initialize CMX_PERFMON:

1. Program PMCR0[SELEVT*n*] to select the specific CPU event to be counted by this counter, where *n* denotes the counter instance.
2. Set PMCR0[CMODE] to filter each counter for the selected event while the CPU is in privilege, user, or either mode.
3. Set PMCR0[SSC] to program the counter to be in one of idle, local start, or local stop states.
4. Set PMCR0[RECTR*n*] to clear the counters independently where *n* denotes the counter instance.

23.6 Memory map and register definition

23.6.1 CMX_PERFMON register descriptions

23.6.1.1 CMX_PERFMON memory map

CMX_PERFMON0 base address: 4018_C000h

CMX_PERFMON1 base address: 4018_D000h

CMX_PERFMON2 base address: 4018_E000h

CMX_PERFMON3 base address: 4018_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Performance Monitor Control (PMCR)	32	RW	0000_0000h
10h	Performance Monitor Instruction Counter (PMICTR_HI)	8	R	00h
14h	Performance Monitor Instruction Counter (PMICTR_LO)	32	R	0000_0000h
18h	Performance Monitor Event Counter (PMECTR1_HI)	8	R	00h
1Ch	Performance Monitor Event Counter (PMECTR1_LO)	32	R	0000_0000h
20h	Performance Monitor Event Counter (PMECTR2_HI)	8	R	00h
24h	Performance Monitor Event Counter (PMECTR2_LO)	32	R	0000_0000h
28h	Performance Monitor Event Counter (PMECTR3_HI)	8	R	00h
2Ch	Performance Monitor Event Counter (PMECTR3_LO)	32	R	0000_0000h

23.6.1.2 Performance Monitor Control (PMCR)

Offset

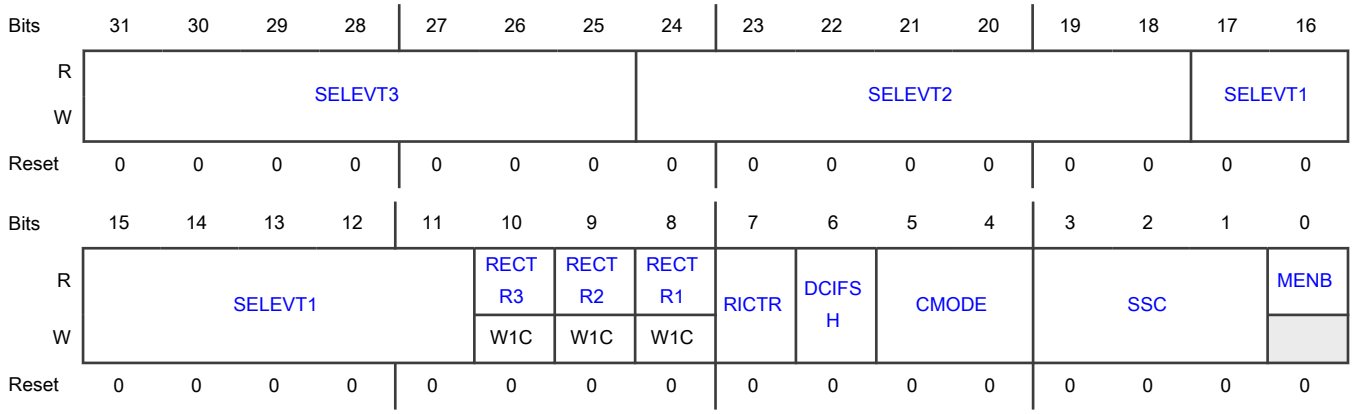
Register	Offset
PMCR	0h

Function

Specifies:

- The events that the PMECTR*n*HI and PMECTR*n*LO count.
- The count mode.
- The start and stop control.
- The enables for the counters.

Diagram



Fields

Field	Function
31-25 SELEVT3	Select Event 3 Selects the event to be counted in PMECTR3. See the chip-specific CMX_PERFMON information to select which event PMECTR3 counts.
24-18 SELEVT2	Select Event 2 Selects the event to be counted in PMECTR2. See the chip-specific CMX_PERFMON information to select which event PMECTR2 counts.
17-11 SELEVT1	Select Event 1 Selects the event to be counted in PMECTR1. See the chip-specific CMX_PERFMON information to select which event PMECTR1 counts.
10 RECTR3	Reset Event Counter 3 Writing 1 to this field resets the event counter at the end of the cycle. 0b - Run normally 1b - Reset
9 RECTR2	Reset Event Counter 2 Writing 1 to this field resets the event counter at the end of the cycle. 0b - Run normally 1b - Reset
8 RECTR1	Reset Event Counter 1 Writing 1 to this field resets the event counter at the end of the cycle. 0b - Run normally 1b - Reset

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>7
RICTR</td><td>Reset Instruction Counter
Determines whether events reset or clear the instruction counter. Write 1 to clear the instruction counter. This field must be 0 to restart the instruction counter.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>CMX_PERFMON0</td><td>PMCR</td><td>—</td></tr><tr><td>CMX_PERFMON1</td><td>PMCR</td><td>—</td></tr><tr><td>CMX_PERFMON2</td><td>—</td><td>PMCR</td></tr><tr><td>CMX_PERFMON3</td><td>—</td><td>PMCR</td></tr></table>
0b - Do not reset
1b - Clear</td></tr>
<tr><td>6
DCIFSH</td><td>Disable Counters if Stopped or Halted
Specifies whether counters continue counting or stop counting when the CPU is halted.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>CMX_PERFMON0</td><td>PMCR</td><td>—</td></tr><tr><td>CMX_PERFMON1</td><td>PMCR</td><td>—</td></tr><tr><td>CMX_PERFMON2</td><td>—</td><td>PMCR</td></tr><tr><td>CMX_PERFMON3</td><td>—</td><td>PMCR</td></tr></table>
0b - Continue
1b - Stop</td></tr>
<tr><td>5-4
CMODE</td><td>Count Mode
Determines whether events are counted for Privileged mode, User mode, or both Privileged and User modes. This setting affects the operation of all event counters.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Counted in both User and Privileged modes 01b - Reserved 10b - Reserved 11b - Reserved
3-1 SSC	Start and Stop Control Provides a three-phase mechanism to start and stop the counters. It includes a prioritized scheme with the following relative priorities: local start > local stop The start or stop affects all performance monitor event counters concurrently, so that the counters are coherent. 000b - Idle or no-op 001b - Local stop 010b,011b - Local start 100b - Reserved 101b - Reserved 110b,111b - Reserved
0 MENB	Module Is Enabled Indicates whether CMX_PERFMON is enabled. Enabling CMX_PERFMON does not start the counters. To start the counters, write a nonzero value to PMCR0[SSC]. 0b - Disabled 1b - Enabled

23.6.1.3 Performance Monitor Instruction Counter (PMICTR_HI)

Offset

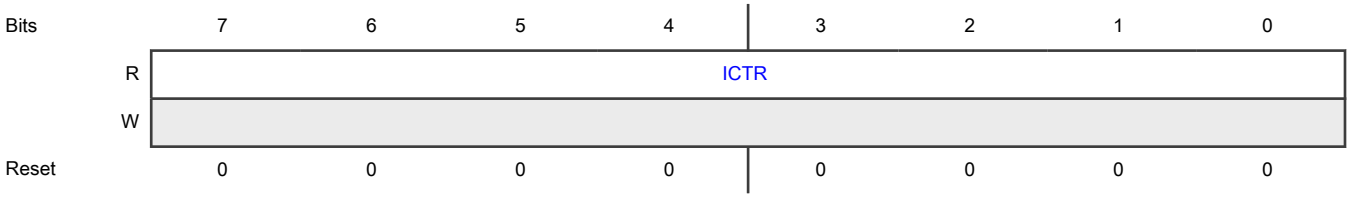
Register	Offset
PMICTR_HI	10h

Function
Provides part of a 40-bit counter that counts executed instructions.

NOTE
Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
CMX_PERFMON0	PMICTR_HI	—
CMX_PERFMON1	PMICTR_HI	—
CMX_PERFMON2	—	PMICTR_HI
CMX_PERFMON3	—	PMICTR_HI

Diagram



Fields

Field	Function
7-0 ICTR	Instruction Counter Specifies the upper 8 bits of the 40-bit instruction counter that reflects the number of instructions executed. This field is updated based on the instruction execution status, as indicated on the processor ETM interface. The ETM interface indicates when instructions are valid, folded, cancelled or stalled, from which the CMX_PERFMON deduces the overall instruction count.

23.6.1.4 Performance Monitor Instruction Counter (PMICTR_LO)

Offset

Register	Offset
PMICTR_LO	14h

Function

Provides part of a 40-bit counter that counts executed instructions.

NOTE

Each module instance supports a different number of registers.

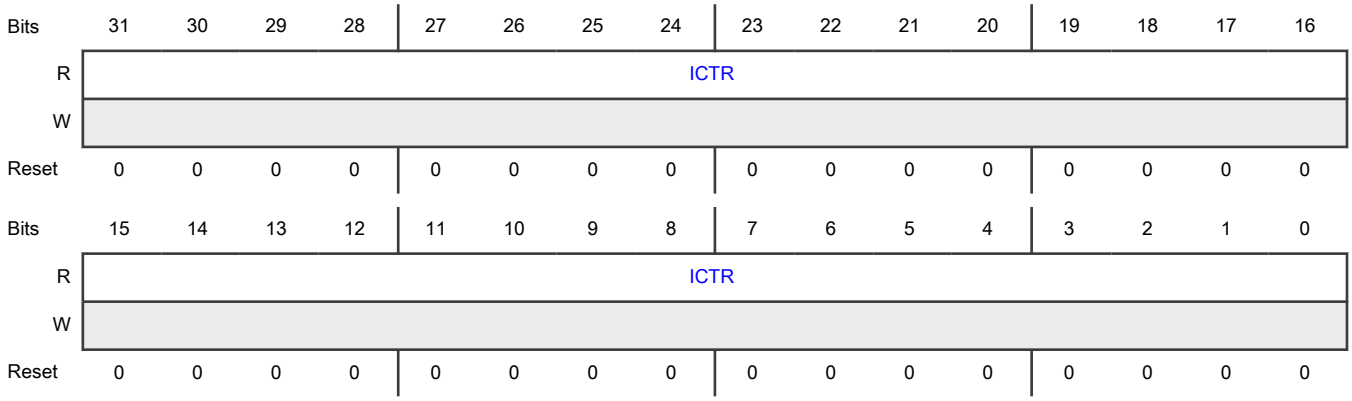
Instance	Register supported	Register not supported
CMX_PERFMON0	PMICTR_LO	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
CMX_PERFMON1	PMICTR_LO	—
CMX_PERFMON2	—	PMICTR_LO
CMX_PERFMON3	—	PMICTR_LO

Diagram



Fields

Field	Function
31-0 ICTR	Instruction Counter Specifies the lower 32 bits of the instruction counter that reflects the number of instructions executed. This field is updated based on the instruction execution status, as indicated on the processor ETM interface. The ETM interface indicates when instructions are valid, folded, cancelled or stalled, from which the CMX_PERFMON deduces the overall instruction count.

23.6.1.5 Performance Monitor Event Counter (PMECTR1_HI - PMECTR3_HI)

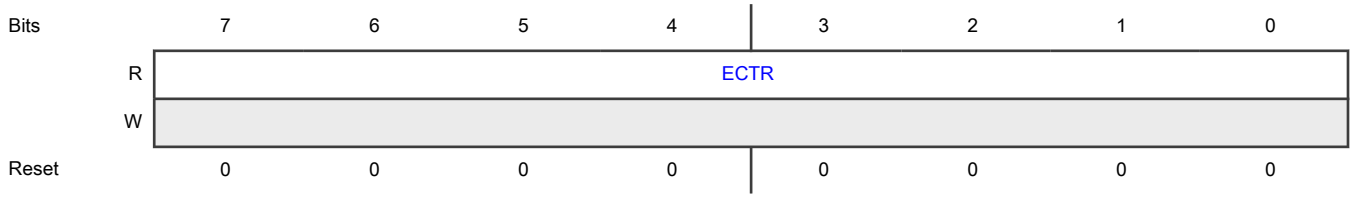
Offset

Register	Offset
PMECTR1_HI	18h
PMECTR2_HI	20h
PMECTR3_HI	28h

Function

Provides part of a 40-bit event counter that you configure with PMCR0[SELEVT*n*].

Diagram



Fields

Field	Function
7-0 ECTR	Event Counter Specifies the upper 8 bits of the event[*n*] counter.

23.6.1.6 Performance Monitor Event Counter (PMECTR1_LO - PMECTR3_LO)

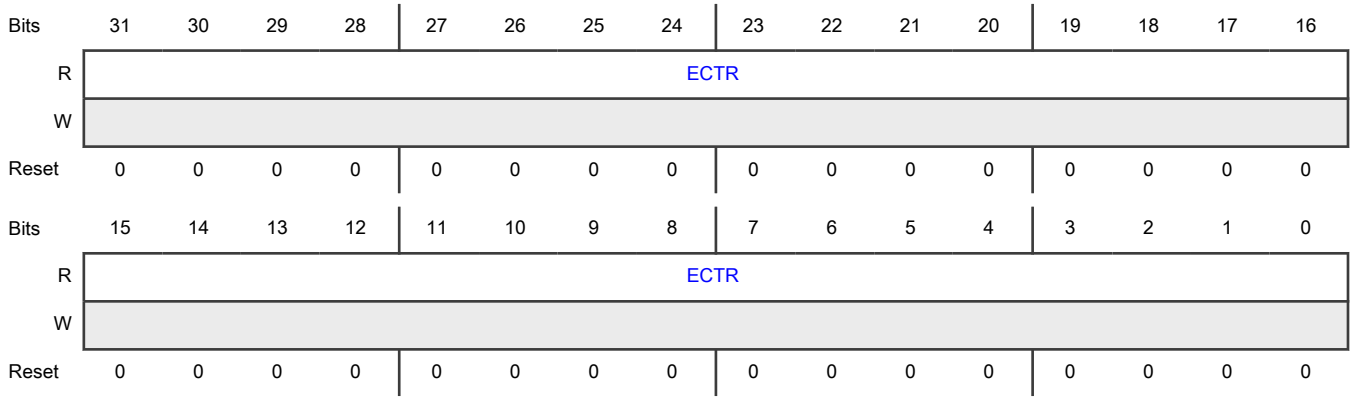
Offset

Register	Offset
PMECTR1_LO	1Ch
PMECTR2_LO	24h
PMECTR3_LO	2Ch

Function

Provides part of a 40-bit event counter that you configure with PMCR0[SELEVT*n*].

Diagram



Fields

Field	Function
31-0	Event Counter

Table continues on the next page...

Field	Function
ECTR	Specifies the lower 32 bits of the event[*n*] counter. The 40-bit counter value increments each time the event selected in [SELEVT*n*] occurs.

Chapter 24
Cache Memory Controller (XCACHE)

24.1 Chip-specific XCACHE information

Table 198. Reference links to related information

Topic	Related module	Reference
Full description	XCACHE	XCACHE
Peripheral memory map		Memory map overview
Cacheable areas		Memory map overview
Power management	PMC	Power overview PMC

24.1.1 Module instances

This device contains two instances of the XCACHE module, XCACHE0 and XCACHE1.

Table 199. XCACHE instances

	XCACHE0	XCACHE1
Located in	VDD2_COMP	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
Functions	CPU0 system cache	CPU0 code cache
Related CMX_PERFMON	CMX_PERFMON1	CMX_PERFMON0

24.1.2 CSAR register

For this chip, the CSAR[28] must be configured as 0b.

24.2 Overview

XCACHE is a general-purpose AMBA AHB bus protocol cache that provides low-latency access to slower memories.

24.2.1 Block diagram

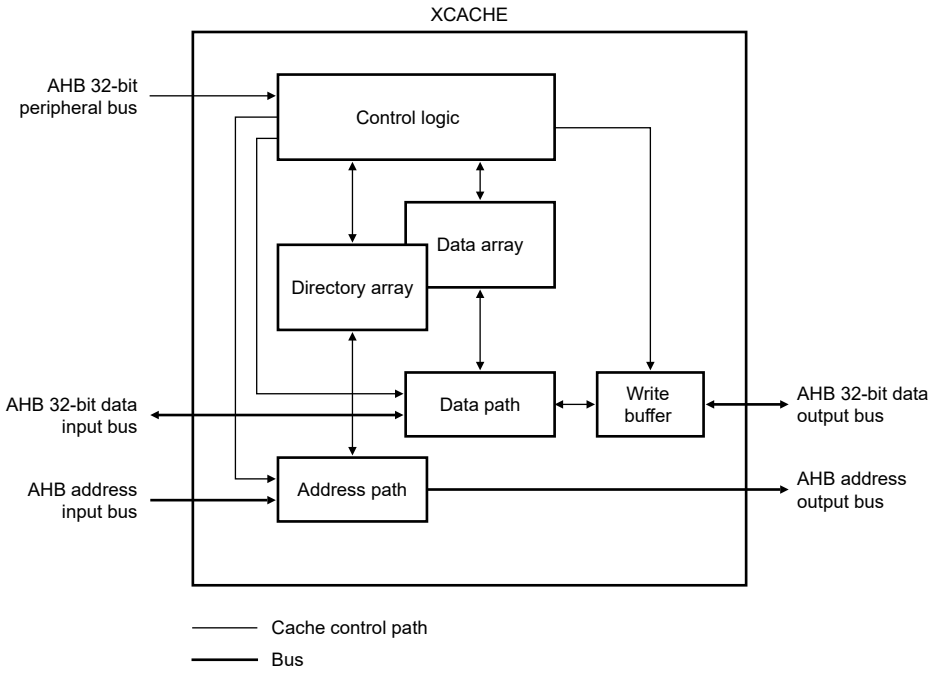


Figure 76. Block diagram

24.2.2 Features

XCACHE is designed to work with AMBA AHB bus systems. It includes:

- An AHB input bus. Accesses have cache mode attributes for cacheable copyback, cacheable write-through, or noncacheable mode operation and for write bufferable mode operation on an access-by-access basis.
- 32-bit data path AHB output bus for downstream cache read miss, cache write-through write, and noncacheable accesses.
- Connections to tag and data RAMs for implementing the cache functions.
- A peripheral interface for accessing the cache's programming model.

XCACHE includes the following AMBA_AHB buses:

- AHB input bus
- AHB output bus

24.3 Functional description

XCACHE is a block of high-speed memory locations that contains address information (commonly known as a tag) and its associated data. It decreases the average time of a memory access.

When data loads into a cache, the access times for subsequent loads and stores are reduced, resulting in overall performance benefits. Access to information already in a cache is known as a cache hit, and other accesses are known as cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the input bus wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

24.3.1 Principles of locality

XCACHE operates using the following two principles of locality.

Table 200. Type of locality

Type of locality	Description	Access
Spatial	Groups several locations together under the same tag to minimize the quantity of control information stored. This logical block is commonly known as a cache line.	Access to one location is possibly followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
Temporal	Retains recently accessed lines in the cache.	Access to an area of memory possibly repeats within a short period of time (for example, execution of a code loop).

24.3.2 Modes of operation

Table 201. Modes of operation

Mode	Description
Write-Through	Access to address spaces with this cache mode are cacheable: • If all cacheable spaces are read-only spaces, then cache contains read-only data and all writes to the cache cause a fault. See chip-specific cacheable space information. • A read miss on the input bus causes a line read on the output bus of a 16-byte-aligned memory address that contains the desired address. This missed data is loaded into the cache and is marked as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus access. • A write miss bypasses the cache and writes to the output bus (no allocation on write-miss policy for Write-Through mode spaces). • A write hit updates the cache hit line with the write data and writes to the output bus. • Caches are input-bus-local and do not support hardware cache coherency. If the input bus accesses write-through regions and an external input bus (such as DMA) need to update these regions, you must first perform explicit cache clears to any needed cache memory range. This ensures all modified cache lines update their associated memories before external inputs modify them, and subsequent cache input bus accesses get the updated memory.
Write-back	Access to address spaces with this cache mode are cacheable: • A read miss on the input bus causes a line read on the output bus of a 16-byte-aligned memory address that contains the desired address. This missed data is loaded into the cache and is marked as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus access. • A write miss performs a read-to-write (allocate on write miss policy for write-back mode spaces). You perform a line read on the output bus of a

Table continues on the next page...

Table 201. Modes of operation (continued)

Mode	Description
	16-byte-aligned memory address containing the desired write address. This missed data is loaded into the cache and marked as valid and modified. Then the write data updates the appropriate cache data locations. • A write hit updates the cache hit line with the write data and marks the line as modified. • Caches do not support hardware cache coherency. If the cache's input bus has written to write-back regions and another input bus that can access the same target memory without going through the cache needs to see these updates, you must perform explicit cache pushes to any needed cache memory range to ensure all modified cache lines update their associated memories before being read by this input. Likewise, if the cache's input bus has accessed write-back or write-through regions and another input bus that can access the same target memory without going through the cache needs to update these regions, you must first perform explicit cache clears to any needed cache memory range. This ensures all modified cache lines update their associated memories before this input modifies them, and subsequent cache input bus accesses get the updated memory.
Non-cacheable	Access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

24.3.3 XCACHE function

XCACHE receives bus requests on the input bus.

You can access the programming model for the cache via XCACHE's AMBA APB bus.

XCACHE then processes the cacheable accesses as needed, when bypassing the noncacheable, cache write-through, cache miss, and cache maintenance accesses to its output bus.

The cache on this chip is structured as follows. It has a 2-way set-associative cache structure with a total size of 32 KB for the cache. The caches have 32-bit addresses and data paths and a 16-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

The tag cache contains the required upper address bits, a modify bit, a valid bit, and up to 8 bits of extended address attributes. The extended address attribute field has information such as the supervisor, user, and/or domain ID of the accesses loading this tag. If any extended address attribute bits are provided, for subsequent accesses that hit a valid tag but have different attributes, the cache coherently forces a miss so that the downstream memory protection logic can check the access.

For this 32 KB cache, each cache tag function uses two 1024 x 20-bit RAM arrays or optionally two up to 1024 x 28-bit RAM arrays (if all 8 bits of the extended address attribute are used). The cache data function uses two 4096 x 32-bit RAM arrays. The cache tag entries store 18 bits of upper address as well as a modified and valid bit per cache line, and any optional extended address attribute bits. The cache data entries store four bytes of code or data.

All normal cache accesses use physical addresses. This leads to the use of the following cache address:

CACHE - 32 KB size = (1024 sets) x (16-byte lines) x (2-way set associative)

Table 202. Tag cache address use

Tag cache address use	Cache
Tag hit address range	Address[31:14]

Table continues on the next page...

Table 202. Tag cache address use (continued)

Tag cache address use	Cache
Tag set select address range	Address[13:4] selects 1 of the1024 sets
Not used	Address[3:0]

Table 203. Data cache address use

Data cache address use	System cache
Not used	Address[31:14]
Data set select address range	Address[13:4] used to select one of the 1024 sets
32-bit word select	Address[3:2] used to select one of the four 32-bit words within a set
Byte select	Address[1:0] used to select the byte within the 32-bit word

24.3.4 XCACHE control

The cache is disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable and configure the cache, you must clear the cache commands and initialize the required tag array.

24.3.4.1 XCACHE set commands

XCACHE set commands may operate on:

- All of way 0
- All of way 1
- All of both ways (complete cache)

You can initiate cache set commands by using the upper bits in Cache Control (CCR). Cache set commands perform their operation on the cache independent of CCR[ENCACHE].

You initiate a cache set command by writing 1 to CCR[GO]. This field also acts as a busy bit for set commands and remains 1 when the command is active. XCACHE writes 0 to it after the completion of the set command.

Table 204 shows a list of supported cache set commands:

- Invalidate − Unconditionally clears valid and modify bits of a cache entry.
- Push − Pushes a cache entry if it is valid and modified, and then clears the modify bit. If the entry is not valid or not modified, leave it as is.
- Clear − Pushes a cache entry if it is valid and modified, and then clears the valid and modify bits. If the entry is not valid or not modified, clear the valid bit.

Table 204. Cache set commands

Cache Control (CCR)[27:24] settings				Command
CCR[PUSHW1]	CCR[INVW1]	CCR[PUSHW0]	CCR[INVW0]	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0

Table continues on the next page...

Table 204. Cache set commands (continued)

Cache Control (CCR)[27:24] settings				Command
CCR[PUSHW1]	CCR[INVW1]	CCR[PUSHW0]	CCR[INVW0]	
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1 and way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, writing 8500_0001h to Cache Control (CCR) invalidates the cache and enables the cache.

24.3.4.2 XCACHE line commands

XCACHE line commands operate on a single line in the cache at a time. You can perform cache line commands using a physical or cache address that consists of a set address and a way select. The line command acts on the specified cache line.

Cache line commands with physical addresses first search both ways of the cache set specified by the following physical address bits (see Table 205). If they hit, the commands perform their action on the hit way.

You specify cache line commands using the upper bits in Cache Line Control (CLCR). Cache line commands perform their operation on the cache independent of CCR[ENCACHE]. Using a cache address, the command can be completely specified using Cache Line Control (CLCR). Using a physical address, the command must also use Cache Search Address (CSAR) to specify the physical address.

You can initiate a line cache command by writing 1 to CLCR[LGO] or CSAR[LGO]. These fields also act as busy bits for line commands and remain 1 when the line cache command is active and XCACHE writes 0 to them after the completion of the command.

Table 205. Cache line commands

Cache Line Control (CLCR)[27:24] settings			Command
CLCR[LACC]	CLCR[LADSEL]	CLCR[LCMD]	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way

Table continues on the next page...

Table 205. Cache line commands (continued)

Cache Line Control (CLCR)[27:24] settings			Command
CLCR[LACC]	CLCR[LADSEL]	CLCR[LCMD]	
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

24.3.4.2.1 Executing a series of line commands using cache addresses

You can perform a series of line commands with incremental cache addresses by writing to Cache Line Control (CLCR):

1. Place the command in the following fields:
   - CLCR[LACC]
   - CLCR[LADSEL]
   - CLCR[LCMD]
2. Write 1 to CLCR[WSEL] and CLCR[TDSEL] as required.
3. Place the cache address in CLCR[CACHEADDR].
4. Write 1 to CLCR[LGO].

After one line command completes, perform the following procedure to initiate the next command:

1. Increment the cache address (at bit 2 to step through data or at bit 4 to step through lines).
2. Write 1 to CLCR[LGO].

24.3.4.2.2 Executing a series of line commands using physical addresses

You can perform a series of line commands with incremental physical addresses by writing to Cache Line Control (CLCR):

1. Place the command in the following fields:
   - CLCR[LACC]
   - CLCR[LADSEL]
   - CLCR[LCMD]
2. Write 1 to CLCR[TDSEL].
3. Place the physical address in CSAR[PHYADDR] and write 1 to CSAR[LGO].

After one line command completes, perform the following procedure to initiate the next command:

1. Increment the physical address (at bit 2 to step through data or at bit 4 to step through lines).

2. Write 1 to CSAR[LGO].

Cache Line Control (CLCR) and Cache Search Address (CSAR) both contain a field called LGO. Therefore, you can complete the aforementioned steps in a single write to CSAR.

24.3.4.2.3 Line command results

After the completion of a line command, Cache Line Control (CLCR) contains information on the initial state of the line targeted by the command. For line commands with cache addresses, read this information before performing a line command action from the targeted cache line. For line commands with physical addresses, read this information on a hit before performing a line command action from the hit cache line, or, clear the initial valid bit if the command misses. In general, if CLCR[LCIVB] becomes 0, the targeted line is invalid at the start of the line command and no line operation is performed.

Table 206. Line command results

Cache Line Control (CLCR)[22:20] settings			For cache address commands	For physical address commands
CLCR[LCWAY]	CLCR[LCIMB]	CLCR[LCIVB]		
0	0	0	Way 0 line is invalid	No hit
0	0	1	Way 0 valid and not modified	Way 0 valid and not modified
0	1	0	Way 0 line is invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line is invalid	No hit
1	0	1	Way 1 valid and not modified	Way 1 valid and not modified
1	1	0	Way 1 line is invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

After the completion of a line command other than a write, Cache Read/Write Value (CCVR) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command uses a physical address and misses, the data is don't care. For write commands, Cache Read/Write Value (CCVR) holds the write data.

The cache does not have line lock capability.

24.3.5 Clocks

XCACHE uses the AHB input bus clock.

24.3.6 Interrupts

XCACHE has no interrupts.

24.4 External signals

XCACHE has no external signals.

24.5 Initialization

To configure and enable the cache, you must perform XCACHE commands to clear and initialize the required tag array fields.

Table 207. Initialization sequence

Purpose	Programming	Notes
Enable the cache after reset, with the write buffer enabled.	Write 8500_0003h to Cache Control (CCR).	After you write to CCR[GO], it initiates the cache command specified by the following fields: • CCR[INVW1] • CCR[INVW0] The aforementioned fields indicate a command to clear all entries in both cache ways. CCR[ENWRBUF] enables the write buffer and CCR[ENCACHE] enables the cache. See XCACHE control for more information.
Enable the cache after reset, with the write buffer disabled.	Write 8500_0001h to Cache Control (CCR).	After you write to CCR[GO], it initiates the cache command specified by the following fields: • CCR[INVW1] • CCR[INVW0] The aforementioned fields indicate a command to clear all entries in both cache ways, and CCR[ENCACHE] enables the cache. See XCACHE control for more information.

24.6 Application information

XCACHE provides a low-latency access to instructions or data. This decouples processor performance from system memory performance, thereby increasing bus availability for other modules and improving system performance.

24.7 XCACHE register descriptions

The XACHE programming model provides various registers for the following functions:

- Configuring and controlling the cache
- Providing indirect access paths to all cache tag and data storage

NOTE

You can access XCACHE registers only in Supervisor mode.

24.7.1 XCACHE memory map

XCACHE0 base address: 4003_3000h

XCACHE1 base address: 4003_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Cache Control (CCR)	32	RW	0000_0000h
4h	Cache Line Control (CLCR)	32	RW	0000_0000h
8h	Cache Search Address (CSAR)	32	RW	0000_0000h
Ch	Cache Read/Write Value (CCVR)	32	RW	0000_0000h

24.7.2 Cache Control (CCR)

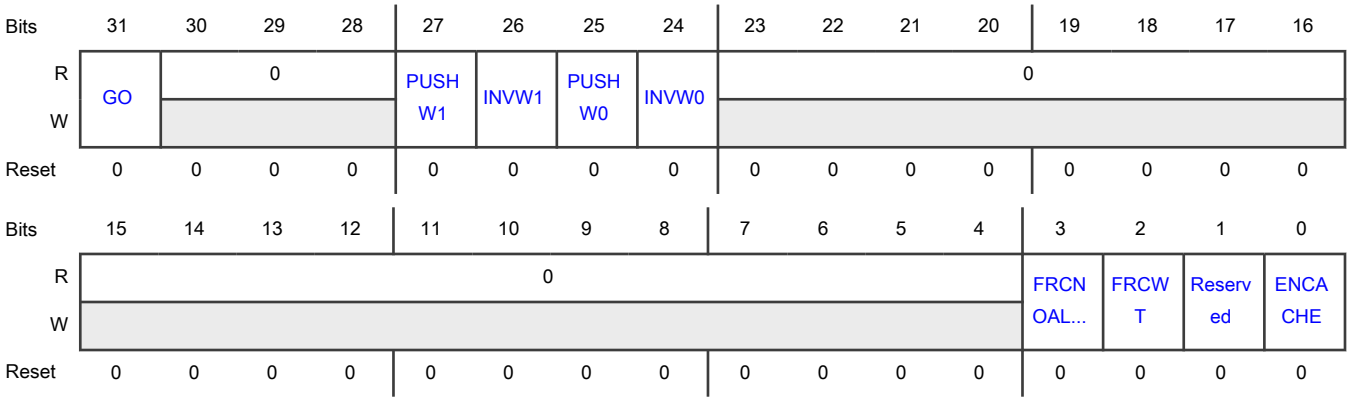
Offset

Register	Offset
CCR	0h

Function

Configures the cache control.

Diagram



Fields

Field	Function
31 GO	Initiate Cache Command Initiates the cache command that PUSHW1, INVW1, PUSHW0, and INVW0 indicate if you write 1 to this field. Reading this field indicates whether a command is active. **NOTE** This field remains 1 until the command completes. Writing 0 to it has no effect. 0b - Write: no effect; Read: no cache command active

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Write: initiate command; Read: cache command active
30-28 —	Reserved
27 PUSHW1	Push Way 1 Pushes all modified lines in way 1 when you write 1 to GO. 0b - No operation 1b - Push all modified lines in way 1
26 INVW1	Invalidate Way 1 Invalidates all lines in way 1 when you write 1 to GO. **NOTE** If PUSHW1 and INVW1 are 1, then after you write 1 to GO, this field pushes all modified lines in way 1 and invalidates all lines in way 1 (clear way 1). 0b - No operation 1b - Invalidate all lines in way 1
25 PUSHW0	Push Way 0 Pushes all modified lines in way 0 when you write 1 to GO. 0b - No operation 1b - Pushes all modified lines in way 0
24 INVW0	Invalidate Way 0 Invalidates all lines in way 0 when you write 1 to GO. **NOTE** If PUSHW0 and INVW0 are 1, then after you write 1 to GO, this field pushes all modified lines in way 0 and invalidates all lines in way 0 (clear way 0). 0b - No operation 1b - Invalidate all lines in way 0
23-4 —	Reserved
3 FRCNOALLC	Force No Allocation on Cache Misses Forces no allocation on cache misses when FRCWT is 1. 0b - Allocate on cache misses 1b - Force no allocation on cache misses

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 FRCWT	Force Write Through Mode Specifies whether all cacheable spaces to write through are forced. 0b - Not force 1b - Force
1 —	Reserved You must always write 0 to this field to maintain compatibility.
0 ENCACHE	Cache Enable Enables the cache. 0b - Disable 1b - Enable

24.7.3 Cache Line Control (CLCR)

Offset

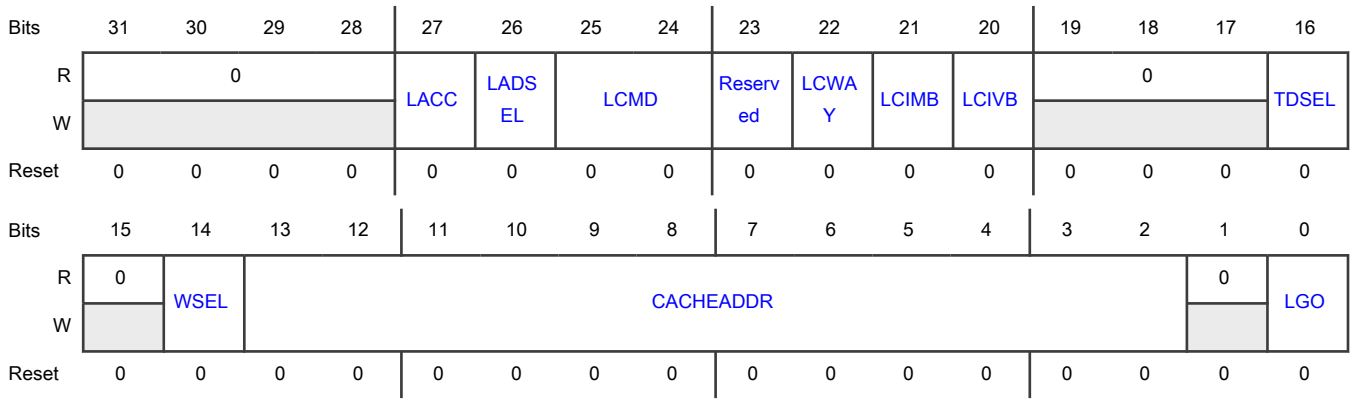
Register	Offset
CLCR	4h

Function

Defines the specific line-sized cache operations that you perform using a specific cache line address or physical address.

If you specify a physical address, both ways of the cache are searched, and the command only performs on the way that hits.

Diagram



Fields

Field	Function
31-28 —	Reserved
27 LACC	Line Access Type Specifies the line access type. 0b - Read 1b - Write
26 LADSEL	Line Address Select Specifies the address (cache or physical). When you use the cache address, you must also specify the way in CLCR[WSEL]. When you use the physical address, both ways are searched and the command performs only if there is a hit. 0b - Cache address 1b - Physical address
25-24 LCMD	Line Command Specifies the line command. 00b - Search and read or write 01b - Invalidate 10b - Push 11b - Clear
23 —	Reserved
22 LCWAY	Line Command Way Indicates the way that the line command uses. Applies only if CLCR[LCIVB] = 1.
21 LCIMB	Line Command Initial Modified Specifies the initial state of the modified bit. If the command uses cache address and way, then this field shows the initial state of the modified bit. If the command uses physical address and a hit, then this field shows the initial state of the modified bit. If the command uses a miss, then this field reads zero.
20 LCIVB	Line Command Initial Valid Specifies the initial state of the valid bit. If the command uses cache address and way, then this field shows the initial state of the valid bit.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If the command uses physical address and a hit, then this field shows the initial state of the valid bit. If the command uses a miss, then this field reads zero.
19-17 —	Reserved
16 TDSEL	Tag or Data Select Selects tag or data for search and read or write commands. 0b - Data 1b - Tag
15 —	Reserved
14 WSEL	Way Select Selects the way for line commands. 0b - Way 0 1b - Way 1
13-2 CACHEADDR	Cache Address Represents bits [13:2] of the cache address: CLCR [13:4] accesses the tag arrays. CLCR [13:2] accesses the data arrays.
1 —	Reserved
0 LGO	Initiate Cache Line Command Initiates the cache command that CLCR[LCMD], CLCR[LADSEL], and CLCR[LACC] indicate if you write 1 to this field. Reading this field indicates whether a line command is active. **NOTE** • This field remains 1 until the command completes. Writing 0 to it has no effect. • This field is shared with CSAR[LGO]. 0b - Write: no effect; Read: no line command active 1b - Write: initiate line command; Read: line command active

24.7.4 Cache Search Address (CSAR)

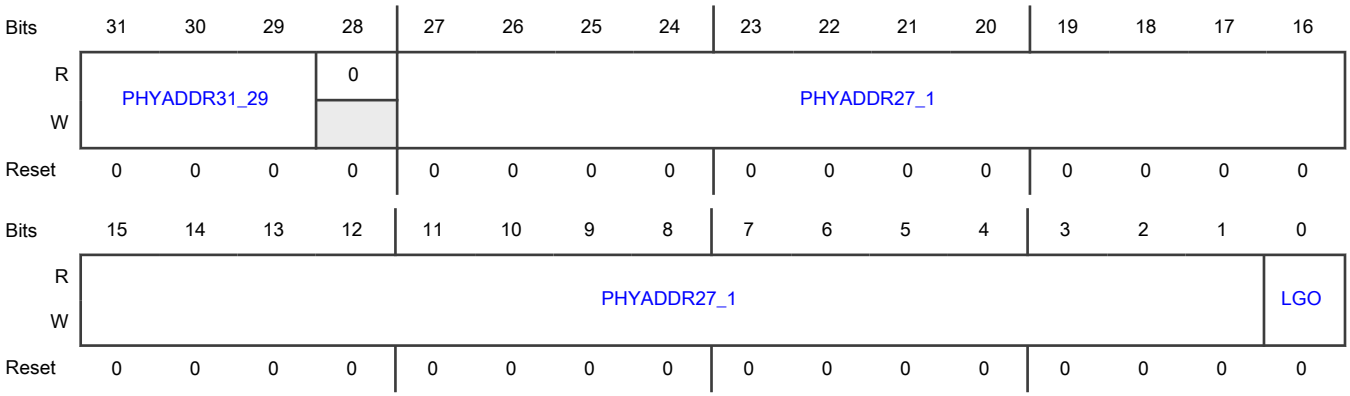
Offset

Register	Offset
CSAR	8h

Function

Defines the explicit cache address or physical address for line-sized commands specified in CLCR[LADSEL].

Diagram



Fields

Field	Function
31-29 PHYADDR31_29	Physical Address PHYADDR31_29 represents bits [31:29] and PHYADDR27_1 represents bits [27:1] of the system address. CSAR[31:14] is used for tag compare. CSAR[13:5] accesses the tag arrays. CSAR[13:3] accesses the data arrays. CSAR[2] specifies which half of the 64 bits of data to place in Cache Read/Write Value (CCVR). CSAR[1] is not used. Writes to this bit are allowed, and the read value is the previously written value. This bit is not functionally used.
28 —	Reserved Always read as zero.
27-1 PHYADDR27_1	Physical Address PHYADDR31_29 represents bits [31:29] and PHYADDR27_1 represents bits [27:1] of the system address. CSAR[31:14] is used for tag compare. CSAR[13:5] accesses the tag arrays.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	CSAR[13:3] accesses the data arrays. CSAR[2] specifies which half of the 64 bits of data to place in Cache Read/Write Value (CCVR). CSAR[1] is not used. Writes to this bit are allowed, and the read value is the previously written value. This bit is not functionally used.
0 LGO	Initiate Cache Line Command Initiates the cache line command that CLCR[LCMD], CLCR[LADSEL], and CLCR[LACC] indicate. Reading this field indicates whether a line command is active. **NOTE** • This field remains 1 until the command completes. Writing 0 to it has no effect. • This field is shared with CLCR[LGO]. 0b - Write: no effect; Read: no line command active 1b - Write: initiate line command; Read: line command active

24.7.5 Cache Read/Write Value (CCVR)

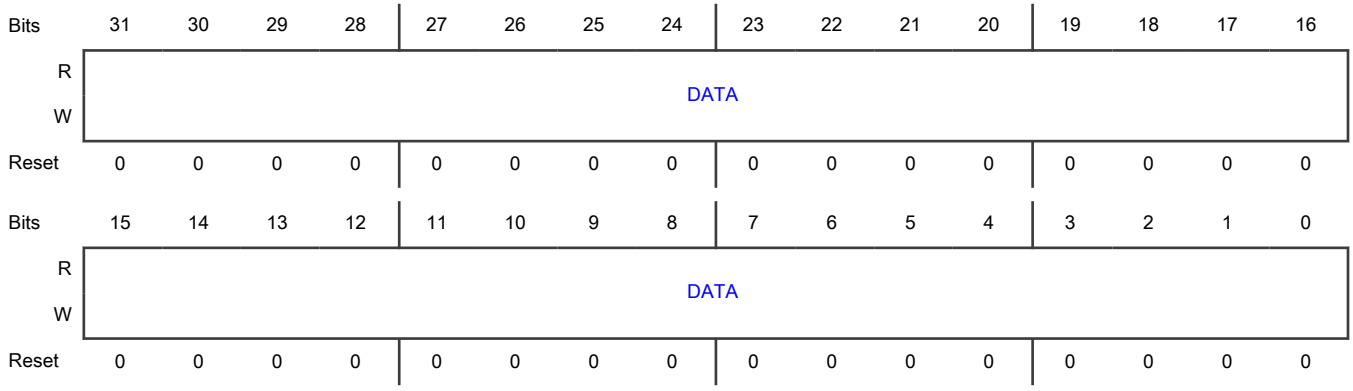
Offset

Register	Offset
CCVR	Ch

Function

Sources write data or returns read data for the commands specified in Cache Line Control (CLCR).

Diagram



Fields

Field	Function
31-0 DATA	Cache Read/Write Data Specifies the data that you want to read or write during tag search or data search. For tag search: • Read or write to CCVR [31:14] for tag array R/W value. • Read or write to CCVR [13:4] for tag set address on reads; unused on writes. • CCVR[3:2] are reserved. • CCVR[1] is tag modify bit. • CCVR[0] is tag valid bit. For data search: • Read or write to CCVR[31:0] for data array R/W value.

Chapter 25
Semaphores2 (SEMA42)

25.1 Chip-specific SEMA42 information

Table 208. Reference links to related information

Topic	Related module	Reference
Full description	SEMA42	SEMA42
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller

NOTE

The AHB ID in AHB bus matrix ports table is the Domain ID.

25.1.1 Module instances

This device contains three instances of the Semaphore module: SEMA42_0, SEMA42_3 and SEMA42_4.

Table 209. SEMA42 instances

	SEMA42_0	SEMA42_3	SEMA42_4
Located in	VDD1_SENSE	VDD1_SENSE	VDD2_COMP
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3	CPU1, HiFi1, eDMA2, eDMA3	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
Connection	• MU0_MUA (CPU0 to HiFi1) • MU0_MUB (HiFi1 to CPU0) • MU1_MUA (CPU0 to CPU1) • MU1_MUB (CPU1 to CPU0) • MU2_MUA (HiFi4 to CPU1) • MU2_MUB (CPU1 to HiFi4)	• MU3_MUA (CPU1 to HiFi1) • MU3_MUB (HiFi1 to CPU1)	• MU4_MUA (CPU0 to HiFi4) • MU4_MUB (HiFi4 to CPU0)

25.2 Overview

SEMA42 is a memory-mapped module that provides robust hardware support needed in multi-core systems for implementing semaphores and provides a simple mechanism to achieve "lock and unlock" operations via a single - write access. The hardware semaphore module provides hardware-enforced gates as well as other useful system functions related to the gating mechanisms.

NOTE

Terminology in this chapter has been updated as follows:

Table 210. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave
Stop	Abort

25.2.1 Block diagram

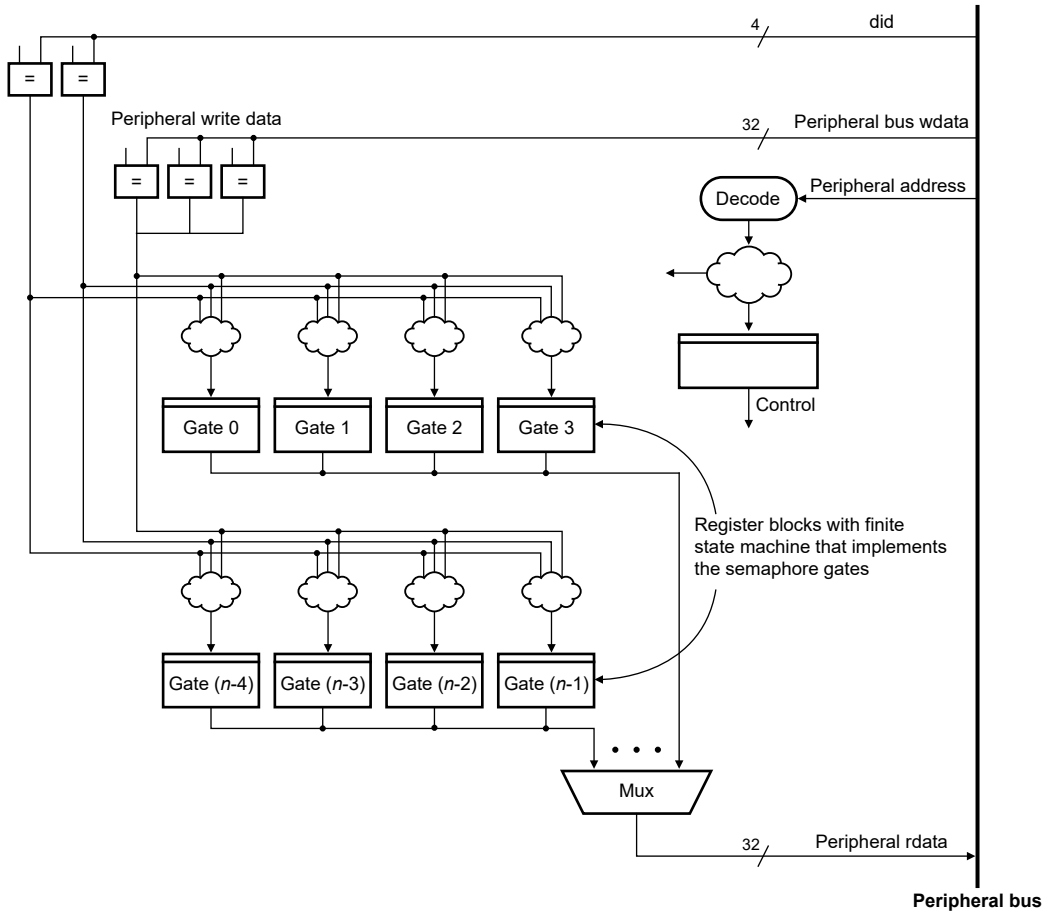


Figure 77. SEMA42 block diagram

25.2.2 Features

SEMA42 implements hardware-enforced semaphores as an IPS-mapped target peripheral device. The feature set includes:

- Support for 64 hardware-enforced gates in a multi-domain configuration that supports up to 15 domains.
- Each hardware gate appears as a 16-state, 4-bit state machine.
- Support for secure reset mechanisms to clear the contents of individual gates, as well as a clear-all capability.
- Memory-mapped target peripheral that offers programming-model accesses.

25.3 Functional description

The intent of the hardware gate implementation is to specify a protocol where the locking domain must unlock the gate. However, some systems may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset. To support this special gate reset requirement, SEMA42 implements a secure reset mechanism that allows you to initialize a hardware gate (or all the gates) by following a specific dual-write access pattern. The secure-gate reset:

- Uses a technique similar to that required for the servicing of a software watchdog timer
- Requires two consecutive writes with pre-defined data patterns from the same domain

You must do this to force the clearing of the specified gate(s). The required access pattern as follows:

1. A domain performs a 16-bit write to the RSTGT memory location. The most significant byte (RSTGT_W[RSTGDP]) must be E2h. The value of least significant byte is irrelevant for this reference and can be anything.
2. The same domain then performs a second 16-bit write to the RSTGT location. For this write, the upper byte (RSTGT_W[RSTGDP]) is the logical complement of the first data pattern (1Dh) and the lower byte (RSTGT_W[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate or all gates to be cleared. If the same domain writes incorrect data on the second access or another domain performs the second write access, SEMA42 stops the special gate reset sequence and does not assert an error signal.
3. Reads of the RSTGT location return information on the 2-bit reset state machine (RSTGT_R[RSTGSM]) that implements these functions:
   - The domain performing the reset (RSTGT_R[RSTGMS])
   - The last-cleared gate number(s) (RSTGT_R[RSTGTN])

   Reads of the RSTGT register do not affect the secure-reset finite state machine in any manner.

25.3.1 Multi-core programming: software gates

Multi-processor systems require a function that can be used to safely and easily provide a locking mechanism for system software to control access to shared data structures, shared hardware resources, and so on. The software uses the gating mechanisms to serialize (and synchronize) accesses to shared data and/or resources to prevent race conditions and preserve memory coherency between different processes and domains.

Consider the following description of a typical use-case: domain*X* enters a section of code, where shared data values are to be updated. The domain must first acquire a semaphore. Think of this as the locking (or closing) of a software gate. After the gate locks, a properly-architected software system does not allow other processes (or domains) to execute the same code segment or modify the shared data structure protected by the gate. In other words, the system locks out other processes/domains. Many software implementations include a spin-wait loop within the lock function until the gate locks. After domain X obtains the lock, domain X continues execution and updates the data values protected by the particular lock. After domain X completes the updates, it unlocks (or opens) the software gate, allowing other processes/domains access to the updated data values.

A correctly-implemented system solution must follow these important rules:

- A gate variable must protect all writes to shared data values or shared hardware resources.
- After a domain locks a gate, the system must block other processes/domains from accessing the shared data or resources. Software conventions enforce this.
- The domain that locks a particular gate is the only domain that can open (unlock) that gate.

Information in the hardware gate identifying the locking domain can be extremely useful for system-level debugging.

25.3.2 64 Hardware-enforced gates

Gates appear as a 64-entry byte-size array with read and write accesses.

Domains lock gates by writing "domainID_number+1" to the appropriate gate and must read back the gate value to verify that the lock operation succeeded.

After the gate locks, the locking domain unlocks the gate by writing zeroes.

- 16-state implementation

  If gate = 0h, then state = unlocked

  if gate = 1h, then state = locked by domain (initiator) 0

  if gate = 2h, then state = locked by domain (initiator) 1

  …

  if gate = Fh, then state = locked by domain (initiator) 14

SEMA42 uses the logical domain number and the specified data patterns as reference attributes to validate all write operation.

After a gate locks, the locking domain must unlock the gate by writing zeroes.

25.3.3 State machine of the GATE*n* registers

This section describes more about the SEMA42 functional operation and specific details of the state machines of the GATE*n* registers.

As described previously, each of the GATE*n* registers implements a 4-bit, 16-state machine. The following figure shows a simplified diagram of the state transitions for each gate.

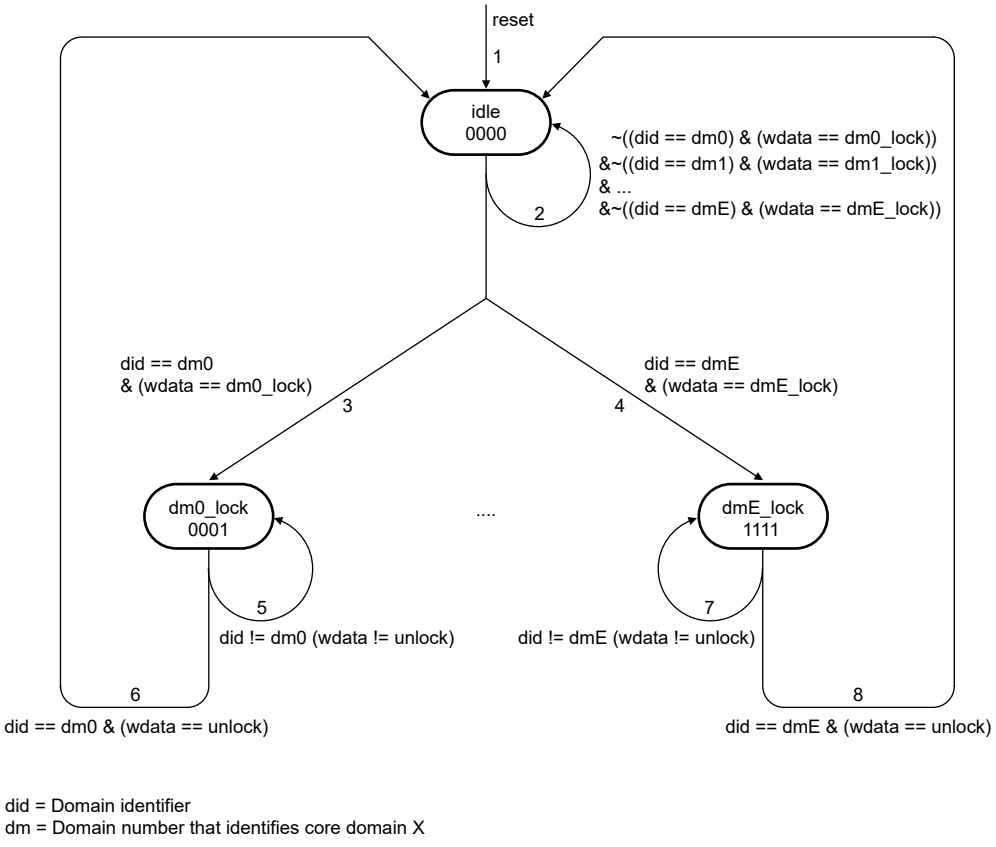


Figure 78. GATE*n* state machine

In the figure above, "dmE" represents domain 14 (E in hexadecimal). The platform passes the domain number to SEMA42.

The following table defines the GATE*n* state transitions.

Table 211. GATE*n* state transitions

Current state	Next state	Transition	Description
–	idle	1	Any reset, whether a system reset or a software-initiated gate reset, unconditionally forces the gate into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding domain occurs, the gate remains in the idle state.
idle	dm0_lock	3	When domain 0h initiates a write of the dm0_lock data value, the gate transitions into the dm0_lock state.
idle	dmE_lock	4	When domain Eh initiates a write of the dmE_lock value, the gate transitions into the dmE_lock state.
dm0_lock	dm0_lock	5	When in this state, the gate remains here if any attempted write is not from domain 0h with the unlock data value.
dm0_lock	idle	6	The gate returns to the idle (unlocked) state after a write from domain 0h with the unlock data value occurs.
dmE_lock	dmE_lock	7	When in this state, the gate remains here if any attempted write is not from domain Eh with the unlock data value.
dmE_lock	idle	8	The gate returns to the idle (unlocked) state after a write from domain Eh with the unlock data value occurs.

SEMA42 uses these gate data values:

- The lock data value is (domain number) + 1.
- The unlock data value is 00h.

25.3.4 Clocking

Type of clock	Description
Module clock (clk)	Functions the module and controls the access to the registers.

25.3.5 Interrupts

This module has no interrupts.

25.4 External signals

This module has no external signals.

25.5 Initialization

This module does not require initialization.

25.6 Memory map/register definition

You can access these registers only in Supervisor mode. User accesses terminate with an error.

25.6.1 SEMA42 register descriptions

25.6.1.1 SEMA42 memory map

SEMA42_0 base address: 4020_6000h

SEMA42_3 base address: 4031_B000h

SEMA42_4 base address: 4018_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 3Fh	Gate (GATE0 - GATE63) [1]	8	RW	00h
42h	Reset Gate Read (RSTGT_R)	16	R	0000h
42h	Reset Gate Write (RSTGT_W)	16	W	See section

1. In this array, the index and offset values of the registers do not increment in direct alignment. For details, see the register description.

25.6.1.2 Gate (GATE0 - GATE63)

Offset

For n = 0 to 63:

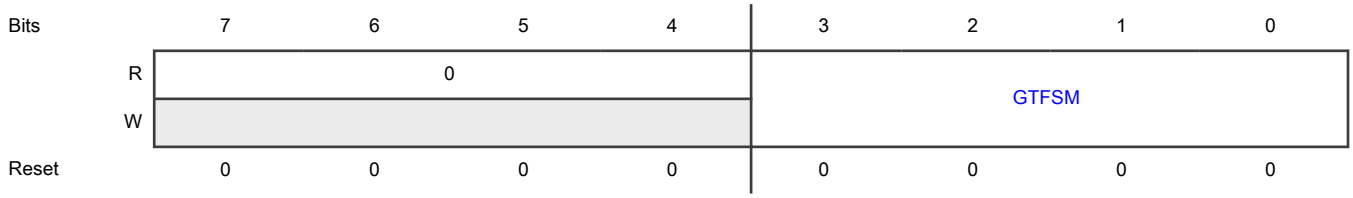
Register	Offset
GATEn	0h + (n + 3 - 2 × (n mod 4))

Function

Implements each semaphore gate in a 4-bit finite state machine, right-justified in a byte data structure. The hardware uses the logical domain-identifier number in conjunction with the data patterns to validate all attempted write operations. Only domain initiators can modify the gate registers. After a gate locks, only the locking domain must open (unlock) the gate.

You can read multiple gate values in a single access. However, you can update only a single gate at a time, via a write operation. If you attempt to write a byte-wide value that is neither the unlock value (00h) nor the appropriate lock value (domainID_number + 1), SEMA42 considers this as "no operation" and does not change any gate state. Attempts to write multiple gates in a single-aligned access with a size larger than 8 bits (byte) generate an error termination and do not allow any gate state changes.

Diagram



Fields

Field	Function
7-4 —	Reserved
3-0 GTFSM	Gate Finite State Machine Indicates the state of the gate for the last domain that locked the gate. This can be useful during system debug. The hardware gate has a 16-state implementation, defined as: 0000b - The gate is unlocked (free). 0001b - Domain 0 locked the gate. 0010b - Domain 1 locked the gate. 0011b - Domain 2 locked the gate. 0100b - Domain 3 locked the gate. 0101b - Domain 4 locked the gate. 0110b - Domain 5 locked the gate. 0111b - Domain 6 locked the gate. 1000b - Domain 7 locked the gate. 1001b - Domain 8 locked the gate. 1010b - Domain 9 locked the gate. 1011b - Domain 10 locked the gate. 1100b - Domain 11 locked the gate. 1101b - Domain 12 locked the gate. 1110b - Domain 13 locked the gate. 1111b - Domain 14 locked the gate.

25.6.1.3 Reset Gate Read (RSTGT_R)

Offset

Register	Offset
RSTGT_R	42h

Function

Describes the specific hardware gate to be reset and records the logic number of domain. Reset Gate Write (RSTGT_W) also describe the same register showing the fields when you write it.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		RSTGSM		RSTGMS				RSTGTN							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-14 —	Reserved
13-12 RSTGSM	Reset Gate Finite State Machine Indicates the encoded state machine value when you read the register. RSTGSM = 10b is valid for only a single machine cycle, so a read can never return this value. SEMA42 maintains the reset state machine in a 2-bit, 3-state implementation, defined as follows: 00b - Idle, waiting for the first data pattern write. 01b - Waiting for the second data pattern write 10b - The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. 11b - This state encoding is never used and therefore reserved.
11-8 RSTGMS	Reset Gate Domain Records the logical number of the domain performing the gate reset function. To succeed, this function requires that the same domain initiate the two consecutive writes to this register. SEMA42 updates the field each time a write to this register occurs.
7-0 RSTGTN	Reset Gate Number Specifies the specific hardware gate to be reset. The second write updates this field. If RSTGTN < 64, SEMA42 resets the single gate defined by RSTGTN. Otherwise, SEMA42 resets all the gates.

25.6.1.4 Reset Gate Write (RSTGT_W)

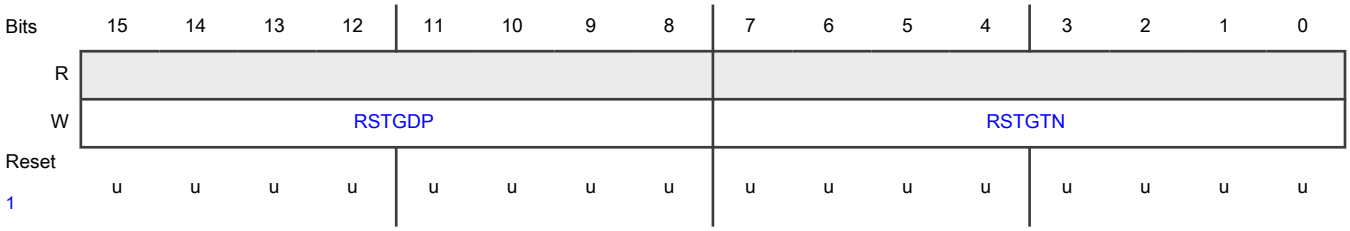
Offset

Register	Offset
RSTGT_W	42h

Function

Specifies the hardware gate to reset when data patterns are specified.

Diagram



1. Reset value is not applicable for writes.

Fields

Field	Function
15-8 RSTGDP	Reset Gate Data Pattern You access this field with the specified data patterns on the two consecutive writes to enable the gate-reset mechanism. For the first write, RSTGDP must be E2h. For the second write, RSTGDP must be 1Dh.
7-0 RSTGTN	Reset Gate Number Specifies the specific hardware gate to be reset. The second write updates this field. If RSTGTN < 64, SEMA42 resets the single gate defined by RSTGTN. Otherwise, SEMA42 resets all the gates.

Chapter 26
Messaging Unit (MU)

26.1 Chip-specific MU information

Table 212. Reference links to related information

Topic	Related module	Reference
Full description	MU	MU
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Power management	PMC	Power overview PMC
Interrupts		Interrupts and DMA events assignments

26.1.1 Module instances

This device has five instances of the MU module.

Table 213. MU instances

MU instances	Located in	Access	Connection	Related SEMA42 instances
MU0_MUA	VDD1_SENSE	CPU0	CPU0 to HiFi1	SEMA42_0
MU0_MUB	VDD1_SENSE	HiFi1	HiFi1 to CPU0	
MU1_MUA	VDD1_SENSE	CPU0	CPU0 to CPU1	
MU1_MUB	VDD1_SENSE	CPU1	CPU1 to CPU0	
MU2_MUA	VDD1_SENSE	HiFi4	HiFi4 to CPU1	
MU2_MUB	VDD1_SENSE	CPU1	CPU1 to HiFi4	
MU3_MUA	VDD1_SENSE	CPU1	CPU1 to HiFi1	SEMA42_3
MU3_MUB	VDD1_SENSE	HiFi1	HiFi1 to CPU1	
MU4_MUA	VDD2_COMP	CPU0	CPU0 to HiFi4	SEMA42_4
MU4_MUB	VDD2_COMP	HiFi4	HiFi4 to CPU0	

26.1.2 Power mode

The following table shows the mapping relations of the power modes used in this chapter to those defined for this chip .

Table 214. Power mode mapping

Power mode in this chapter	Power mode for this chip
Run	Active mode
Wait[1]	Sleep mode
Halt, Stop, Power down	Not supported, ignore all the related content in this chapter

1. For this chip, the low-power mode in this chapter refers to wait mode.

26.1.3 CCR0[BOOT]

For this chip, the CCR0[BOOT] is reserved for each instance.

26.2 Overview

MU enables two processors on a chip to communicate and coordinate by passing messages (for example, data, status, and control) through the MU interface. MU also allows one processor to signal the other processor using interrupts.

MU must synchronize the accesses from one side to the other because MU can manage messaging between processors using different clocks. MU accomplishes synchronization using two sets of matching registers: processor A-facing and processor B-facing.

26.2.1 Block diagram

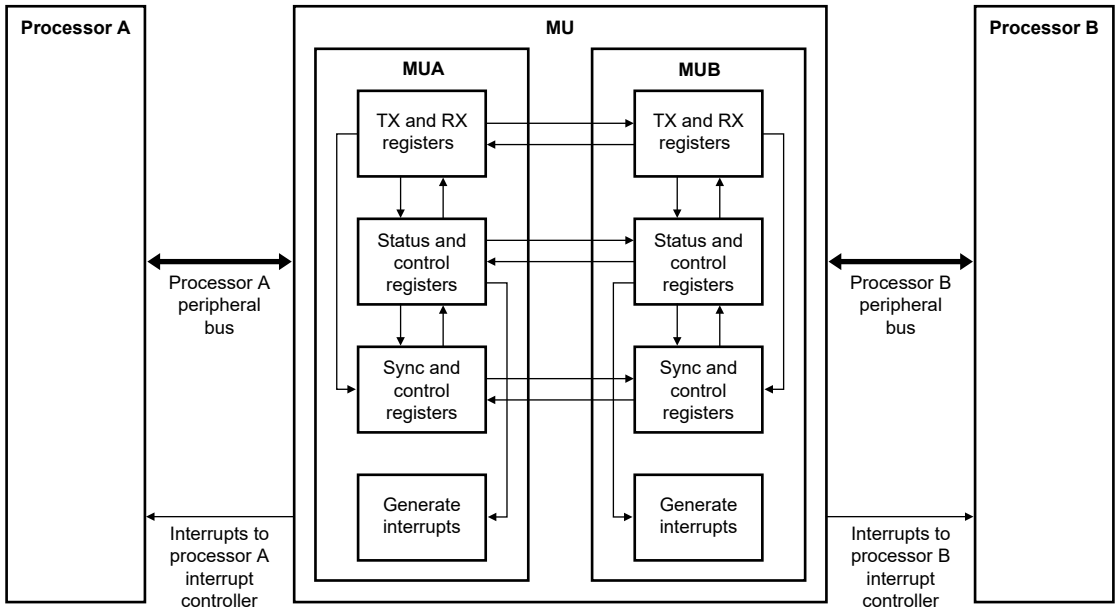


Figure 79. Block diagram

26.2.2 Features

- Memory-mapped registers (MU is connected as a peripheral under the peripheral bus on the processor A-side and processor B-side)
- Synchronized message transfers between cores
  - To send data or messages from one side to the other, messaging unit A (MUA) provides 4 transmit registers and 4 receive registers. Messaging unit B (MUB) provides 4 transmit registers and 4 receive registers.

  - Transmit empty flags (TSR[TEn]) and receive full flags (RSR[RFn]) facilitate the transfer of data or messages between cores on both sides of MU.
  - A synchronization mechanism updates the transmit and the receive flags. There is an inherent latency between updating the flag on one side and reflecting its status on the other side. See Event update timing.
  - MU has a 3-bit flag data register, which you can use to send flag data between the two MU sides.
- Interprocessor interrupts
  - MU has 12 interrupt sources on each side (processor A-side, processor B-side) for signaling the other processor. You can use the interrupts for notification of receive and transmit events and for general-purpose signaling between processors. There are 4 general-purpose interrupt requests available and 8 receive and transmit interrupt sources.
  - Processor A can initiate a non-maskable interrupt (NMI) to processor B.
  - Processor B can initiate a non-maskable interrupt (NMI) to processor A.
- Reset (Each processor can issue a reset to MU via CR[MUR], which is a self-clearing field).

26.3 Functional description

MU enables two cores (processor A and processor B) to communicate with each other:

- By sharing messages and data.
- By enabling one core to wake the other core by using interrupts.

The messaging, control, and status registers of the two cores are mapped to processor A memory and processor B memory as a regular peripheral. The peripheral data bus is 32 bits wide inside MU.

Messaging logic is used with shared memory. Various messaging methods can implement a messaging protocol. For example, a message could mean one of the following:

- A message of *n* words has been written starting at offset *x* in the memory.
- The previous data block that was sent has been read.

The ability to keep messaging logic independent of the shared memory is not restricted to a predefined hardware protocol. The software required to manage the messaging is short and straightforward.

Most of the messaging mechanisms are symmetric. They are duplicated and are available on both the processor A side and processor B side. The messaging mechanisms are:

- 4 32-bit transmit registers, which are each reflected in 4 read-only receive registers on the side of the other processor. These registers can transfer 32-bit word messages or the frame information of the messages written to the shared memory. For example, they can transfer the number of words, initial address, and message type code.
- Writing to a transmitter-side transmit register (TR*n*) clears the Transmitter Empty flag in the transmitter-side Status register (TS*n*[TE*n*]), and sets the Receiver Full flag in the receiver-side Status register (RS*n*[RF*n*]). Setting the flag on the receiver side can trigger an interrupt on the receiver side (a maskable receive interrupt).
- Reading a receiver-side receive register (RR*n*) clears the Receiver Full flag in the receiver-side Status register (RS*n*[RF*n*]), and sets the Transmitter Empty flag in the transmitter-side Status register (TS*n*[TE*n*]). Setting the Transmitter Empty flag can trigger an interrupt on the transmitter side (a maskable transmit interrupt).
- 4 general-purpose interrupt request flags are reflected in General-purpose Status (GSR) on the receiver side.
- 3-bit flag data is transmitted from Flag Control (FCR) to Flag Status (FSR) on the side receiving the flag data. SR[FUP] sets when the flag data is transmitted and clears when the receiving side acknowledges the flag data (the flag is updated).

Writing to a transmit register signals to the receiver side that data is ready for retrieval.

Do not write to the transmit register again without verifying that the data is retrieved. The transmitter side cannot determine the exact time that the receiver attempts to retrieve the data. Before attempting to write to the transmit register again, the transmitter side must wait for the Transmitter Empty interrupt or it must poll TSR[TE*n*].

Reading a receive register signals to the transmitter side that data can be written to that register.

Do not read the receive register again without verifying that the data is written. The receiver side cannot determine the exact time that the transmitter attempts to write the data. Before attempting to read the receive register again, the receiver side must wait for the Receiver Full interrupt or it must poll RSR[RF*n*].

26.3.1 Submodules

26.3.1.1 MUA side

MUA receives its register configuration via the processor A peripheral bus. It sends or receives messages to or from MUB. Processor A can receive messages by reading MUA registers, and MUA can send interrupts to processor A when interrupts are enabled.

NOTE

Processor B should not access MUA registers. Processor C, if exist, should also not access MUA registers. TRDC might be needed to prevent such illegal access of Processor B or C to MUA registers.

26.3.1.2 MUB side

MUB receives its register configuration via the processor B peripheral bus. It sends or receives messages to or from MUA. Processor B can receive messages by reading MUB registers, and MUB can send interrupts to processor B when interrupts are enabled.

NOTE

Processor A should not access MUB registers. Processor C, if exist, should also not access MUB registers. TRDC might be needed to prevent such illegal access of Processor A or C to MUB registers.

26.3.2 Event update timing

The messaging side of each processor has a hardware mechanism to send event update requests to the other processor. An event occurs when the status register of the receiving processor must reflect any information change. The event update latency is the delay between the event being ready at one processor and the resulting update at the status register of the other processor:

- The minimum event latency is (1 clock cycle of the sending side) + (2.5 clock cycles of the receiving side). This minimum case happens when no event is pending when the new event occurs.
- The maximum event latency is (6 clock cycles of the sending side) + (6.5 clock cycles of the receiving side). The maximum case happens when the event occurs just after a previous event is sent to the other side.

The event update latency varies depending on the time at which the subsequent event is triggered.

26.3.3 Clocking

Table 215. MU clocks

Clock name	Description
Bus clock MUA	Used only for bus accesses to MUA control and configuration registers
Bus clock MUB	Used only for bus accesses to MUB control and configuration registers

26.3.4 Resets

The following sections list the reset sources included in MU. Each reset performs a different function for the MU module compared to the function it performs for the system.

26.3.4.1 Asynchronous system reset

When the asynchronous system reset on either side of MU is asserted, SR[MURS] becomes 1 until the asynchronous system reset sequence on both the MUA and MUB sides ends. Verify that SR[MURS] becomes 0 before accessing MU.

The asynchronous system reset on one side of MU resets the other side of MU. The reset forces all control and status registers to return to their default values and clears all internal states. The following table shows the exceptions to this behavior.

Table 216. Exceptions to asynchronous system reset

MUA-side exceptions	MUB-side exceptions
MUB_SR[MURS] MUB_CSSR0/MUB_CSR0	MUA_SR[MURS] MUA_CSSR0/MUA_CSR0

26.3.4.2 MU software reset

Writing 1 to CR[MUR] causes most control and status registers to return to their default values and clears all internal states.

The instruction immediately following the assertion of CR[MUR] must not write to the MU registers. The reset sequence may overwrite such a write, with the register retaining the reset value. To know the end of the reset sequence for both processors, monitor the value of SR[MURS]. After the reset sequence on both processors has ended, a write to the MU registers can be attempted.

NOTE

The process of CR[MUR] becoming 1 is delicate because it asynchronously affects the registers on the other side. CR[MUR] becoming 1 may cause unpredictable behavior if, for example, the other processor is concurrently testing an MU register field (TSR[TE*n*] in the other processor). Before writing 1 to CR[MUR], verify that the other processor is not engaged in an MU signaling activity.

26.3.5 Interrupts

MU controls interrupt requests that one processor makes to the other processor. This section describes all the interrupts that the module generates.

MU can generate these interrupt sources individually to send to the processors:

- 4 receive interrupts (asserted when the Receive Full flags are set in Receive Status (RSR) and enabled in Receive Control (RCR)) for each receive register
- 4 transmit interrupts (asserted when the Transmit Empty flags are set in Transmit Status (TSR) and enabled in Transmit Control (TCR)) for each transmit register
- 4 general-purpose interrupts (asserted when the GIP flags are set in General-Purpose Interrupt Enable (GIER) and enabled in General-Purpose Interrupt Enable (GIER))
- One non-maskable interrupt (NMI) flag (asserted when the NMI flag is set in Core Control 0 (CCR0))

All interrupts except NMI are maskable in the processor control registers: TCR, RCR, GIER, and CR. MU does not assume any internal priority of these interrupts. Multiple interrupts (for example, receive 0 and receive 1, or any transmit or general-purpose interrupts) can be asserted simultaneously. The interrupt controller must resolve the priority of these interrupts at the chip level.

Triggering any enabled interrupt wakes the processor from below mode before servicing the interrupt

- Wait mode

The software (as part of the interrupt handler) must clear the general-purpose interrupt pending flags (GSR[GIP*n*]) to deassert the request to the interrupt controller.

The software must also clear the NMI by writing CSSR0[NIMC] (as a part of the interrupt handler) to deassert the NMI request.

When a processor writes to the general-purpose interrupt (GCR[GIR*n*]), MU synchronizes the write event to the other processor to set the general-purpose interrupt request pending flag (GSR[GIP*n*]). When GSR[GIP*n*] is set and the general-purpose interrupt is enabled on the receiving side (GIER[GIE*n*] is 1), the transmitting-side general-purpose interrupt is issued to the receiving processor, which clears this interrupt by writing 1 to GSR[GIP*n*]. The interrupt is deasserted as soon as the write to GSR[GIP*n*] occurs. MU synchronizes the write event of GSR[GIP*n*] to the other processor. The synchronized signal writes 0 to the GIR*n* interrupt.

Before writing 1 to GCR[GIR*n*], verify that the field is 0, which means that a general interrupt is not pending. Generally, MU ignores writing 1 to this field while the field is already 1, but in some cases it may issue a second interrupt.

26.4 External signals

This module has no external signals.

26.5 Initialization

MU does not require initialization.

26.6 Application information

MU facilitates messages between processors. For example, MU passes:

- Short messages. Transmit registers can pass short messages from 1 to 4 words in length for processor A and from 1 to 4 words for processor B. For example, for a four-word message, only one register must have its corresponding interrupt enabled on the receiving side. The first three words of the message are written to the registers with masked interrupts. The fourth word is written to the last register, triggering an interrupt on the receiving side.
- Frame information. Transmit registers can pass frame information for long messages written to the shared system memory. Such frame information normally includes a start address and a number of words. It can include a message type code.
- Event notices and requests. MU can signal events and requests that do not include data words between processors using general-purpose interrupts. For example, one such event is acknowledging that a long message is read from the shared system memory.
- Fixed-length data. Formatted data with a fixed length can be written to predetermined locations in the shared memory. A processor can use general-purpose interrupts to signal to the other processor that the data is ready.
- Announcements. A processor can use the 3 flags to announce its current program state or other billboard messages to the other processor.

26.6.1 Messaging protocols using interrupts

The example below describes a four-word messaging sequence sent between the processor A and processor B.

The transmitting processor writes to the transmit registers sequentially. When *n* = 0, 1, and 2, the interrupts are disabled, so no interrupt goes to the processor (although interrupt conditions occur). For *n* = 3, the interrupt is enabled, and MU generates the last receive interrupt request.

1. Write sequence:
    a. The transmitting processor writes the message information sequentially to its Transmit registers 0, 1, 2.
    b. When the write to Transmit register 3 occurs, RSR[RF3] is set after synchronization. It immediately triggers the receive full 3 interrupt on the receiving processor.
2. Read sequence:
    a. The receiving processor receives the receive full 3 interrupt and starts reading the message transferred from the receive registers.
    b. After the receiving processor reads Receive register 3, MU clears RSR[RF3].

Table 217 and Figure 80 describe the messaging model, using transmit and receive registers and the interrupt messaging protocol.

Table 217. Interrupt messaging protocol (generalized)

Step	Action	Description
1	Processor A writes data.	RR*n* on processor B's side reflects a data write to TR*n* by processor A.
2	Clear the transmitter empty flag and set the receiver full flag.	The data write to TR*n*: • Clears TSR[TE*n*] on the processor A side. • Sets RSR[RF*n*] on the processor B side.
3	Generate the receive interrupt request.	Setting RSR[RF*n*] generates a receive interrupt request to processor B.
4	Processor B reads the data.	After receiving the receive interrupt request, processor B performs a data read of RR*n*.
5	Clear the receiver full flag and set the transmitter empty flag.	Reading the data from the RR*n* register: • Clears RSR[RF*n*] on the processor B side. • Sets TSR[TE*n*] on the processor A side.
6	Generate the transmit interrupt request.	Setting TSR[TE*n*] generates a transmit interrupt request to processor A.

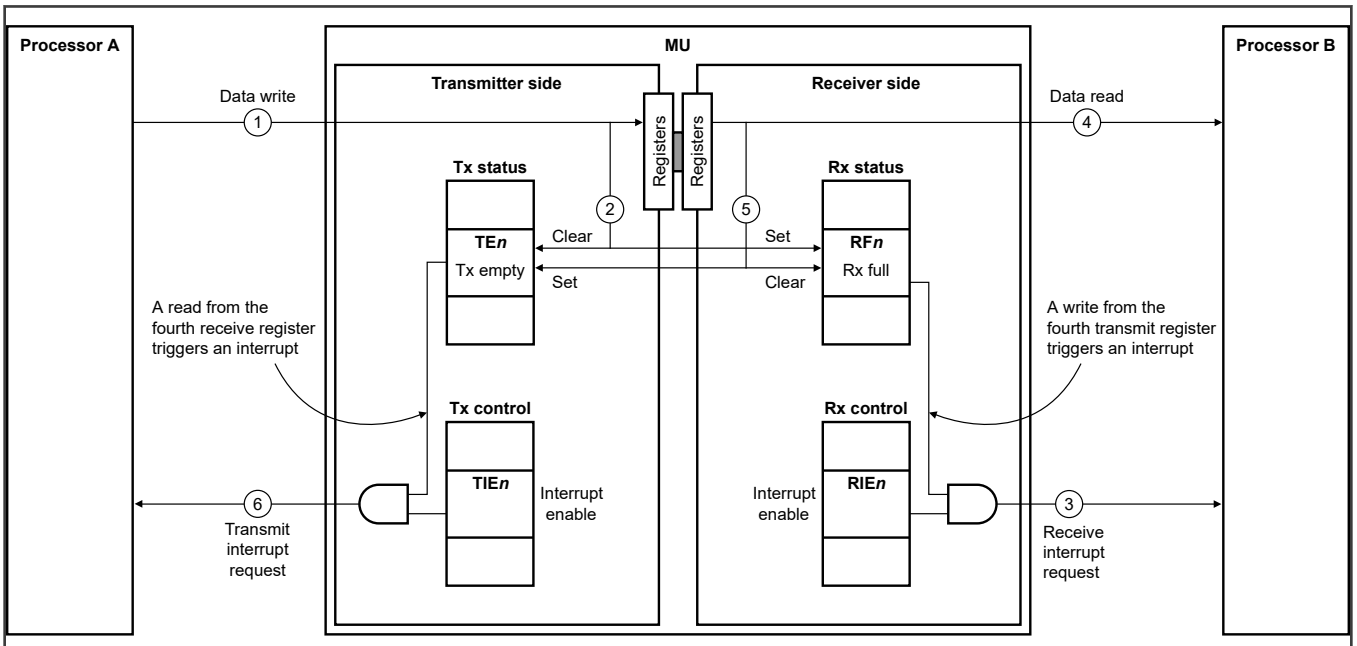


Figure 80. Messaging model using transmit and receive registers

You can use the messaging hardware to implement messaging protocols for an array of message types. MU provides full support for interrupt and polling management schemes.

26.6.2 Messaging protocols using event interrupts

MU can signal events and requests that do not include data words between processors using general-purpose interrupts.

Formatted data with a fixed length can be written to predetermined locations in the shared memory. A processor can use a general-purpose interrupt to signal to the other processor that the data is ready.

A processor can use the 3 flags to announce its current program state (or similar messages) to the other processor.

Table 218 and Figure 81 describe the event steps when the processor triggers a general-purpose interrupt.

Table 218. General-purpose interrupt messaging protocol (generalized)

Step	Action	Description
1	Processor A sets its associated general-purpose interrupt request flag.	Processor A sets GCR[GIR*n*].
2	The general-purpose interrupt request pending status flag is set.	GSR[GIP*n*] is set.
3	MU generates the general-purpose interrupt request to processor B.	Setting GSR[GIP*n*] generates the general-purpose interrupt request to processor B. GIER[GIE*n*] must be set for processor B.
4	Processor B reads the status register.	Processor B reads GSR[GIP*n*].
5	Processor B services the interrupt.	—
6	Processor B sets GSR[GIP*n*] to clear the interrupt.	Processor B writes 1 to the corresponding GSR[GIP*n*] flag to clear the interrupt.
7	GCR[GIR*n*] is cleared.	Setting GSR[GIP*n*] clears GCR[GIR*n*] on the processor A side.

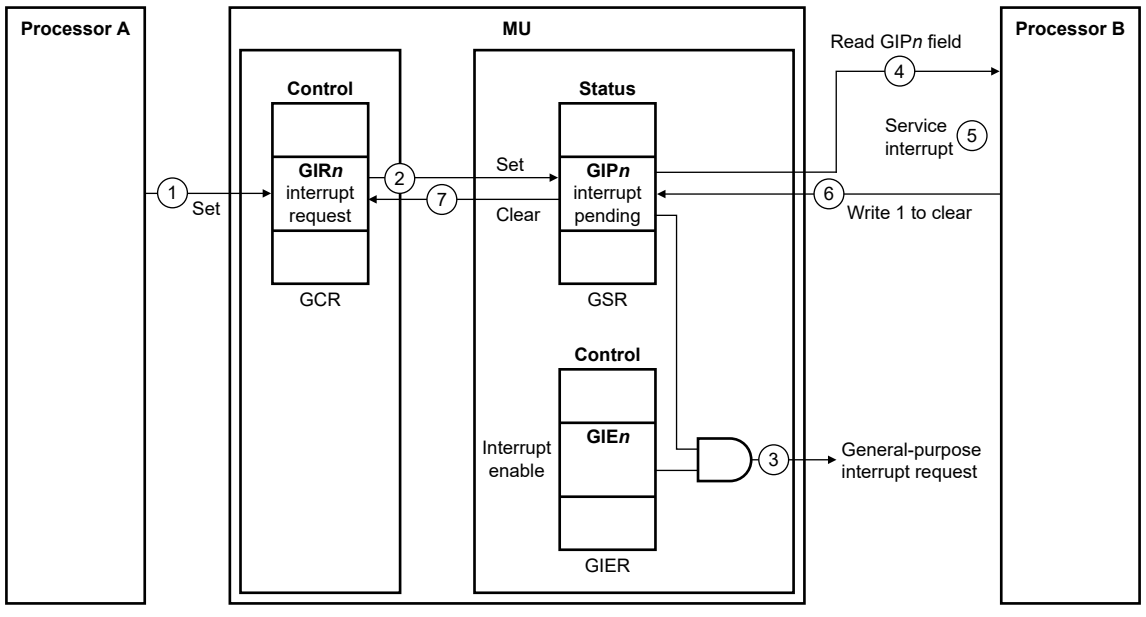


Figure 81. Messaging model using a general-purpose interrupt

26.6.3 Exclusive access to shared memory

MU can signal one processor about its current access to shared memory. This signaling prevents the other processor from overwriting data during the exclusive memory access period.

The following tables describe the signaling protocol that processor A uses to inform processor B about its current access (write) to shared memory:

- Table 219 shows processor A performing an exclusive access to shared memory.
- Table 220 shows processor B scanning for transaction information.
- Table 221 shows processor B accepting exclusive access from processor A.
- Table 222 shows processor B rejecting exclusive access from processor A.

According to the examples shown in the following tables, GCR[GIR*n*], RR*n*, and TR*n* are reserved to support exclusive access to the shared memory protocol.

Table 219. Processor A performs an exclusive access to shared memory

Step	Action	Description
1	Processor A sends the GIR*n* request to processor B using the processor A control register.	Before processor A performs an exclusive access to the shared memory, it sends a GIR*n* request to processor B.
2	Processor A sends an exclusive-access request using a transmit data register (TR*n*).	Processor A sends an exclusive-access request (command, location, and length of target access) to processor B using a selected transmit data register (TR0).
3	Processor A waits for a dedicated interrupt from processor B.	Processor A waits for a dedicated interrupt (as an acknowledgment) triggered by processor B before proceeding.
4	Processor A accesses the shared memory.	After receiving a dedicated interrupt from processor B, processor A proceeds.

Table 220. Processor B scans for transaction information

Step	Action	Description
1	Processor B receives an interrupt from a receive data register (RR*n*).	—
2	Processor B reads the receive data register (RR*n*).	—
3	Processor B scans the receive data register contents.	Processor B scans for transaction information (whether processor A has requested exclusive access).

Table 221. Processor B accepts exclusive access from processor A

Step	Action	Description
1	Processor B triggers a dedicated interrupt.	Processor B acknowledges the request from processor A by triggering a dedicated interrupt (ack) to processor A.
2	Processor B sends a code message to processor A.	With the acknowledge interrupt, processor B sends a code message to processor A through the selected transmit register (TR*n*). The message informs processor A that it can exclusively access the shared memory.

Table 222. Processor B rejects exclusive access from processor A

Step	Action	Description
1	Processor B ignores the request from processor A for exclusive access.	If processor B does not provide permission to processor A, processor B ignores the exclusive-access request.

26.6.4 Packet data transfers

The following table describes an example packet transfer sequence between processor B and processor A subsystems.

Table 223. Packet data transfer sequence

Step	Action	Description
1	Processor B requests DMA.	Processor B sends a DMA request to initiate the packet data transfer.
2	DMA data transfer	DMA acknowledges.
3		DMA starts transferring data from the specified processor B memory location to the specified shared memory.
4		DMA interrupts processor B to signal that the packet transfer has finished.
5	Processor B informs processor A that data is in shared memory.	Using a B-side transmit register, processor B sends a packet information message to processor A about the arrival of new packet data stored in shared memory. The message contains the command, location, and length of packet data.
6	Processor A receives an interrupt.	Processor A receives an interrupt (assuming its corresponding processor A MU-side receive interrupt is enabled). The pending processing task becomes active and processes packet data from memory.
7	Processor A reads data, then writes data.	Processor A reads or processes packet data from shared memory.
8		Processor A writes the result from packet processing to a separate buffer.
9	Processor A informs processor B that the transfer is finished.	After the processing of the packet data finishes, processor A informs processor B (using MU processor A-side transmit register, MUA_TR*n*).
10	Processor A sends an interrupt to processor B (a request for more data).	Processor B receives the next interrupt from processor A, in which processor A requests more packet data.

26.6.5 Freeing a processor from deadlock

During normal operation, one processor may determine that the other processor is not working or is deadlocked. Using Status (SR), the processor can use the methods in the following table to identify and correct the problem.

Table 224. Freeing a processor from deadlock

Processor mode	Technique	Description
—	Processor issues an interrupt.	The other processor can interrupt the processor by issuing one of the 12 (general purpose, receive, or transmit) interrupts.
Normal	Processor issues a non-maskable interrupt.	If a processor is in Normal mode, the other processor can issue a non-maskable interrupt (NMI). You can configure the NMI routine to execute a diagnostic routine that is predefined for such an event.

26.7 Register definition

MU provides transmit and receive data registers (*x*TR0–*n*, *x*RR0–*n*) for communication between processor A and processor B. It also provides control registers (*x*CR) for operations such as interrupts and resets, and status registers (*x*SR) for checking the status of the other MU-side. Figure 82 shows the schematic for MU registers.

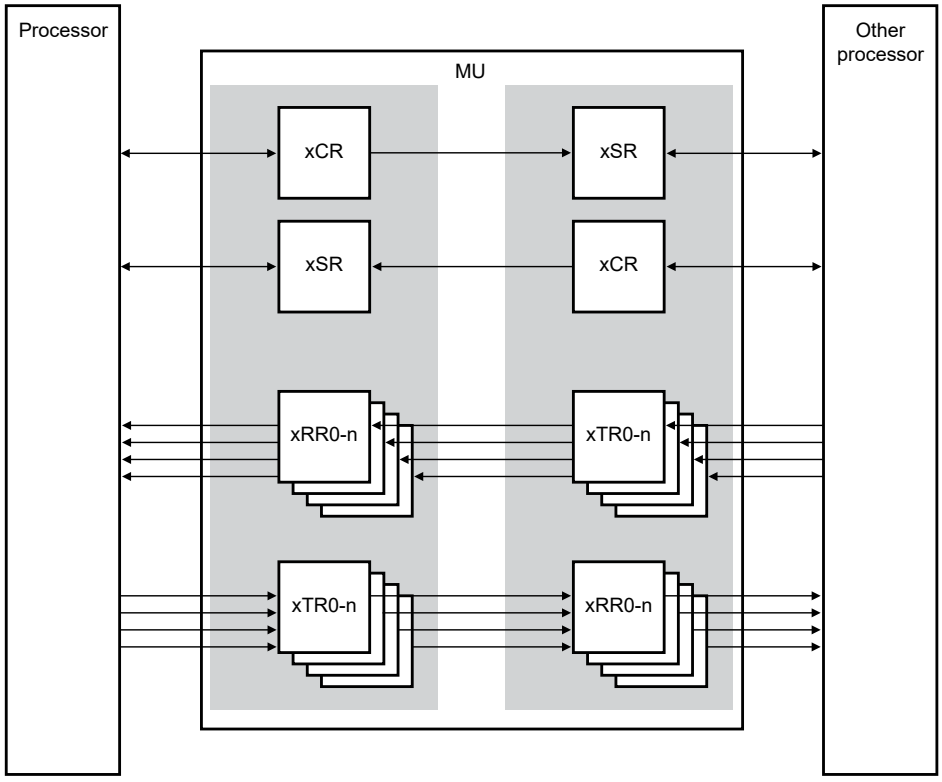


Figure 82. MU registers

26.7.1 MUA register descriptions

This section contains the detailed register descriptions for MUA registers.

NOTE

A module transfer error to processor A or processor B is generated when:

- A read or write access is made to an invalid location.
- A write operation is performed on a read-only register on the processor A side or processor B side of MU.

The only exception to this behavior is the MUA SR register. No transfer error occurs when accessing this location.

26.7.1.1 MUA memory map

MU0.MUA base address: 4020_0000h

MU1.MUA base address: 4020_2000h

MU2.MUA base address: 4020_4000h

MU3.MUA base address: 4031_9000h

MU4.MUA base address: 4018_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VER)	32	R	0309_000Fh
4h	Parameter (PAR)	32	R	0304_0404h
8h	Control (CR)	32	RW	0000_0000h
Ch	Status (SR)	32	R	See section
10h	Core Control 0 (CCR0)	32	RW	0000_0014h
14h	Core Interrupt Enable 0 (CIER0)	32	RW	0000_0000h
18h	Core Sticky Status 0 (CSSR0)	32	RW	See section
1Ch	Core Status 0 (CSR0)	32	R	0000_0000h
100h	Flag Control (FCR)	32	RW	0000_0000h
104h	Flag Status (FSR)	32	R	0000_0000h
110h	General-Purpose Interrupt Enable (GIER)	32	RW	0000_0000h
114h	General-Purpose Control (GCR)	32	RW	0000_0000h
118h	General-purpose Status (GSR)	32	RW	0000_0000h
120h	Transmit Control (TCR)	32	RW	0000_0000h
124h	Transmit Status (TSR)	32	R	0000_000Fh
128h	Receive Control (RCR)	32	RW	0000_0000h
12Ch	Receive Status (RSR)	32	R	0000_0000h
200h - 20Ch	Transmit (TR0 - TR3)	32	RW	0000_0000h
280h - 28Ch	Receive (RR0 - RR3)	32	R	0000_0000h

26.7.1.2 Version ID (VER)

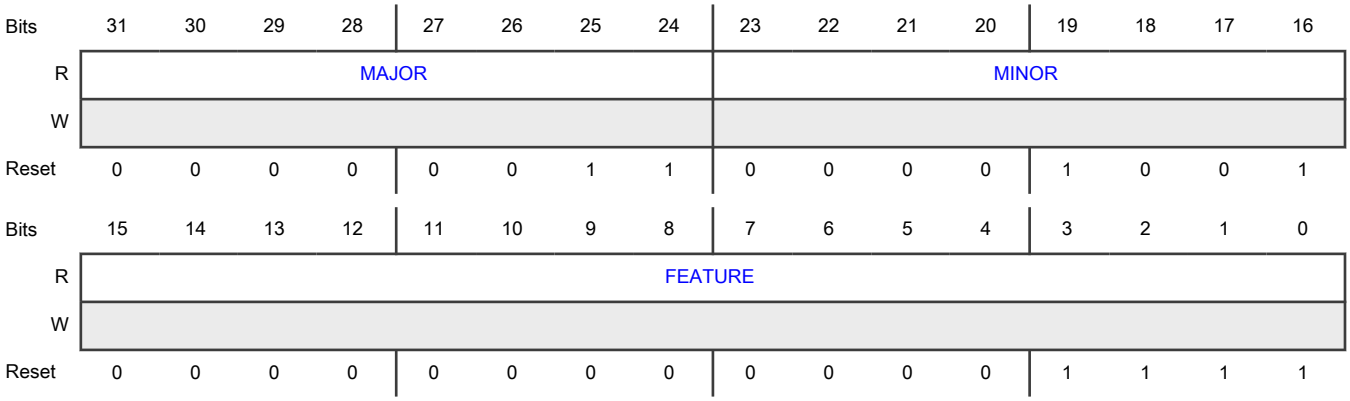
Offset

Register	Offset
VER	0h

Function

Determines the version ID and feature set number of MUA.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number
23-16 MINOR	Minor Version Number
15-0 FEATURE	Feature Set Number Indicates the feature set number. Each bit of this field specifies a different feature: • If FEATURE[0] = 1: Standard features are implemented. • If FEATURE[1] = 1: RAIP and RAIE fields are implemented. • If FEATURE[2] = 1: Core Control and Status registers are implemented in MUA and MUB. • If FEATURE[3] = 1: Expand the number of TR*n* and RR*n* registers. • FEATURE[4] - FEATURE[15]: Reserved for future use.

26.7.1.3 Parameter (PAR)

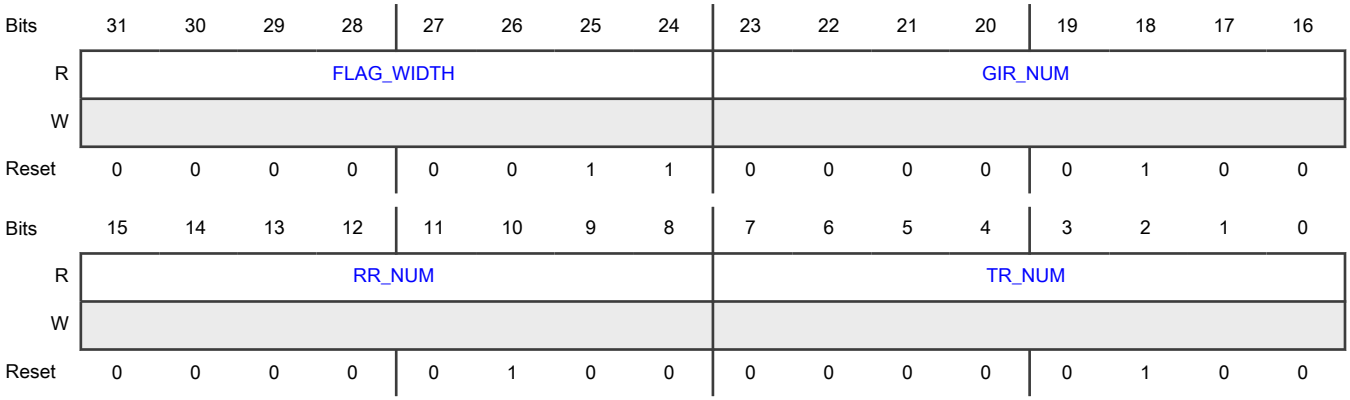
Offset

Register	Offset
PAR	4h

Function

Defines the number of flags, transmit registers, receive registers, and general-purpose interrupt requests available for MU.

Diagram



Fields

Field	Function
31-24 FLAG_WIDTH	Flag Width Specifies the number of flag bits (3) in the Flag Control (FCR) and Flag Status (FSR) registers.
23-16 GIR_NUM	General-Purpose Interrupt Request Number Specifies the number of general-purpose interrupt requests available (4).
15-8 RR_NUM	Receive Register Number Specifies the number of receive registers (4).
7-0 TR_NUM	Transmit Register Number Specifies the number of transmit registers (4).

26.7.1.4 Control (CR)

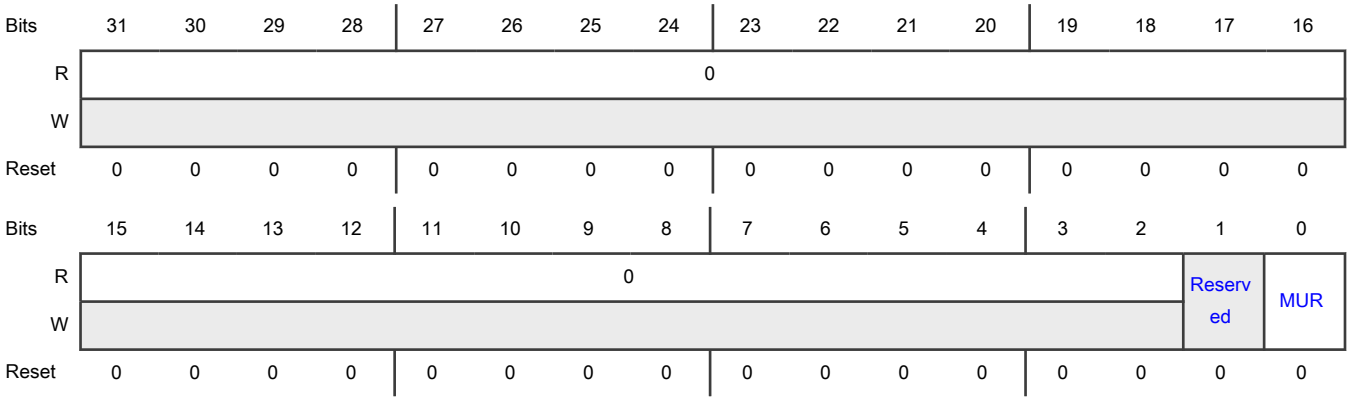
Offset

Register	Offset
CR	8h

Function

Controls MU reset and reset interrupt enable.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 —	Reserved
0 MUR	MU Reset Resets MU. Writing 1 to this field resets the MUA and MUB sides. All internal states are cleared. It forces all control and status registers to return to their default values (except HR, HRM in MUA/B_CCR0 registers; MURS in MUA/B_SR registers and the MUA/B_CSSR0/CSR0 registers). Before writing 1 to this field, interrupt processor B because writing 1 to this field may affect the ongoing processor B program. After writing 1 to this field, monitor the value of MUA_SR[MURS] to know when the reset sequence on the processor B-side has ended. You can only write 1 to MUR when processor B is not in Stop mode or Power-Down mode. This field always reads 0, and it becomes 0 during the MU reset sequence. 0b - Idle 1b - Reset

26.7.1.5 Status (SR)

Offset

Register	Offset
SR	Ch

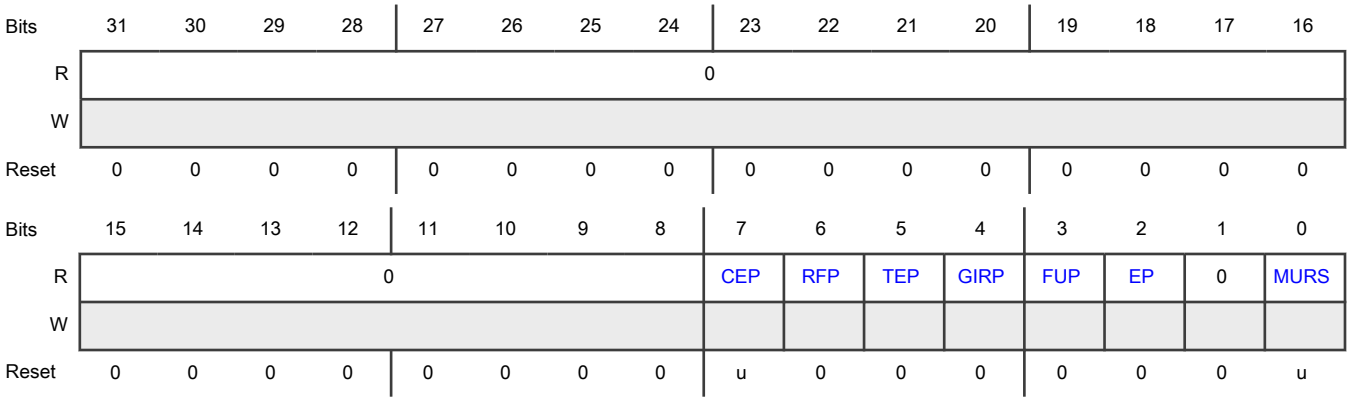
Function

Shows the status of MU resets and the status of pending events and requests.

NOTE

Writing to this register does not generate a transfer error.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 CEP	Processor B Event Pending Indicates whether a processor event is pending. This field becomes 1 when MU detects any core operation mode entry event. After this field becomes 1, MU checks the CSSR0 register to determine which core operation mode entry event in Processor B occurs. This field becomes 0 when all core operation mode entry events are serviced by writing 1 to CSSR0[WAIT] and CSSR0[HRIP]. The reset value of this field is chip-specific. 0b - No event pending 1b - Event pending
6 RFP	MUA Receive Full Pending Indicates whether a receive full message is pending. This field becomes 1 when MUB writes to a TR*n* register to send data to MUA. After this field becomes 1, MU checks RSR[RF*n*] to determine whether the data in the Receive register is ready for MUA to read it. This field becomes 0 when all MUA RR*n* registers are read, or when MU is reset. 0b - Not pending; MUB is not writing to a Transmit register 1b - Pending; MUB is writing to a Transmit register
5	MUA Transmit Empty Pending

Table continues on the next page...

Table continued from the previous page...

Field	Function
TEP	Indicates whether a transmit empty message is pending. This field becomes 1 when any TCR[TIE*n*] field is 1 and TSR[TE*n*] flag is set. After this field becomes 1, MU checks TSR[TE*n*] to determine whether the data in the Transmit (TR*n*) register is ready for MUA to write to it. This field becomes 0 when write operations to all MUA Transmit (TR*n*) registers where TCR[TIE*n*] = 1 (transfer interrupt enabled) are completed, or when MU is reset. 0b - No MUA transmit empty event pending 1b - Pending; any TCR[TIEn] field is 1 and TSR[TEn] flag is set
4 GIRP	MUA General-Purpose Interrupt Pending Indicates that MUB has sent a general-purpose interrupt request. This field becomes 1 when the MUB side sends a general-purpose interrupt request to the MUA side. GSR[GIP*n*] identifies which general-purpose interrupt request is received. This field becomes 0 when all MUA_GSR[GIP*n*] fields are cleared, or when MU is reset. 0b - No request sent 1b - Request sent
3 FUP	MUA Flag Update Pending Indicates whether a flag update request is pending. MU generates this request when there is a change to the F*n*[2:0] bits of MUA_FCR. This field becomes 1 when the MUA side sends a flag update request to the MUB side. This field becomes 0 when MU acknowledges this flag update request internally (the flag is updated) from the MUB side, or during MU reset. No flag update changes are allowed when this field is 1. When FUP = 1, a write to the F*n*[2:0] bits of MUA_FCR does not generate a flag update event. The F*n*[2:0] bits do not change. If SR[EP] = 1 (event pending), writing to MUA_FCR does not immediately cause this field to become 1. 0b - No pending update flags (initiated by MUA) 1b - Pending update flags (initiated by MUA)
2 EP	MUA Side Event Pending Indicates a pending side event when the MUA side sends an event update request to the MUB side. An event is any hardware message that the Status register on the MUB side reflects. For example, an event occurs when Transmit register 0 is the target of a write operation. During normal operations, the update mechanism for this field operates automatically. MU clears this field automatically when the event update acknowledgment is received, or when MU resets. To ensure that events are posted to MUB before entering Stop mode, verify that this field is 0. If it is 1, wait and continue to poll this field before entering Stop mode until it becomes 0. Reading MUA_SR[EP] must be the last access to MUA. Perform this read operation before entering Stop or Wait mode. Otherwise, subsequent actions may cause this field to become 1. If MUB is in Stop mode and the MUA event is triggered, this field remains 1 until MUB wakes up.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not pending 1b - Pending
1 —	Reserved
0 MURS	MUA and MUB Reset State Indicates the reset state of MUA and MUB. This field becomes 1 during any system reset or MU reset from the MUA or MUB side. This field becomes 0 when the reset sequence on both MUA and MUB sides ends. After issuing any of the aforementioned reset events, verify that this field is 0 before starting any access. 0b - Out of reset 1b - In reset

26.7.1.6 Core Control 0 (CCR0)

Offset

Register	Offset
CCR0	10h

Function

Allows MUA to control the processor on the MUB side.

Also allows control of the processor A hardware reset mask.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			Reserved	0	Reserved	0	NMI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Function
31-7 —	Reserved
6-5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 NMI	MUB Nonmaskable Interrupt Request Indicates whether Processor A has issued a nonmaskable interrupt to Processor B. When this field becomes 1, it initiates a nonmaskable interrupt to processor B. This field becomes 0 after 1 is written to MUB_CSSR0[NMIC] to clear that field. After this field becomes 0, MUA can initiate another nonmaskable interrupt to MUB. This field is cleared when MU resets. 0b - Nonmaskable interrupt not issued 1b - Nonmaskable interrupt issued

26.7.1.7 Core Interrupt Enable 0 (CIER0)

Offset

Register	Offset
CIER0	14h

Function
Controls interrupt enables.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								Reserved	Reserved	WAITIE	Reserved	Reserved	Reserved	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 WAITIE	Processor B Wait Mode Entry Interrupt Enable Enables Processor A Wait Mode Entry Interrupt request due to Processor B Wait Mode Entry. If the value of this field is 1, an MU Wait Mode Entry Interrupt is issued to the processor A when MUA_CSSR0[WAIT] = 1. If the value of this field is 0, MU ignores the value of MUA_CSSR0[WAIT] and no MU Wait Mode Entry Interrupt is issued. 0b - Disable 1b - Enable
4 —	Reserved
3 —	Reserved
2 —	Reserved
1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 —	Reserved

26.7.1.8 Core Sticky Status 0 (CSSR0)

Offset

Register	Offset
CSSR0	18h

Function

Shows the status of interrupts pending (W1C).

The reset value is chip-specific.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	WAIT	0	0	0	0	NMIC
W											W1C					W1C
Reset	0	0	0	0	0	0	0	0	0	0	u	0	0	0	0	u

Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
5 WAIT	Processor B Wait Mode Entry Interrupt Pending Indicates to Processor A whether Processor B has entered Wait mode. This flag is set after Processor B enters Wait mode. If CIER0[WAITIE] enables the interrupt, the Processor A Wait Mode Entry Interrupt request is issued when Processor B enters Wait mode. Clearing this flag also clears the MU Wait Mode Entry Interrupt request. 0b - Processor B did not enter Wait Mode. 1b - Processor B entered Wait Mode.
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 NMIC	Processor A Nonmaskable Interrupt Clear Clears the nonmaskable interrupt (NMI) request from the MUB side. The MUA-side NMI service routine uses this field. Writing 1 to this field clears MUB_CCR0[NMI], deasserting the NMI request and enabling MUB_CCR0[NMI] to trigger another NMI request. This field always reads as 0, so you cannot poll it. You can only use this field as part of the NMI service routine, in which you must write 1 to this field only once. This field is cleared when MU resets. 0b - Default 1b - Clear MUB_CCR0[NMI]

26.7.1.9 Core Status 0 (CSR0)

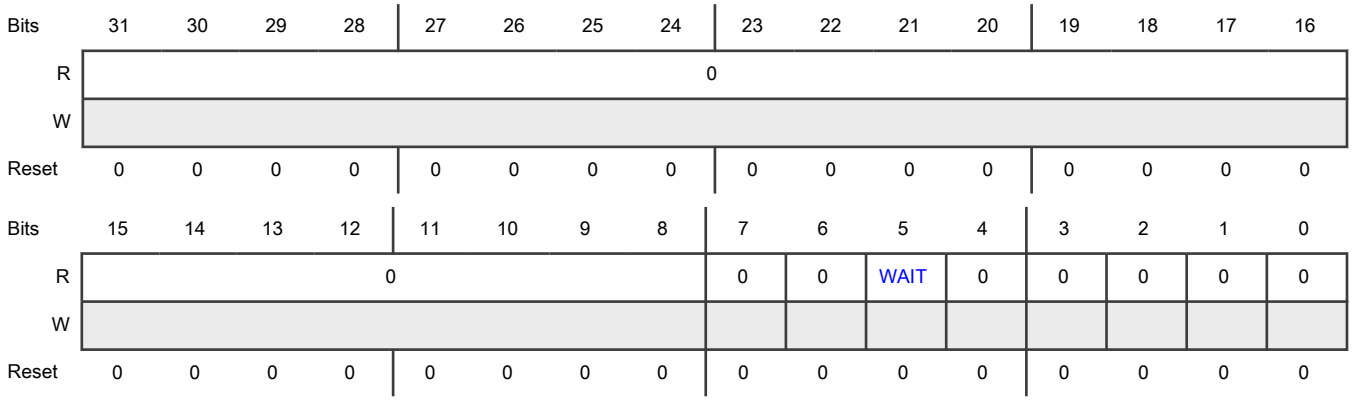
Offset

Register	Offset
CSR0	1Ch

Function

Shows the status of core events.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 WAIT	Processor B Wait Mode Entry Indicates whether Processor B is in Wait mode. This field becomes 1 after Processor B enters Wait mode. This field becomes 0 after Processor B exits Wait mode. 0b - Not in Wait mode 1b - In Wait mode
4 —	Reserved
3 —	Reserved
2 —	Reserved
1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 —	Reserved

26.7.1.10 Flag Control (FCR)

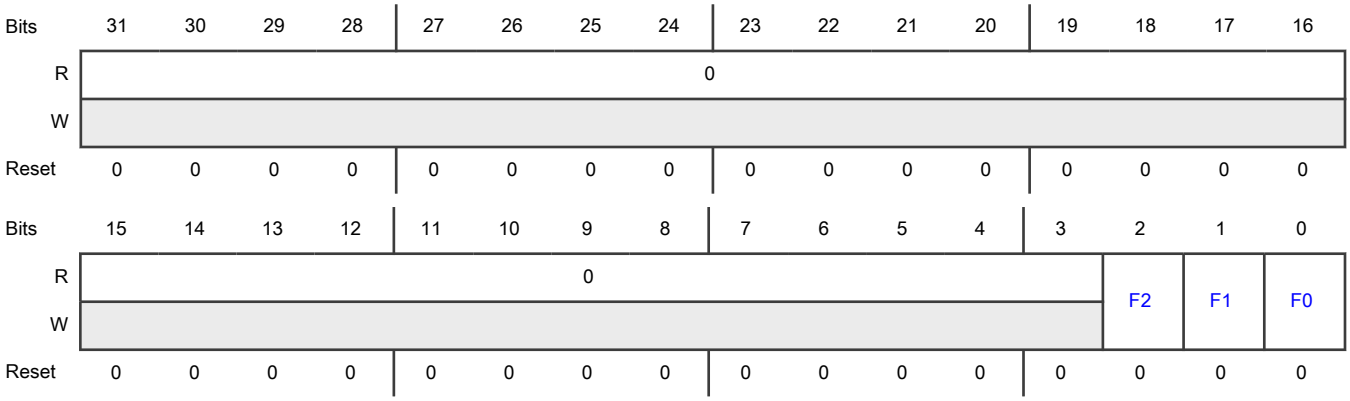
Offset

Register	Offset
FCR	100h

Function

Configures MUB_FSR[F*n*] flags.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 Fn	MUA to MUB Flag Configures MUB_FSR[F*n*] flags, where *n* = 0 to 2. F*n* configures the corresponding MUB_FSR[F*n*] flag. F*n* becomes 0 when MU resets. 0b - Clear MUB_FSR[Fn] 1b - Set MUB_FSR[Fn]

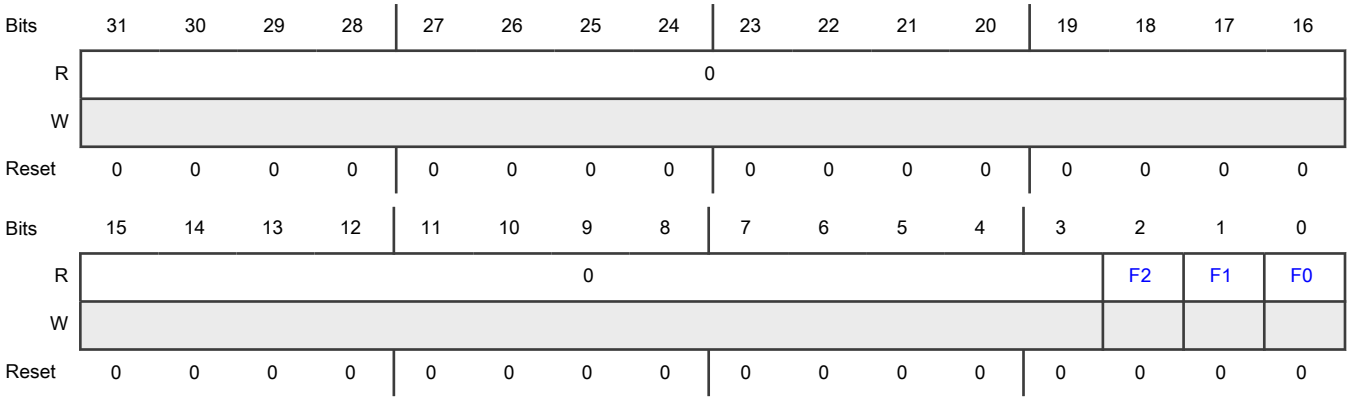
26.7.1.11 Flag Status (FSR)

Offset

Register	Offset
FSR	104h

Function
Contains flags configured by the values written to MUB_FCR[F*n*].

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 Fn	MUB to MUA-Side Flag Contains flags configured by the values written to MUB_FCR[F*n*], where *n* = 0 to 2. F*n* is the MUA-side flag configured by the values written to MUB_FCR[F*n*]. When MUB_FCR[F*n*] is written to, the write event updates MUA_FSR[F*n*], after the event update latency. 0b - MUB_FCR[Fn] = 0 1b - MUB_FCR[Fn] = 1

26.7.1.12 General-Purpose Interrupt Enable (GIER)

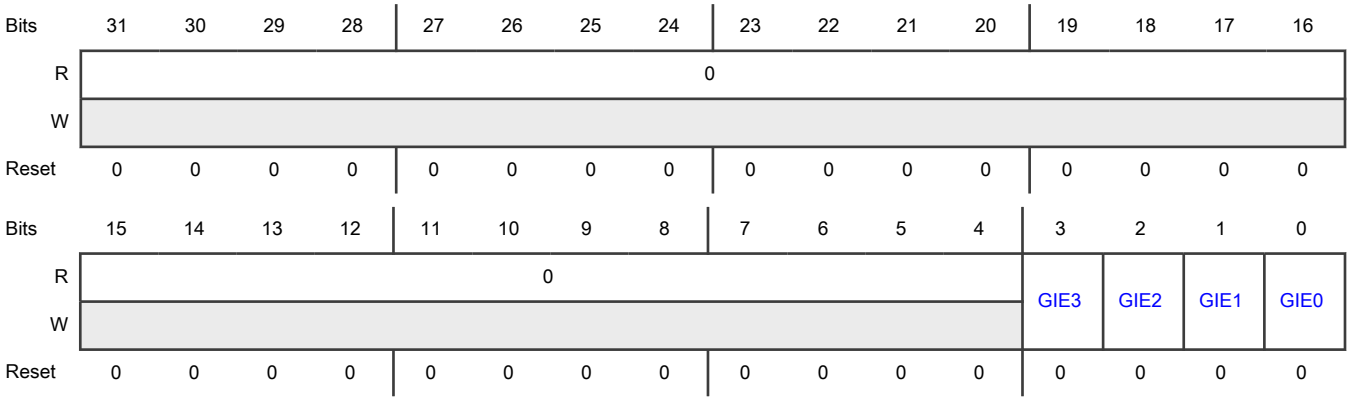
Offset

Register	Offset
GIER	110h

Function

Contains the MUA general-purpose interrupt enable fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIEn	MUA General-purpose Interrupt Enable Enables general-purpose interrupt. There are 4 general-purpose interrupts (*n* = 0 to 3). When GIE*n* = 1, a general-purpose interrupt *n* request is issued to processor A when MUA GSR[GIP*n*] = 1. If GIE*n* = 0, the general-purpose interrupt request pending does not trigger the general-purpose interrupt. GIE*n* becomes 0 when MU resets. 0b - Disable 1b - Enable

26.7.1.13 General-Purpose Control (GCR)

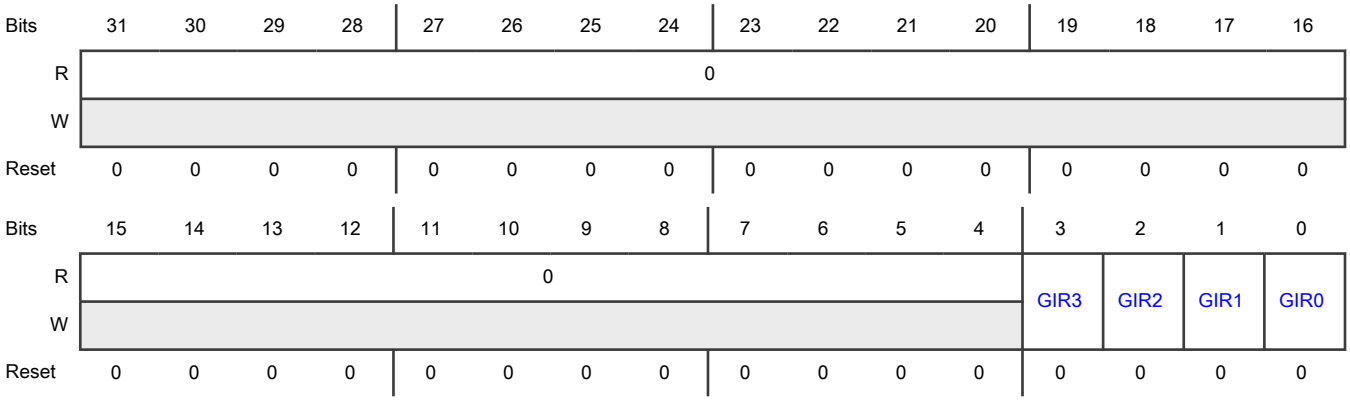
Offset

Register	Offset
GCR	114h

Function

Contains the MUA general-purpose interrupt request fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIRn	MUA General-Purpose Interrupt Request Specifies whether general-purpose interrupts are requested to MUB. There are 4 general-purpose interrupts (*n* = 0 to 3). Writing 1 to GIR*n* sets MUB_GSR[GIP*n*]. If MUB_GIER[GIE*n*] = 1, a general-purpose interrupt request is triggered on processor B. This field becomes 0 when MUB_GSR[GIP*n*] is cleared. This clearing informs MUA that the interrupt is accepted (cleared by software). To ensure proper operations, verify that GIR*n* is 0 (no pending interrupt) before writing 1 to it. This field becomes 0 when MU resets. 0b - Not requested 1b - Requested

26.7.1.14 General-purpose Status (GSR)

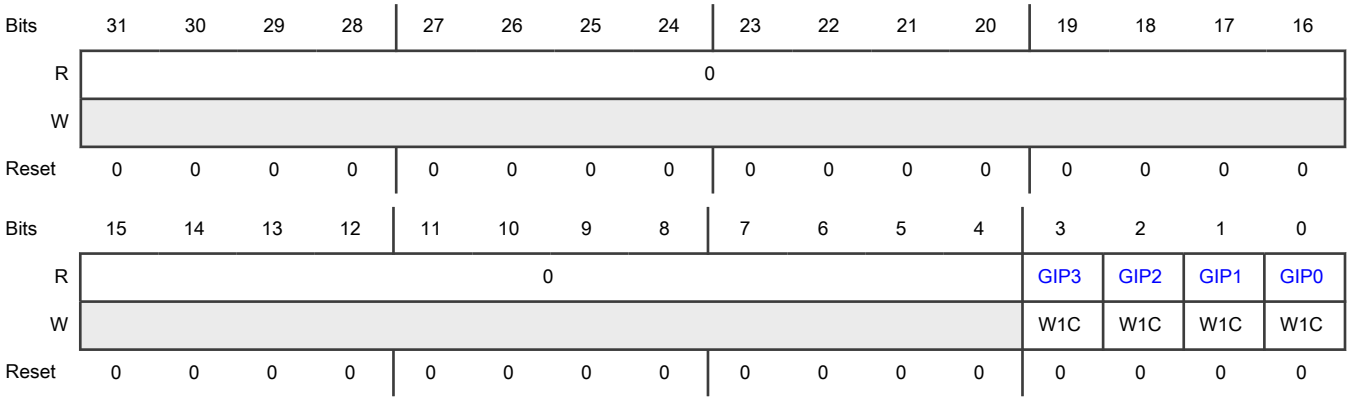
Offset

Register	Offset
GSR	118h

Function

Contains the status of the MUA general-purpose interrupt pending requests.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIPn	MUA General-Purpose Interrupt Request Pending Indicates whether a general-purpose interrupt request is pending. There are 4 general-purpose interrupts (*n* = 0 to 3). GIP*n* informs MUA that MUB_GCR[GIR*n*] changed from 0 to 1. If MUA_GIER[GIE*n*] = 1, a general-purpose interrupt request is issued to processor A. GIP*n* is cleared when MU resets. After GIP*n* is cleared, if MUA_GIER[GIE*n*] = 1, the general-purpose interrupt request is cleared on the MUA side. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not pending 1b - Pending When writing 0b - No effect 1b - Clear the flag

26.7.1.15 Transmit Control (TCR)

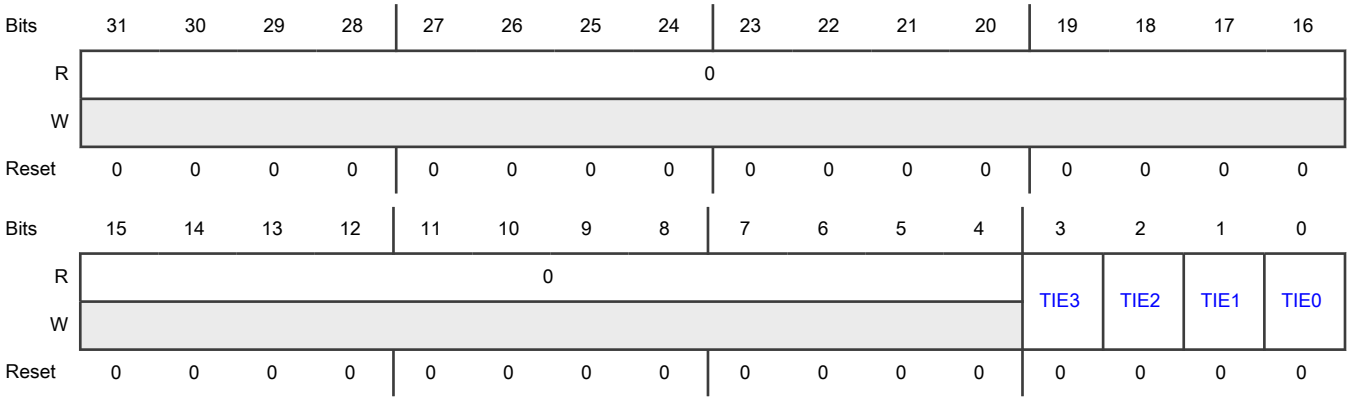
Offset

Register	Offset
TCR	120h

Function

Contains the MUA transmit interrupt enable fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TIEn	MUA Transmit Interrupt Enable Enables MUA transmit interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUA transmit interrupt *n* request is issued when MUA_TSR[TE*n*] is set. If this field is 0, MU ignores the value of MUA_TSR[TE*n*], and no MUA transmit interrupt *n* request is issued. 0b - Disable 1b - Enable

26.7.1.16 Transmit Status (TSR)

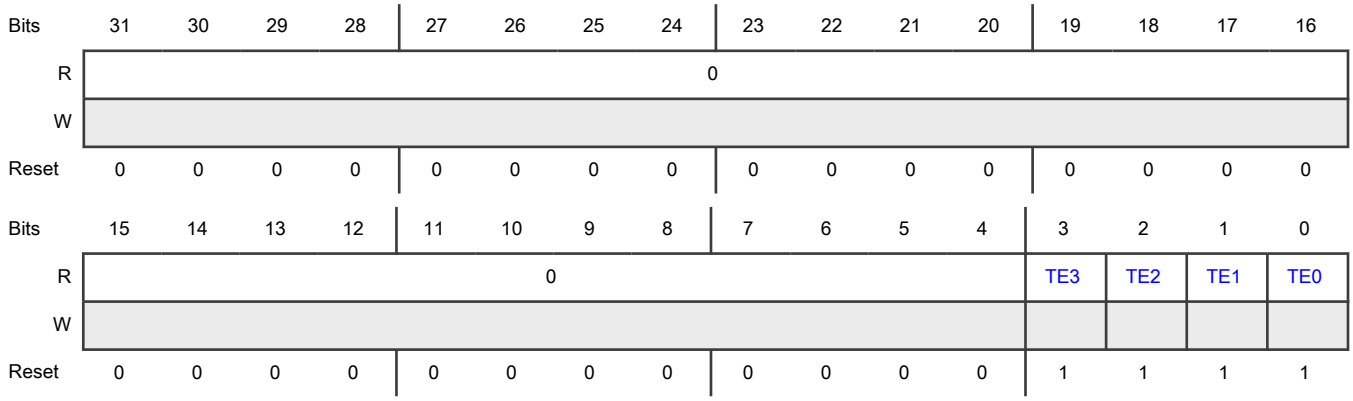
Offset

Register	Offset
TSR	124h

Function

Indicates whether the MUA transmit registers are empty.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TEn	MUA Transmit Empty Indicates whether MUA Transmit (TR*n*) register is empty, where *n* = 0 to 3. This field becomes 1 after the MUB_RR*n* register is read on the MUB side. When TE*n* = 1, it indicates to the MUA side that the MUA_TR*n* register is ready to be written on the MUA side. If MUA_TCR[TIE*n*] = 1, a transmit *n* interrupt is issued on the MUA side. This field becomes 0 after the MUA_TR*n* register is written to on the MUA side. After this field becomes 0, if MUA_TCR[TIE*n*] = 1, the transmit *n* interrupt request is cleared on the MUA side. This field becomes 1 when MU resets. 0b - Not empty 1b - Empty

26.7.1.17 Receive Control (RCR)

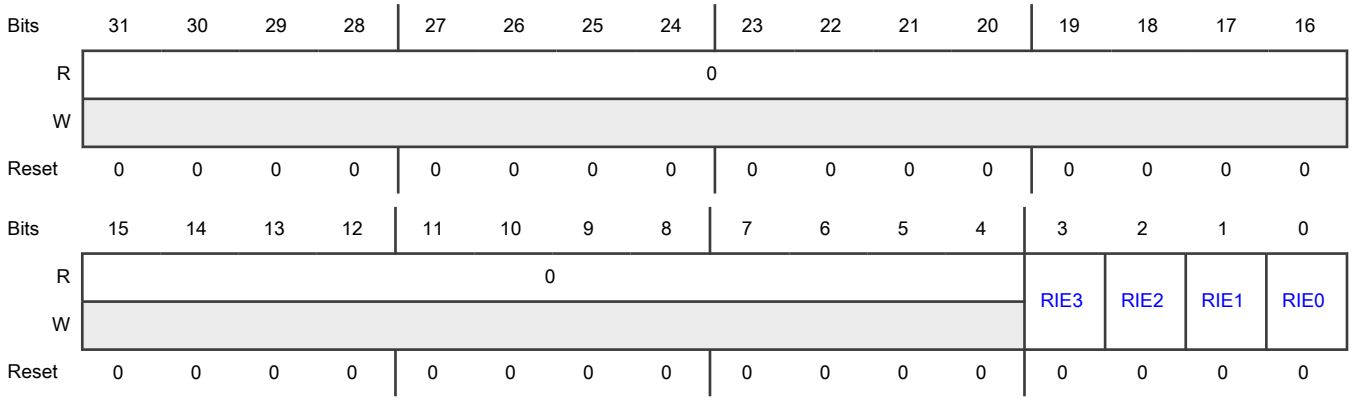
Offset

Register	Offset
RCR	128h

Function

Contains the MUA receive interrupt enables.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 RIEn	MUA Receive Interrupt Enable Enables MUA receive interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUA receive interrupt *n* request is issued when MUA_RSR[RF*n*] is set. If this field is 0, MU ignores the value of MUA_RSR[RF*n*], and no MUA receive interrupt request is issued. 0b - Disable 1b - Enable

26.7.1.18 Receive Status (RSR)

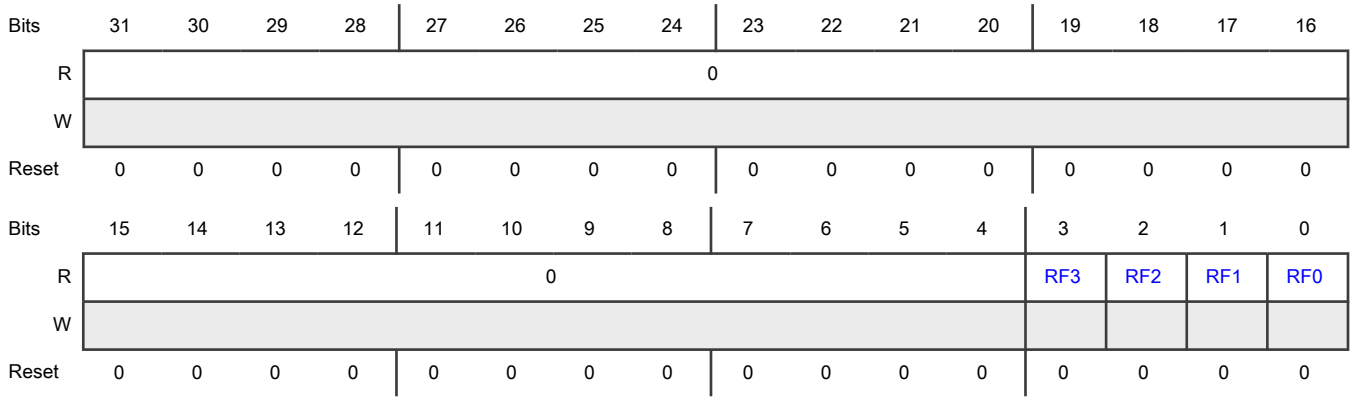
Offset

Register	Offset
RSR	12Ch

Function

Indicates whether the MUA receive registers are full.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 RFn	MUA Receive Register Full Indicates whether MUA Receive register (RR*n*) is full, where *n* = 0 to 3. RF*n* becomes 1 when the MUB_TR*n* register is written to on the MUB side. When RF*n* is 1, it indicates to the MUA side that new data in the MUA_RR*n* register is ready for MUA to read it. If MUA_RCR[RIE*n*] = 1, a receive *n* interrupt is issued on the MUA side. RF*n* becomes 0 when the MUA_RR*n* register is read, or when MU is reset. After RF*n* becomes 0, if MUA_RCR[RIE*n*] = 1, the receive *n* interrupt request is cleared on the MUA side. 0b - Not full 1b - Full

26.7.1.19 Transmit (TR0 - TR3)

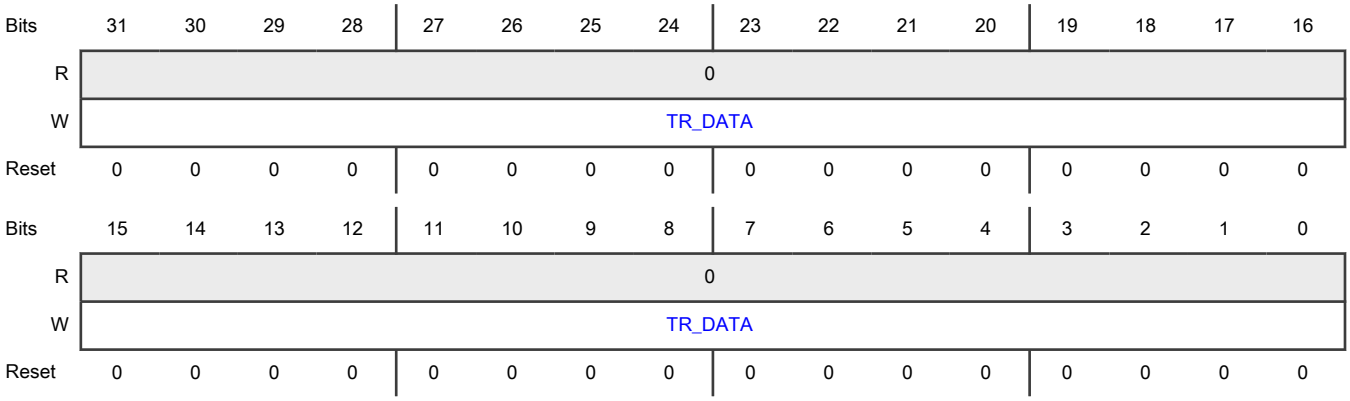
Offset

Register	Offset
TR0	200h
TR1	204h
TR2	208h
TR3	20Ch

Function

Contains MUA transmit data.

Diagram



Fields

Field	Function
31-0 TR_DATA	MUA Transmit Data Contains MUA transmit data. MUB_RR*n* reflects the data written to this register. The TR*n* and RR*n* registers are not double-buffered. Writing to MUA_TR*n* overrides the data readable in the MUA_RR*n* register. A write to the Transmit register clears MUA_TSR[TE*n*] on the transmitter side, and sets MUB_RSR[RF*n*] on the receiver side. You can write to this register only when MUA_TSR[TE*n*] = 1. Reading this register returns all zeroes.

26.7.1.20 Receive (RR0 - RR3)

Offset

Register	Offset
RR0	280h
RR1	284h
RR2	288h
RR3	28Ch

Function

Contains MUA receive data.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 RR_DATA	MUA Receive Data Reflects the data written to MUB TR*n*. Reading this register clears MUA_RSR[RF*n*] on the receiver side, and sets MUB_TSR[TE*n*] on the transmitter side. You can read this register only when MUA_RSR[RF*n*] = 1. Reading it before MUA_RSR[RF*n*] becomes 1 may result in reading incorrect data. Poll MUA_RSR[RF*n*] to confirm it is set before reading RR*n*. Writing to this register generates an error response to MUA.

26.7.2 MUB register descriptions

This section contains the detailed register descriptions for MUB registers.

NOTE

A module transfer error to processor A or processor B is generated when:

- A read or write access is made to an invalid location.
- A write operation is performed on a read-only register on the processor A side or processor B side of MU.

The only exception to this behavior is the MUB SR register. No transfer error occurs when accessing this location.

26.7.2.1 MUB memory map

MU0.MUB base address: 4020_1000h

MU1.MUB base address: 4020_3000h

MU2.MUB base address: 4020_5000h

MU3.MUB base address: 4031_A000h

MU4.MUB base address: 4018_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VER)	32	R	0309_000Fh
4h	Parameter (PAR)	32	R	0304_0404h
8h	Control (CR)	32	RW	0000_0000h
Ch	Status (SR)	32	R	See section
10h	Core Control 0 (CCR0)	32	RW	0000_0014h
14h	Core Interrupt Enable 0 (CIER0)	32	RW	0000_0000h
18h	Core Sticky Status 0 (CSSR0)	32	RW	See section
1Ch	Core Status 0 (CSR0)	32	R	0000_0000h
100h	Flag Control (FCR)	32	RW	0000_0000h
104h	Flag Status (FSR)	32	R	0000_0000h
110h	General-Purpose Interrupt Enable (GIER)	32	RW	0000_0000h
114h	General-Purpose Control (GCR)	32	RW	0000_0000h
118h	General-purpose Status (GSR)	32	RW	0000_0000h
120h	Transmit Control (TCR)	32	RW	0000_0000h
124h	Transmit Status (TSR)	32	R	0000_000Fh
128h	Receive Control (RCR)	32	RW	0000_0000h
12Ch	Receive Status (RSR)	32	R	0000_0000h
200h - 20Ch	Transmit (TR0 - TR3)	32	RW	0000_0000h
280h - 28Ch	Receive (RR0 - RR3)	32	R	0000_0000h

26.7.2.2 Version ID (VER)

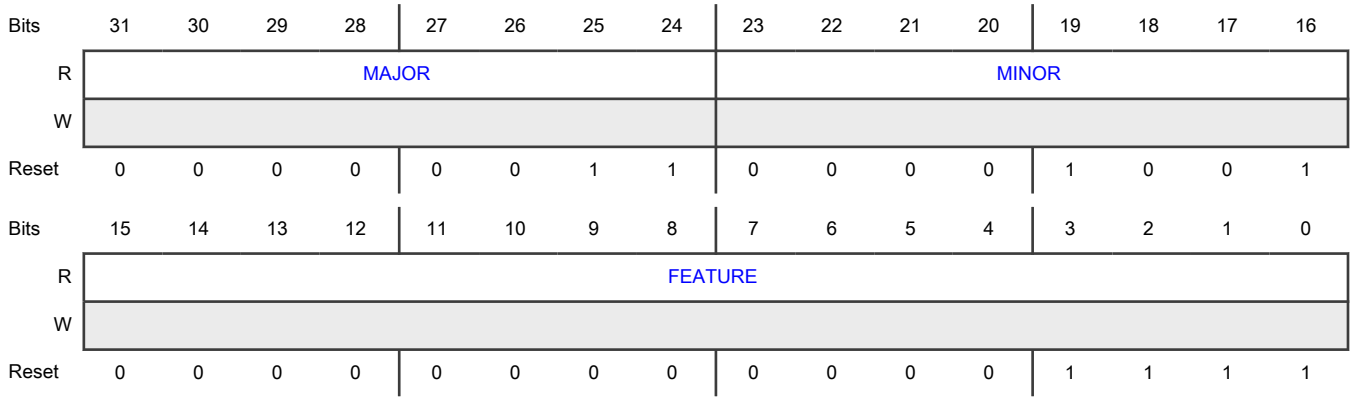
Offset

Register	Offset
VER	0h

Function

Determines the version ID and feature set number of MUB.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number
23-16 MINOR	Minor Version Number
15-0 FEATURE	Feature Set Number Indicates the feature set number. Each bit of this field specifies a different feature: • If FEATURE[0] = 1: Standard features are implemented. • If FEATURE[1] = 1: RAIP and RAIE fields are implemented. • If FEATURE[2] = 1: Core Control and Status registers are implemented in MUA and MUB. • If FEATURE[3] = 1: Expand the number of TR*n* and RR*n* registers. • FEATURE[4] - FEATURE[15]: Reserved for future use.

26.7.2.3 Parameter (PAR)

Offset

Register	Offset
PAR	4h

Function

Defines the number of flags, transmit registers, receive registers, and general-purpose interrupt requests available for MU.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FLAG_WIDTH								GIR_NUM							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RR_NUM								TR_NUM							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Fields

Field	Function
31-24 FLAG_WIDTH	Flag Width Specifies the number of flag bits (3) in the Flag Control (FCR) and Flag Status (FSR) registers.
23-16 GIR_NUM	General-Purpose Interrupt Request Number Specifies the number of general-purpose interrupt requests available (4).
15-8 RR_NUM	Receive Register Number Specifies the number of receive registers (4).
7-0 TR_NUM	Transmit Register Number Specifies the number of transmit registers (4).

26.7.2.4 Control (CR)

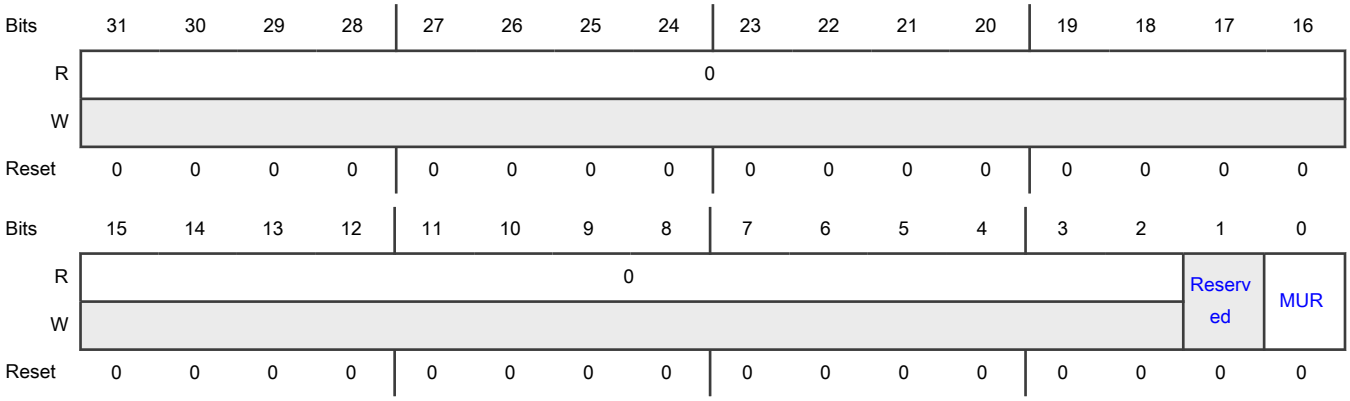
Offset

Register	Offset
CR	8h

Function

Controls MU reset and reset interrupt enable.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 —	Reserved
0 MUR	MU Reset Resets MU. Writing 1 to this field resets the MUB and MUA sides. All internal states are cleared. It forces all control and status registers to return to their default values (except HR, HRM in MUB/A_CCR0 registers; MURS in MUB/A_SR registers and the MUB/A_CSSR0/CSR0 registers). Before writing 1 to this field, interrupt processor A because writing 1 to this field may affect the ongoing processor A program. After writing 1 to this field, monitor the value of MUB_SR[MURS] to know when the reset sequence on the processor A-side has ended. You can only write 1 to MUR when processor A is not in Stop mode or Power-Down mode. This field always reads 0, and it becomes 0 during the MU reset sequence. 0b - Idle 1b - Reset

26.7.2.5 Status (SR)

Offset

Register	Offset
SR	Ch

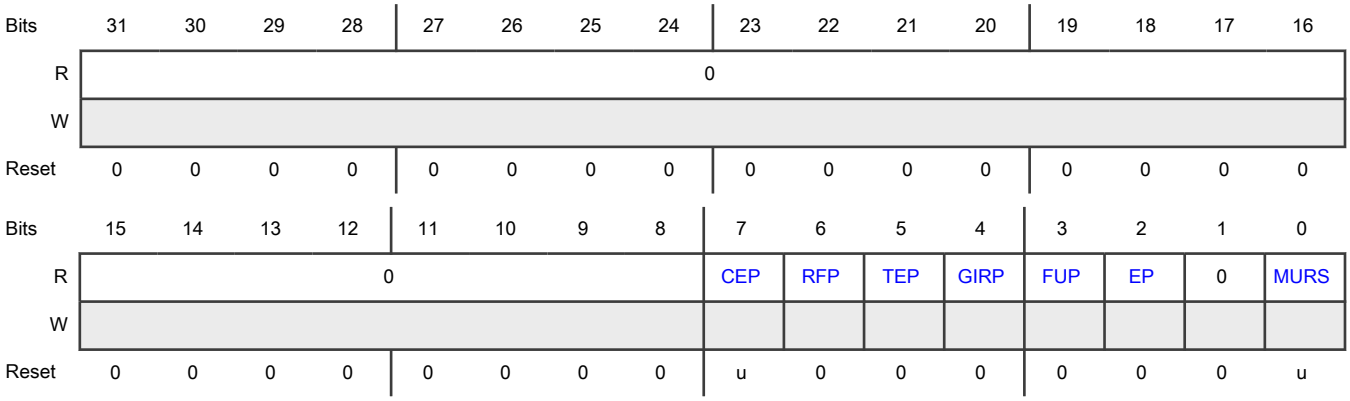
Function

Shows the status of MU resets and the status of pending events and requests.

NOTE

Writing to this register does not generate a transfer error.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 CEP	Processor A Event Pending Indicates whether a processor event is pending. This field becomes 1 when MU detects any core operation mode entry event. After this field becomes 1, MU checks the CSSR0 register to determine which core operation mode entry event in Processor A occurs. This field becomes 0 when all core operation mode entry events are serviced by writing 1 to CSSR0[WAIT] and CSSR0[HRIP]. The reset value of this field is chip-specific. 0b - No event pending 1b - Event pending
6 RFP	MUB Receive Full Pending Indicates whether a receive full message is pending. This field becomes 1 when MUA writes to a TR*n* register to send data to MUB. After this field becomes 1, MU checks RSR[RF*n*] to determine whether the data in the Receive register is ready for MUB to read it. This field becomes 0 when all MUB RR*n* registers are read, or when MU is reset. 0b - Not pending; MUA is not writing to a Transmit register 1b - Pending; MUA is writing to a Transmit register
5 TEP	MUB Transmit Empty Pending Indicates whether a transmit empty message is pending.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field becomes 1 when any TCR[TIE*n*] field is 1 and TSR[TE*n*] flag is set. After this field becomes 1, MU checks TSR[TE*n*] to determine whether the data in the Transmit (TR*n*) register is ready for MUB to write to it. This field becomes 0 when write operations to all MUB Transmit (TR*n*) registers where TCR[TIE*n*] = 1 (transfer interrupt enabled) are completed, or when MU is reset. 0b - No MUB transmit empty event pending 1b - Pending; any TCR[TIEn] field is 1 and TSR[TEn] flag is set
4 GIRP	MUB General-Purpose Interrupt Pending Indicates that MUA has sent a general-purpose interrupt request. This field becomes 1 when the MUA side sends a general-purpose interrupt request to the MUB side. GSR[GIP*n*] identifies which general-purpose interrupt request is received. This field becomes 0 when all MUB_GSR[GIP*n*] fields are cleared, or when MU is reset. 0b - No request sent 1b - Request sent
3 FUP	MUB Flag Update Pending Indicates whether a flag update request is pending. MU generates this request when there is a change to the F*n*[2:0] bits of MUB_FCR. This field becomes 1 when the MUB side sends a flag update request to the MUA side. This field becomes 0 when MU acknowledges this flag update request internally (the flag is updated) from the MUA side, or during MU reset. No flag update changes are allowed when this field is 1. When FUP = 1, a write to the F*n*[2:0] bits of MUB_FCR does not generate a flag update event. The F*n*[2:0] bits do not change. If SR[EP] = 1 (event pending), writing to MUB_FCR does not immediately cause this field to become 1. 0b - No pending update flags (initiated by MUB) 1b - Pending update flags (initiated by MUB)
2 EP	MUB Side Event Pending Indicates a pending side event when the MUB side sends an event update request to the MUA side. An event is any hardware message that the Status register on the MUA side reflects. For example, an event occurs when Transmit register 0 is the target of a write operation. During normal operations, the update mechanism for this field operates automatically. MU clears this field automatically when the event update acknowledgment is received, or when MU resets. To ensure that events are posted to MUA before entering Stop mode, verify that this field is 0. If it is 1, wait and continue to poll this field before entering Stop mode until it becomes 0. Reading MUB_SR[EP] must be the last access to MUB. Perform this read operation before entering Stop or Wait mode. Otherwise, subsequent actions may cause this field to become 1. If MUA is in Stop mode and the MUB event is triggered, this field remains 1 until MUA wakes up.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not pending 1b - Pending
1 —	Reserved
0 MURS	MUA and MUB Reset State Indicates the reset state of MUA and MUB. This field becomes 1 during any system reset or MU reset from the MUA or MUB side. This field becomes 0 when the reset sequence on both MUA and MUB sides ends. After issuing any of the aforementioned reset events, verify that this field is 0 before starting any access. 0b - Out of reset 1b - In reset

26.7.2.6 Core Control 0 (CCR0)

Offset

Register	Offset
CCR0	10h

Function

Allows MUB to control the processor on the MUA side.

Also allows control of the processor B hardware reset mask.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			Reserved	0	Reserved	0	NMI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Function
31-7 —	Reserved
6-5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 NMI	MUA Nonmaskable Interrupt Request Indicates whether Processor B has issued a nonmaskable interrupt to Processor A. When this field becomes 1, it initiates a nonmaskable interrupt to processor A. This field becomes 0 after 1 is written to MUA_CSSR0[NMIC] to clear that field. After this field becomes 0, MUB can initiate another nonmaskable interrupt to MUA. This field is cleared when MU resets. 0b - Nonmaskable interrupt not issued 1b - Nonmaskable interrupt issued

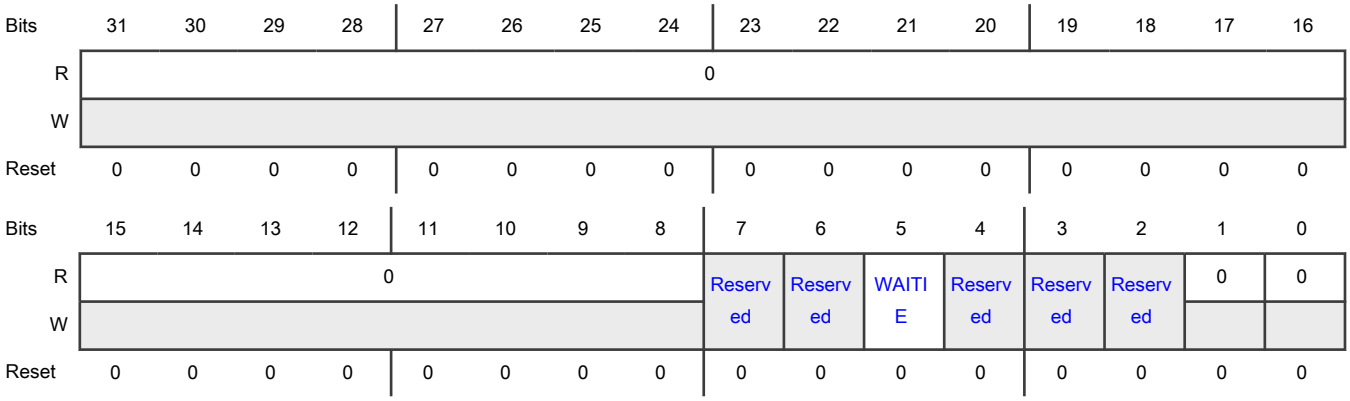
26.7.2.7 Core Interrupt Enable 0 (CIER0)

Offset

Register	Offset
CIER0	14h

Function
Controls interrupt enables.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 WAITIE	Processor A Wait Mode Entry Interrupt Enable Enables Processor B Wait Mode Entry Interrupt request due to Processor A Wait Mode Entry. If the value of this field is 1, an MU Wait Mode Entry Interrupt is issued to the processor B when MUB_CSSR0[WAIT] = 1. If the value of this field is 0, MU ignores the value of MUB_CSSR0[WAIT] and no MU Wait Mode Entry Interrupt is issued. 0b - Disable 1b - Enable
4 —	Reserved
3 —	Reserved
2 —	Reserved
1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 —	Reserved

26.7.2.8 Core Sticky Status 0 (CSSR0)

Offset

Register	Offset
CSSR0	18h

Function

Shows the status of interrupts pending (W1C).

The reset value is chip-specific.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	WAIT	0	0	0	0	NMIC
W											W1C					W1C
Reset	0	0	0	0	0	0	0	0	0	0	u	0	0	0	0	u

Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
5 WAIT	Processor A Wait Mode Entry Interrupt Pending Indicates to Processor B whether Processor A has entered Wait mode. This flag is set after Processor A enters Wait mode. If CIER0[WAITIE] enables the interrupt, the Processor B Wait Mode Entry Interrupt request is issued when Processor A enters Wait mode. Clearing this flag also clears the MU Wait Mode Entry Interrupt request. 0b - Processor A did not enter Wait Mode. 1b - Processor A entered Wait Mode.
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 NMIC	Processor B Nonmaskable Interrupt Clear Clears the nonmaskable interrupt (NMI) request from the MUA side. The MUB-side NMI service routine uses this field. Writing 1 to this field clears MUA_CCR0[NMI], deasserting the NMI request and enabling MUA_CCR0[NMI] to trigger another NMI request. This field always reads as 0, so you cannot poll it. You can only use this field as part of the NMI service routine, in which you must write 1 to this field only once. This field is cleared when MU resets. 0b - Default 1b - Clear MUA_CCR0[NMI]

26.7.2.9 Core Status 0 (CSR0)

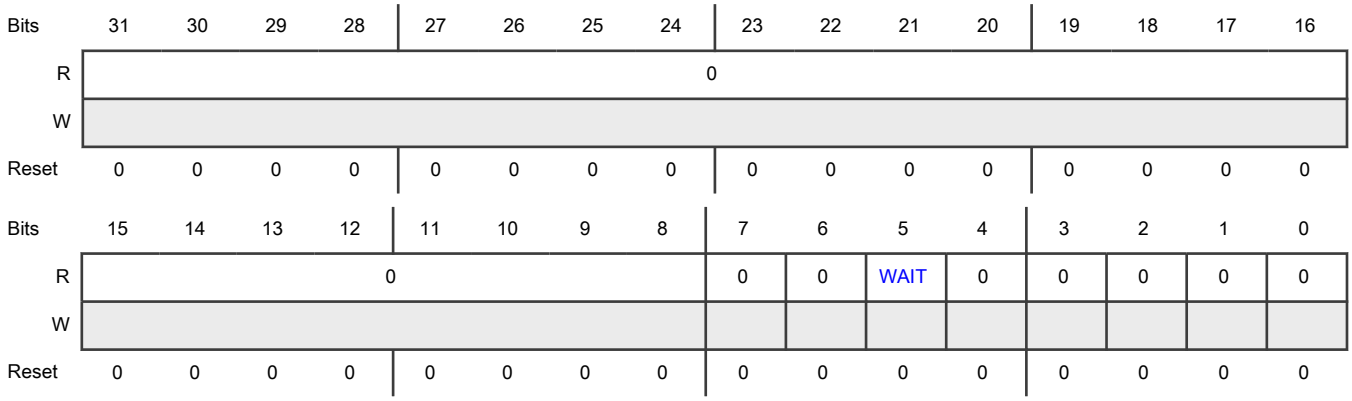
Offset

Register	Offset
CSR0	1Ch

Function

Shows the status of core events.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 WAIT	Processor A Wait Mode Entry Indicates whether Processor A is in Wait mode. This field becomes 1 after Processor A enters Wait mode. This field becomes 0 after Processor A exits Wait mode. 0b - Not in Wait mode 1b - In Wait mode
4 —	Reserved
3 —	Reserved
2 —	Reserved
1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 —	Reserved

26.7.2.10 Flag Control (FCR)

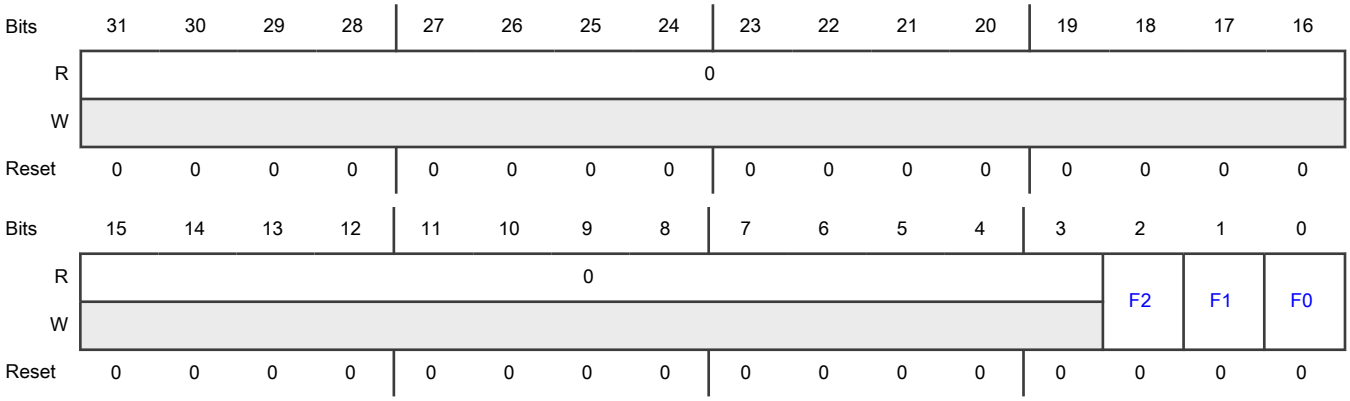
Offset

Register	Offset
FCR	100h

Function

Configures MUA_FSR[F*n*] flags.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 Fn	MUB to MUA Flag Configures MUA_FSR[F*n*] flags, where *n* = 0 to 2. F*n* configures the corresponding MUA_FSR[F*n*] flag. F*n* becomes 0 when MU resets. 0b - Clear MUA_FSR[Fn] 1b - Set MUA_FSR[Fn]

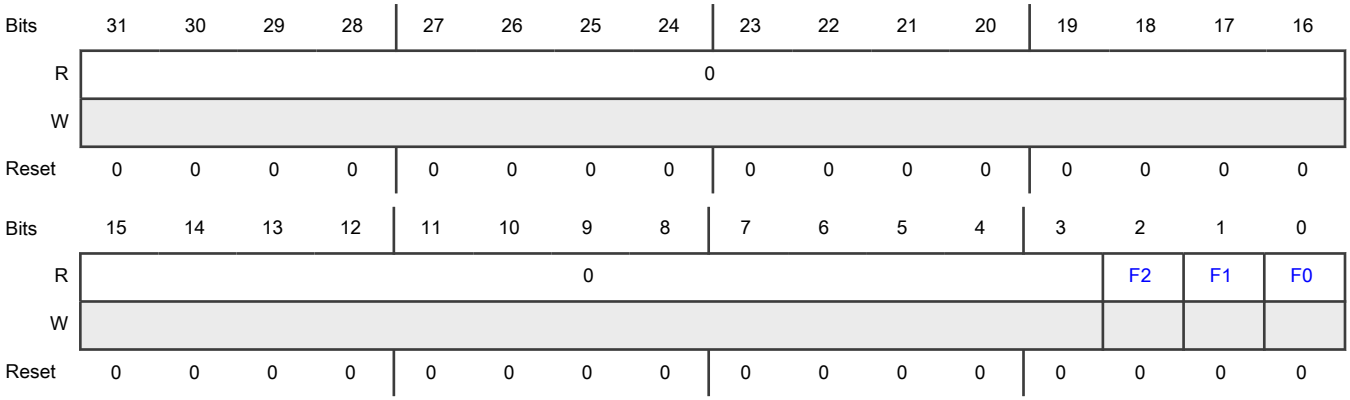
26.7.2.11 Flag Status (FSR)

Offset

Register	Offset
FSR	104h

Function
Contains flags configured by the values written to MUA_FCR[F*n*].

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 Fn	MUA to MUB-Side Flag Contains flags configured by the values written to MUA_FCR[F*n*], where *n* = 0 to 2. F*n* is the MUB-side flag configured by the values written to MUA_FCR[F*n*]. When MUA_FCR[F*n*] is written to, the write event updates MUB_FSR[F*n*], after the event update latency. 0b - MUA_FCR[Fn] = 0 1b - MUA_FCR[Fn] = 1

26.7.2.12 General-Purpose Interrupt Enable (GIER)

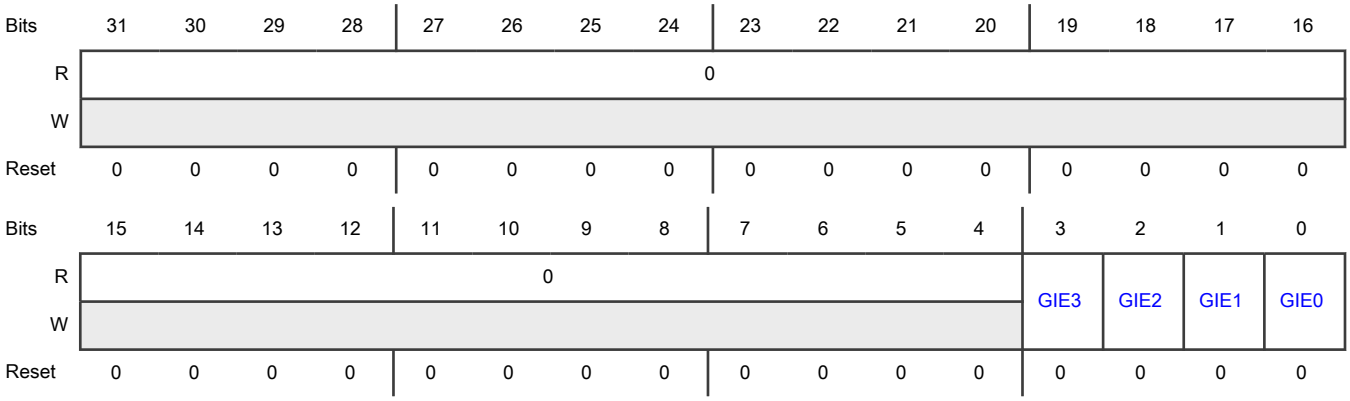
Offset

Register	Offset
GIER	110h

Function

Contains the MUB general-purpose interrupt enable fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIEn	MUB General-purpose Interrupt Enable Enables general-purpose interrupt. There are 4 general-purpose interrupts (*n* = 0 to 3). When GIE*n* = 1, a general-purpose interrupt *n* request is issued to processor B when MUB GSR[GIP*n*] = 1. If GIE*n* = 0, the general-purpose interrupt request pending does not trigger the general-purpose interrupt. GIE*n* becomes 0 when MU resets. 0b - Disable 1b - Enable

26.7.2.13 General-Purpose Control (GCR)

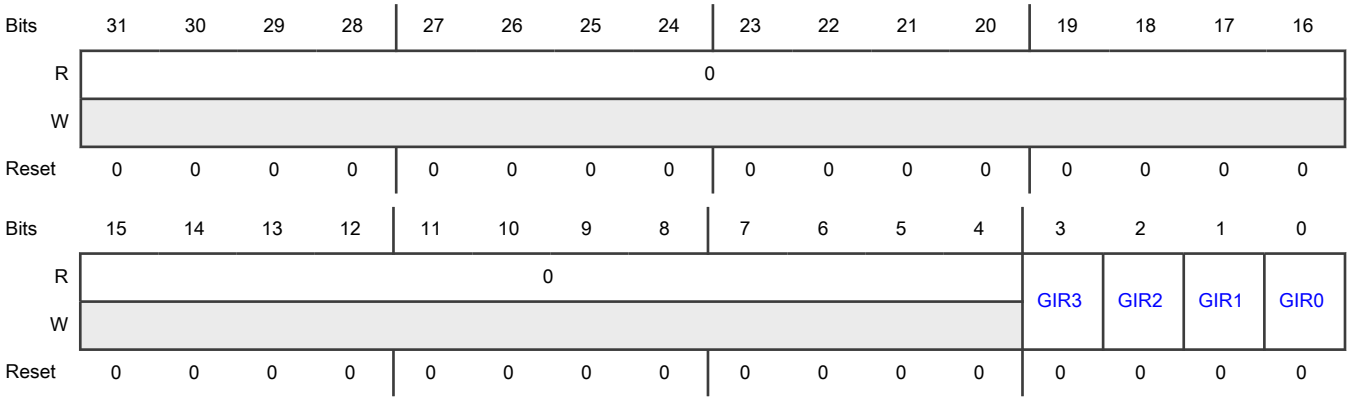
Offset

Register	Offset
GCR	114h

Function

Contains the MUB general-purpose interrupt request fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIRn	MUB General-Purpose Interrupt Request Specifies whether general-purpose interrupts are requested to MUA. There are 4 general-purpose interrupts (*n* = 0 to 3). Writing 1 to GIR*n* sets MUA_GSR[GIP*n*]. If MUA_GIER[GIE*n*] = 1, a general-purpose interrupt request is triggered on processor A. This field becomes 0 when MUA_GSR[GIP*n*] is cleared. This clearing informs MUB that the interrupt is accepted (cleared by software). To ensure proper operations, verify that GIR*n* is 0 (no pending interrupt) before writing 1 to it. This field becomes 0 when MU resets. 0b - Not requested 1b - Requested

26.7.2.14 General-purpose Status (GSR)

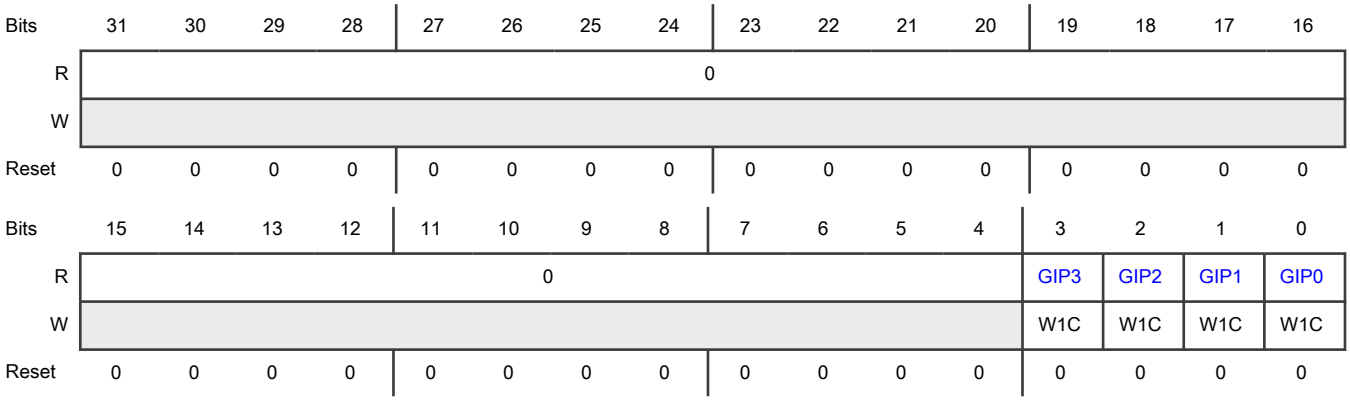
Offset

Register	Offset
GSR	118h

Function

Contains the status of the MUB general-purpose interrupt pending requests.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 GIPn	MUB General-Purpose Interrupt Request Pending Indicates whether a general-purpose interrupt request is pending. There are 4 general-purpose interrupts (*n* = 0 to 3). GIP*n* informs MUB that MUA_GCR[GIR*n*] changed from 0 to 1. If MUB_GIER[GIE*n*] = 1, a general-purpose interrupt request is issued to processor B. GIP*n* is cleared when MU resets. After GIP*n* is cleared, if MUB_GIER[GIE*n*] = 1, the general-purpose interrupt request is cleared on the MUB side. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not pending 1b - Pending When writing 0b - No effect 1b - Clear the flag

26.7.2.15 Transmit Control (TCR)

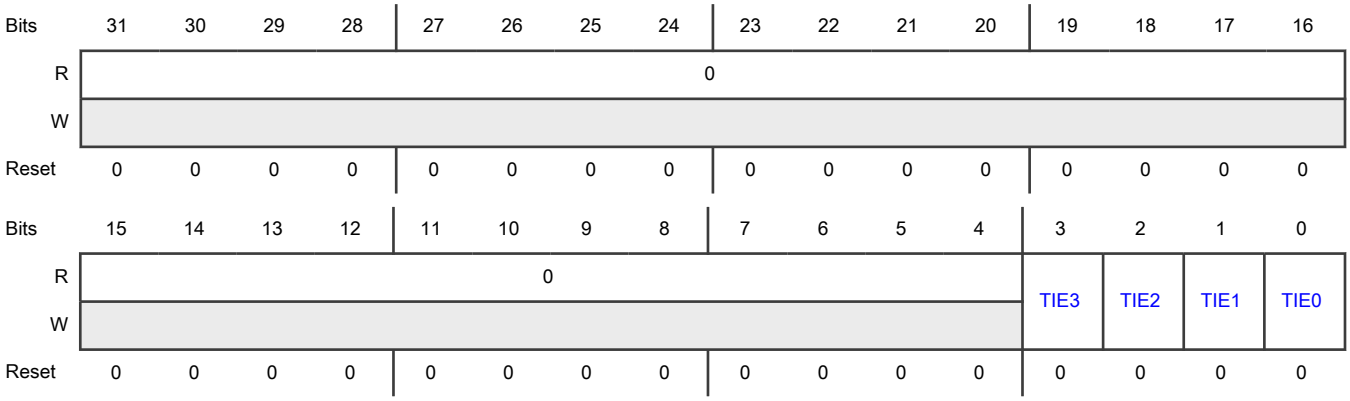
Offset

Register	Offset
TCR	120h

Function

Contains the MUB transmit interrupt enable fields.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TIEn	MUB Transmit Interrupt Enable Enables MUB transmit interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUB transmit interrupt *n* request is issued when MUB_TSR[TE*n*] is set. If this field is 0, MU ignores the value of MUB_TSR[TE*n*], and no MUB transmit interrupt *n* request is issued. 0b - Disable 1b - Enable

26.7.2.16 Transmit Status (TSR)

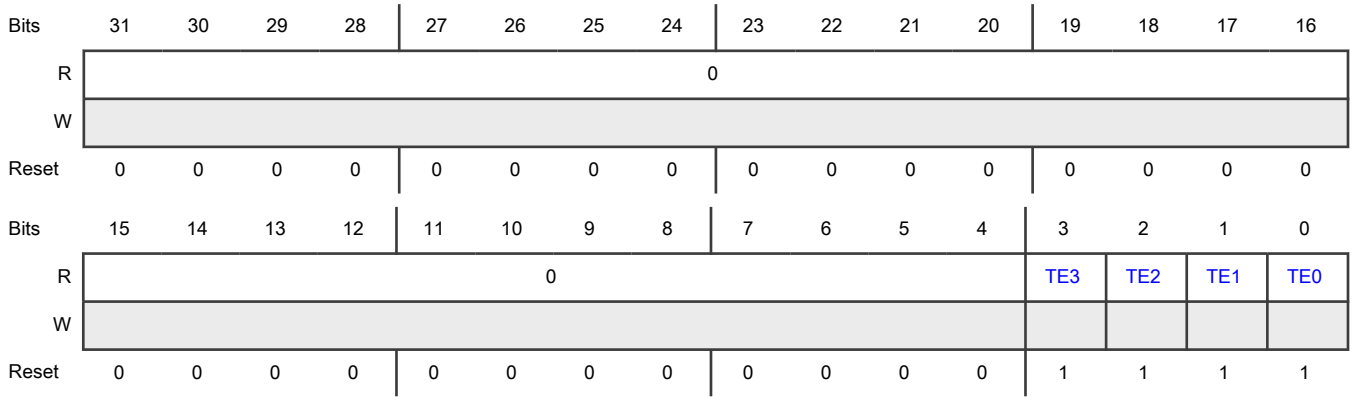
Offset

Register	Offset
TSR	124h

Function

Indicates whether the MUB transmit registers are empty.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TEn	MUB Transmit Empty Indicates whether MUB Transmit (TR*n*) register is empty, where *n* = 0 to 3. This field becomes 1 after the MUA_RR*n* register is read on the MUA side. When TE*n* = 1, it indicates to the MUB side that the MUB_TR*n* register is ready to be written on the MUB side. If MUB_TCR[TIE*n*] = 1, a transmit *n* interrupt is issued on the MUB side. This field becomes 0 after the MUB_TR*n* register is written to on the MUB side. After this field becomes 0, if MUB_TCR[TIE*n*] = 1, the transmit *n* interrupt request is cleared on the MUB side. This field becomes 1 when MU resets. 0b - Not empty 1b - Empty

26.7.2.17 Receive Control (RCR)

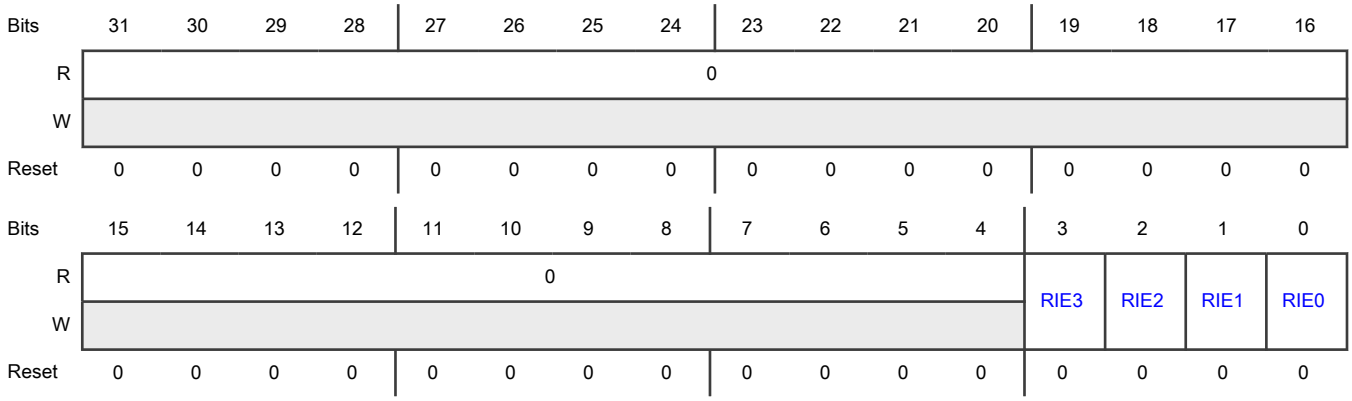
Offset

Register	Offset
RCR	128h

Function

Contains the MUB receive interrupt enables.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 RIEn	MUB Receive Interrupt Enable Enables MUB receive interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUB receive interrupt *n* request is issued when MUB_RSR[RF*n*] is set. If this field is 0, MU ignores the value of MUB_RSR[RF*n*], and no MUB receive interrupt request is issued. 0b - Disable 1b - Enable

26.7.2.18 Receive Status (RSR)

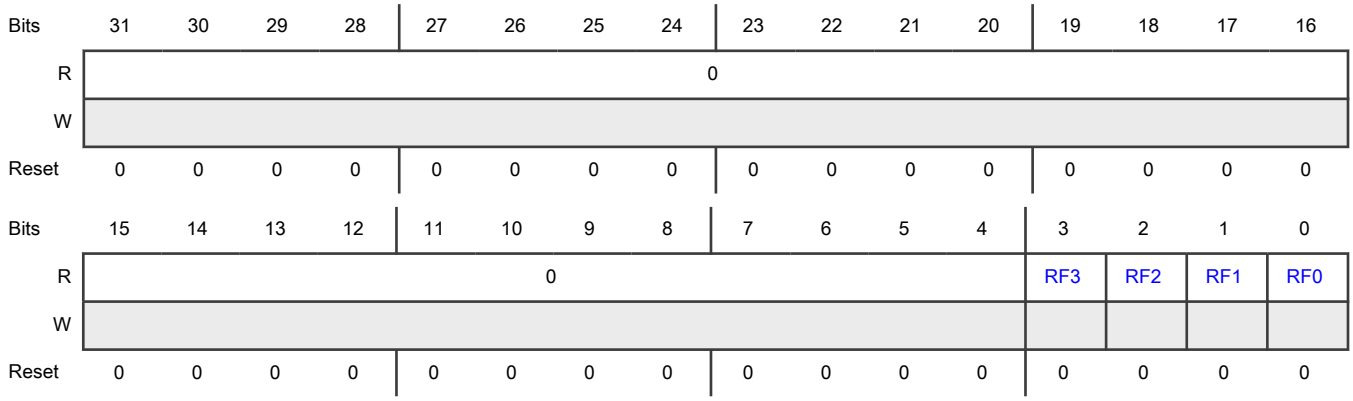
Offset

Register	Offset
RSR	12Ch

Function

Indicates whether the MUB receive registers are full.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 RFn	MUB Receive Register Full Indicates whether MUB Receive register (RR*n*) is full, where *n* = 0 to 3. RF*n* becomes 1 when the MUA_TR*n* register is written to on the MUA side. When RF*n* is 1, it indicates to the MUB side that new data in the MUB_RR*n* register is ready for MUB to read it. If MUB_RCR[RIE*n*] = 1, a receive *n* interrupt is issued on the MUB side. RF*n* becomes 0 when the MUB_RR*n* register is read, or when MU is reset. After RF*n* becomes 0, if MUB_RCR[RIE*n*] = 1, the receive *n* interrupt request is cleared on the MUB side. 0b - Not full 1b - Full

26.7.2.19 Transmit (TR0 - TR3)

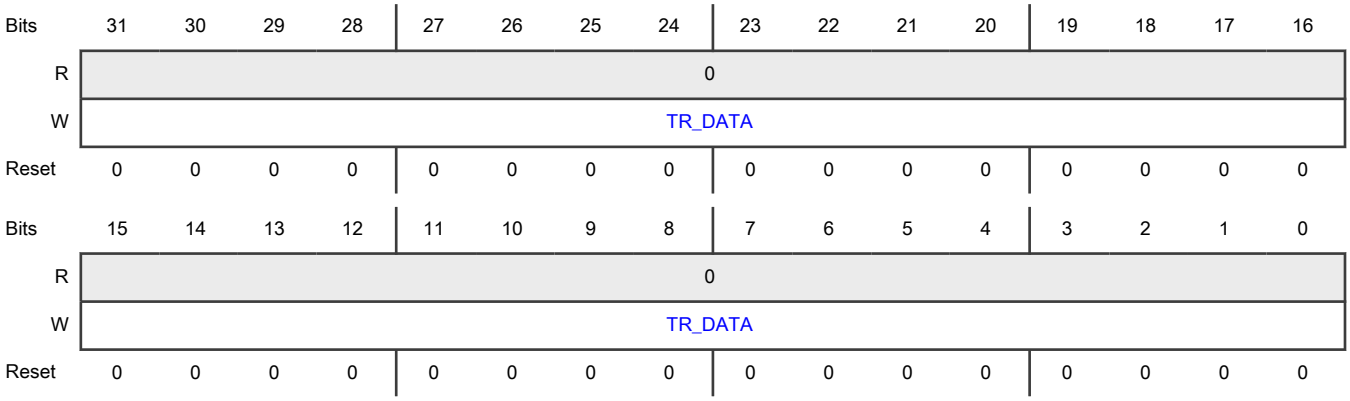
Offset

Register	Offset
TR0	200h
TR1	204h
TR2	208h
TR3	20Ch

Function

Contains MUB transmit data.

Diagram



Fields

Field	Function
31-0 TR_DATA	MUB Transmit Data Contains MUB transmit data. MUA_RR*n* reflects the data written to this register. The TR*n* and RR*n* registers are not double-buffered. Writing to MUB_TR*n* overrides the data readable in the MUA_RR*n* register. A write to the Transmit register clears MUB_TSR[TE*n*] on the transmitter side, and sets MUA_RSR[RF*n*] on the receiver side. You can write to this register only when MUB_TSR[TE*n*] = 1. Reading this register returns all zeroes.

26.7.2.20 Receive (RR0 - RR3)

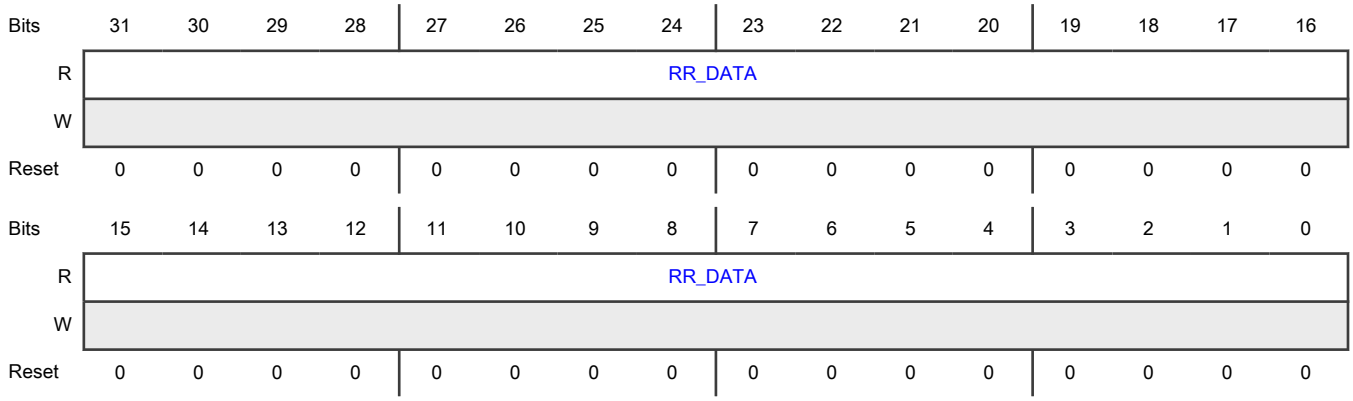
Offset

Register	Offset
RR0	280h
RR1	284h
RR2	288h
RR3	28Ch

Function

Contains MUB receive data.

Diagram



Fields

Field	Function
31-0 RR_DATA	MUB Receive Data Reflects the data written to MUA TR*n*. Reading this register clears MUB_RSR[RF*n*] on the receiver side, and sets MUA_TSR[TE*n*] on the transmitter side. You can read this register only when MUB_RSR[RF*n*] = 1. Reading it before MUB_RSR[RF*n*] becomes 1 may result in reading incorrect data. Poll MUB_RSR[RF*n*] to confirm it is set before reading RR*n*. Writing to this register generates an error response to MUB.

Chapter 27
Power Management Controller (PMC)

27.1 Chip-specific PMC information

Table 225. Reference links to related information

Topic	Related module	Reference
Full description	PMC	PMC
Peripheral memory map		Memory map overview
Power overview		Power overview
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Sleep Controller	SLEEPCON	SLEEPCON

NOTE

The device has one analog voltage glitch detector (aGDET) and two digital voltage glitch detectors (GDET) which enhance protection against voltage manipulation. The NXP boot ROM configures aGDET and GDET during device initialization to reset the device when voltage manipulation is detected. The aGDET and GDET were validated on silicon across multiple process, voltage, and temperature corners with maximum noise profile generated by activating a large number of possible boot paths. To avoid false triggering due to unforeseen environment or application behavior, NXP recommends disabling aGDET and GDET in early start-up code. You can use aGDET and/or GDET in the application when security-critical operations are taking place, e.g., cryptographic operation using private or symmetric keys (digital signature, encryption, decryption, etc.), or sensitive decision-making process. However, characterization of the system noise profile and interactions with the sensors are required.

27.1.1 Module instances

This chip has two instances of the PMC module: PMC0 and PMC1.

Table 226. PMC instances

	PMC0	PMC1
Located in	VDD1V8_PMC	VDD1V8_PMC
Access	CPU0, eDMA0, eDMA1	CPU1, eDMA2, eDMA3

27.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

27.1.3 Clock for PMC register access

The PMC instances are attached to the VDD1_SENSE bus. For either CPU to be able to access the PMC registers, SENSE_MAIN_CLK_1 must be enabled (controlled by RUNCFG[SENSES_MAINCLK_SHUTOFF] and/or SLEEPCFG[SENSES_MAINCLK_SHUTOFF] fields of the SLEEPCON0 and SLEEPCON1).

27.2 Overview

PMC controls the power-related functions of the chip, including the following:

- Internal regulators
- Voltage and current references
- Supply voltage monitors
- Power switches
- Asymmetric forward body bias (AFBB) and reverse body bias (RBB)
- Oscillators for the PMC analog and logic blocks
- Temperature sensor
- SRAM power switches
- State retention and reset of the GPIOs
- Logic to respond to both PMC and SoC initiated mode changes.

The PMC control logic includes state machines and controls for regulator modes, supply levels, power switch sequencing, PMC low-power mode entry and exit sequencing, event triggered wake-ups and interrupts, memory supply power switch control, pad reset and state retention, and the PMIC_MODE[1:0] and PMIC_IRQN pins control interface.

The power management functions support the defined power modes of the chip, the power and reset sequences controlled by the chip's reset and mode controls, and miscellaneous power control and monitoring functions. This chapter covers the hardware details of the PMC digital logic, analog submodules, and the functions they support.

27.2.1 Block diagram

The block diagram below shows the analog submodules controlled by the PMC logic, the logic blocks, and the main power supplies used and/or generated by PMC.

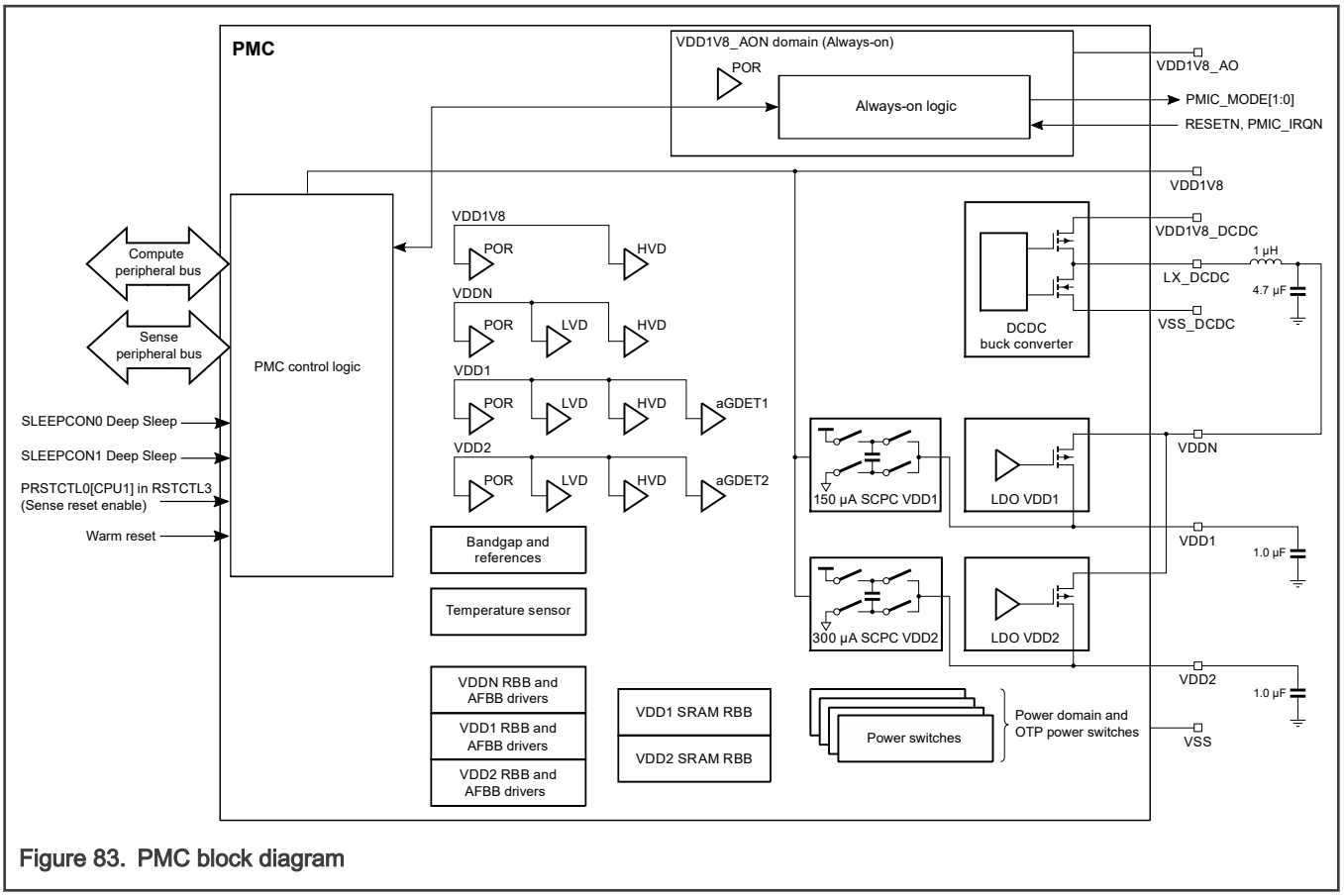


Figure 83. PMC block diagram

27.2.2 Features

- Three separate core supplies (VDDN, VDD1, VDD2) to support the lowest possible voltages needed by the various logic and memory domains, and some analog blocks' interfaces.
- Ultra low-power DC to DC (DCDC) buck regulator with a low-power mode for optimum efficiency over a wide load current range to optionally power VDDN.
- Two (2x) low-dropout (LDO) regulators with bypass mode, supplied by VDDN, to optionally power VDD1 and VDD2.
- Two (2x) switched capacitor power converters (SCPC) to optionally power VDD1 and VDD2, supports state retention of logic and memory, while VDDN is powered down.
- Multiple application supply configurations:
  - Supply the 3 low-voltage core supply domains using the on-chip DCDC for VDDN and the two on-chips LDOs and two on-chip SCPC regulators for VDD1 and VDD2
  - Supply VDDN externally, with VDD1 and VDD2 supplied by the internal LDOs or SCPCs
  - All low-voltage supplies from an external PMIC.
- Analog PMC block containing a band gap reference, current and voltage references, voltage monitors, clock sources, and a temperature sensor.
- Support for reduced supply voltage and lower active power using asymmetric forward body biasing (AFBB) on low-voltage logic.
- Support full-logic state retention at very low-power including reverse body biasing (RBB) for the core logic and SRAMs.
- Low-leakage power switches for each internal low-voltage domain.

- Power mux for internal VDDN_SR rail allows the VDDN supply to power off while the VDDN logic and memory are retained using VDD2 as the supply.
- 1.8 V PMC control logic (VDD1V8):
  - Hardware sequencing for isolation, references, voltage monitors, regulators, body bias, logic power switches and mux, SRAM power switches, and OTP power switch during mode transitions.
  - Generate logic POR for 1.8 V (VDD1V8) and low-voltage domains.
  - Monitor supply voltages for overvoltage (HVD) and low voltage (LVD) and analog glitch detection (aGDET) and generate corresponding interrupts and/or resets.
  - Monitor analog submodules for error conditions and generate corresponding interrupts and/or resets.
  - Provide software and hardware interface to the chip's power management functions (CPUs and SLEEPCON blocks).
  - Provide enable bits for triggering interrupts, wake-up, and resets from various detectors, timers, etc.
  - Provide status and sticky flag bits for software polling and interrupt service routine (ISR) acknowledgement.
  - Automatic pad state retention controls.
- 1.8 V PMC always on control logic (VDD1V8_AO):
  - RESETN, PMIC_IRQN, and PMIC_MODE pad interface
  - RTC wake-up interface logic
  - Always-on VDD1V8_AO POR

27.3 Functional description

The supported domains and operations of hardware logic blocks and connections of PMC are covered in the sections below.

NOTE

PMC includes two sets of registers used to configure operation in Active mode and low-power modes. Many control options are duplicated between these two sets of registers, so throughout this chapter, "*" is used in register names, where the * can be replaced by "RUN" or "SLEEP." For example, PD*CFG0 refers to PDRUNCFG0 and PDSLEEPCFG0 registers.

27.3.1 Analog submodules

27.3.1.1 Regulators

There are five on-chip regulators in PMC. They can be used to supply the low-voltage power rails in the chip, or can be disabled by software when the low-voltage rails of the chip are powered by external supplies. Mixed on-chip and off-chip supply configurations are also supported.

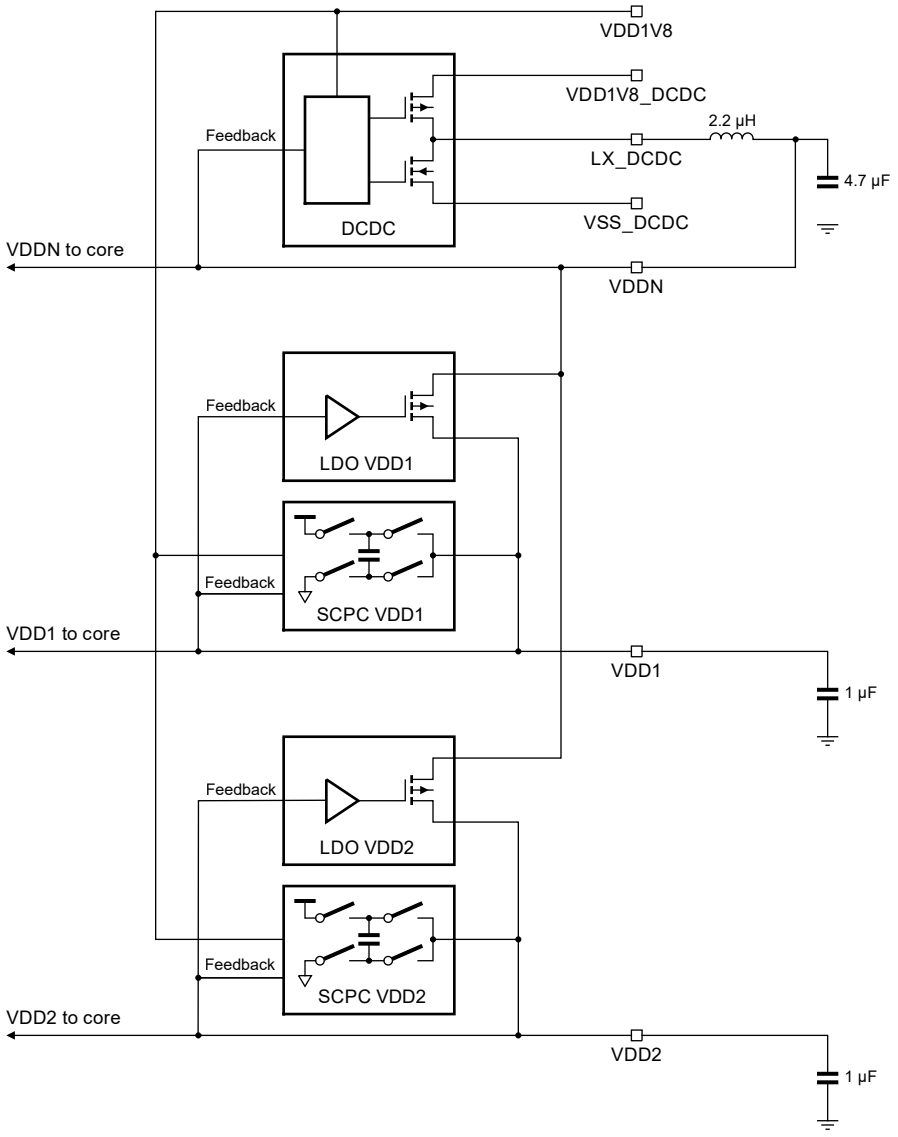


Figure 84. PMC regulators and associated package pins

27.3.1.1.1 DCDC buck converter (DCDC)

The on-chip DCDC Buck Converter is a step down inductive switched regulator that converts the 1.8 V input supply down to a 0.5 to 1.15 V level. This is optionally used to power the VDDN supply for logic and the 2 on-chip LDO regulators.

The DCDC features include:

- Input range from 1.71 to 1.98 V
- Output range programmable from 0.5 V to 1.15 V in steps of 6.25 mV (see DCDCVSEL registers)
- High efficiency even at small loads
- High-power (HP) mode supports active chip power modes.
- Low-power (LP) mode supports low activity modes at higher efficiency than HP mode
- Ultra-low-power (ULP) mode supports retention modes while maintaining high efficiency

- Provides overcurrent warnings that can trigger interrupts.

The DCDC control interface uses these registers:

- Power mode: PD*CFG0[DCDC_MODE]
- Voltage level: PD*CFG0[DCDC_VSEL] and DCDCVSEL registers
- Disable DCDC when using external VDDN by setting POWERCFG[DCDCPD]
- Enable DCDC during FDSR mode by clearing POWERCFG[FDDCPD]
- Overcurrent warning flags and interrupt enables: FLAGS and INTRCTRL

27.3.1.1.2 Low-dropout (LDO) regulators

PMC has two LDO regulators, LDO VDD1 for VDD1 and LDO VDD2 for VDD2 supplies respectively. They can be powered down when VDD1 and VDD2 are externally powered.

These regulators' output drivers are supplied by VDDN and drop the voltage by as little as 50 mV in bypass mode, or regulate down to as low as 0.5 V in either high-power (HP) or low-power (LP) regulation modes.

The output voltages and power modes are separately controllable, allowing the VDD1 and VDD2 logic supplies to be independently optimized to support the required logic performance.

The LDO features include:

- High-power mode with fast load step response
- Low-power regulation mode with low quiescent current consumption
- Low-power bypass mode with low quiescent current consumption and only a 50 mV drop from VDDN (see PD*CFG0[LDOn_MODE])
- Configurable output voltage from 0.5 V to 1.15 V by the step of 12.5 mV
- Soft start up by limiting the output current

The LDO control interface uses these registers:

- Power mode: PD*CFG0[LDOn_MODE]
- Voltage level: PD*CFG0[LDOn_VSEL] and LDOVDDnVSEL registers
- Disable LDO when using external VDD1 and VDD2 by setting POWERCFG[LDOnPD]
- Enable LDO during FDSR mode by clearing POWERCFG[FDLDOnPD]

27.3.1.1.3 Switched capacitor power converters (SCPC)

The on-chip switched capacitor power converters, SCPC VDD1 and SCPC VDD2, optionally supply the state retained logic and SRAM memory arrays during full deep sleep retention (FDSR) mode. These regulators divide the VDD1V8 supply by a ratio of about 3 to supply 0.5 V. When the SCPCs are used to power VDD1 and VDD2 in FDSR mode, LDO VDD1 and LDO VDD2 can be powered down. This is the default configuration in POWERCFG.

- One to three ratio down conversion
- The SCPC's undervoltage flag is set when the voltage drops too low to support logic retention. This event can be configured to trigger an interrupt.
- High efficiency at very low loads

The SCPC control interface uses these registers:

- PDSLEEPCFG0[FDSR]: Set in both Compute and Sense domain to enter FDSR mode when both CPUs enter Deep Sleep, enabling the SCPCs depending on the POWERCFG settings.
- POWERCFG[SCPCPD] = 1 disables both regulators all the time. Set this bit when application powers VDD1 and VDD2 externally.

- POWERCFG[FDSCPnPD]: Set to disable the SCPCs in FDSR mode when using the on chip LDOs during FDSR mode.

27.3.1.2 PMC voltage and current references

The PMC voltage and current references are used inside the PMC subsystem and available for blocks outside PMC like ADC.

The band gap, voltage, and current references automatically go to high-power mode if any PMC module (e.g. regulators, monitors, or temperature sensor) needs them. For external modules that need the high-power references, software should clear the [PMCREF_LP] field in the appropriate PD*CFG1 register to force the references into high-power mode.

The high-power references are turned off when no PMC blocks need high-power references and the aggregated PD*CFG1[PMCREF_LP] bit is set.

The SCPC regulators only use the low-power references, so they never cause the references to go to high power mode. The LDO regulators use only low-power references when in bypass or low-power mode. However, the LDO regulators automatically enable the high-power references when in high-power mode. The DCDC, PORs, and LVDs use high-power references when they are in their respective high-power modes, so force the references into high-power mode when they are in high-power mode. The HVDs, aGDETs, and PMC temperature sensor use high-power references so force the references into high-power mode when they are enabled.

27.3.1.3 Supply voltage monitors

There are four different types of voltage monitors in PMC: PORs, HVDs, LVDs, and aGDETs. Each monitor type is described below, including which voltage supplies are monitored by each type.

27.3.1.3.1 Power-on reset (POR)

There are five power-on reset modules that provide undervoltage detection for these supply rails:

- VDD1V8_AO
- VDD1V8
- VDDN
- VDD1
- VDD2

POR modules always trigger cold resets when tripped:

- The VDD1V8_AO POR triggers a full chip reset, including the always on logic.
- The VDD1V8 POR triggers a full chip reset, except for a few bits in the FLAGS register.
- The VDD1, VDD2, and VDDN PORs trigger cold resets that reset all the PD*CFG registers to the default values, including the memory and logic power switch controls.

The POR modules will hold the chip in reset until the supplies exceed the trip point. They assert reset when the supplies drop below the trip point minus a small amount of hysteresis. The VDD1, VDD2, and VDDN POR resets are forced active when PMC is in deep power down (DPD) mode or full deep power down (FDPD) mode. The VDD1V8 POR is forced active when PMC is in FDPD mode except when configured to use for wake-up by clearing POWERCFG[FDPDBGLP], which causes PMC to exit FDPD when VDD1V8 drops below the trip point minus hysteresis then rises back up above the trip point. Note this keeps the voltage references in high-power mode which consumes about 85 µA.

The VDD1, VDD2, and VDDN PORs have programmable trip points. The VDD1V8_AO and VDD1V8 PORs have fixed trip points.

27.3.1.3.1.1 VDD1, VDD2, VDDN POR features

- Programmable falling trip point from 0.40 V to 0.71 V (32 steps of 10 mV)
- Always enabled when VDD1V8 is powered
- Low-power or high-power mode
- When in high-power mode it automatically puts the references into high-power mode

27.3.1.3.1.2 VDD1V8 POR features

- Fixed falling trip voltage
- Always enabled when VDD1V8 is powered
- Low-power or high-power mode based on reference voltage power mode

27.3.1.3.1.3 VDD1V8_AO POR features

- Fixed falling trip voltage = 1.46 V (typical)
- Always enabled when VDD1V8_AO is powered
- Always in low-power mode

27.3.1.3.2 High Voltage Detectors (HVD)

There are four HVDs, one each for VDD1, VDD2, and VDDN, and one for VDD1V8. Each HVD can optionally trigger a reset and interrupt, or simply set their FLAGS and STATUS bits.

The HVDs are enabled or powered down by their respective aggregated PD*CFG bits. When enabled the references are automatically put in high-power mode.

27.3.1.3.2.1 HVD features

- HVDs can be powered down
- HVDs do not have a low-power mode, so the references are forced high-power mode when an HVD is enabled
- VDD1, VDD2, VDDN assertion (rising VDD) trip voltage = 1.25 V (typical)
- VDD1V8 assertion trip voltage = 2.115 V (typical)

27.3.1.3.3 Low Voltage Detectors (LVD)

There are three LVDs, one each for VDD1, VDD2, and VDDN. Each can optionally trigger a reset, an interrupt, or simply set their FLAGS and STATUS bits.

The LVDs are also used to detect the supply rails when using off chip supplies for the core supplies. When PMC changes modes, the LVDs must detect voltages higher than the currently selected trip points to complete the mode transition.

The LVD trip points are selected by the aggregated PD*CFG0[DCDC_VSEL] for VDDN which selects one of two values from the LVDVDDNCTRL register, the PD*CFG0[LDO1_VSEL] field which selects one of four values from the LVDVDD1CTRL register, and the PD*CFG0[LDO2_VSEL] field which selects one of four values from the LVDVDD2CTRL register.

27.3.1.3.3.1 LVD features

- Trimmable falling trip point from 0.5 V to 1.13 V (64 steps of 10 mV)
- Always enabled when VDD1V8 is powered
- Low-power or high-power mode
- When in high-power mode it automatically puts the references into high-power mode

27.3.1.3.4 Analog Glitch Detectors (aGDET)

There are two aGDETs, one each for VDD1 and VDD2. Each can optionally trigger a reset, an interrupt, or simply set their FLAGS bits when the supply drops quickly or VSS spikes up quickly. These blocks are intended to detect voltage attacks by malicious users.

The aGDETs are enabled when the aggregated PD*CFG1[AGDET1_PD] and/or [AGDET2_PD] fields are cleared. When the aGDETs are enabled, the references are in high-power mode.

27.3.1.3.4.1 aGDET features

- The aGDET is triggered by low going supply glitches or high going VSS glitches of 300 mV or more, but is not triggered by glitches less than 60 mV in magnitude to avoid triggering due to normal supply noise. Glitches with magnitudes between 60 mV and 300 mV may trigger the aGDET. Typically glitches of about 150 mV in magnitude will trigger the aGDET.
- When enabled the aGDET detector automatically puts the references into high power mode

27.3.1.4 Power switches and power mux

There are nine power switches implemented in PMC. All are controlled by the aggregated values of their respective PD*CFG PD (Power Down) or DSR (Deep Sleep Retention) bits: [V2COMP_DSR], [V2NMED_DSR] (controls both VDDN_MEDIA and VDD2_MEDIA switches), [V2COM_DSR], [VNCOM_DSR], [V2DSP_PD], [V2MIPI_PD], and [OTP_PD].

There is also one power mux. It is used during FDSR mode to switch the supply for the VDDN powered logic over to VDD2, allowing the VDDN supply to be powered off.

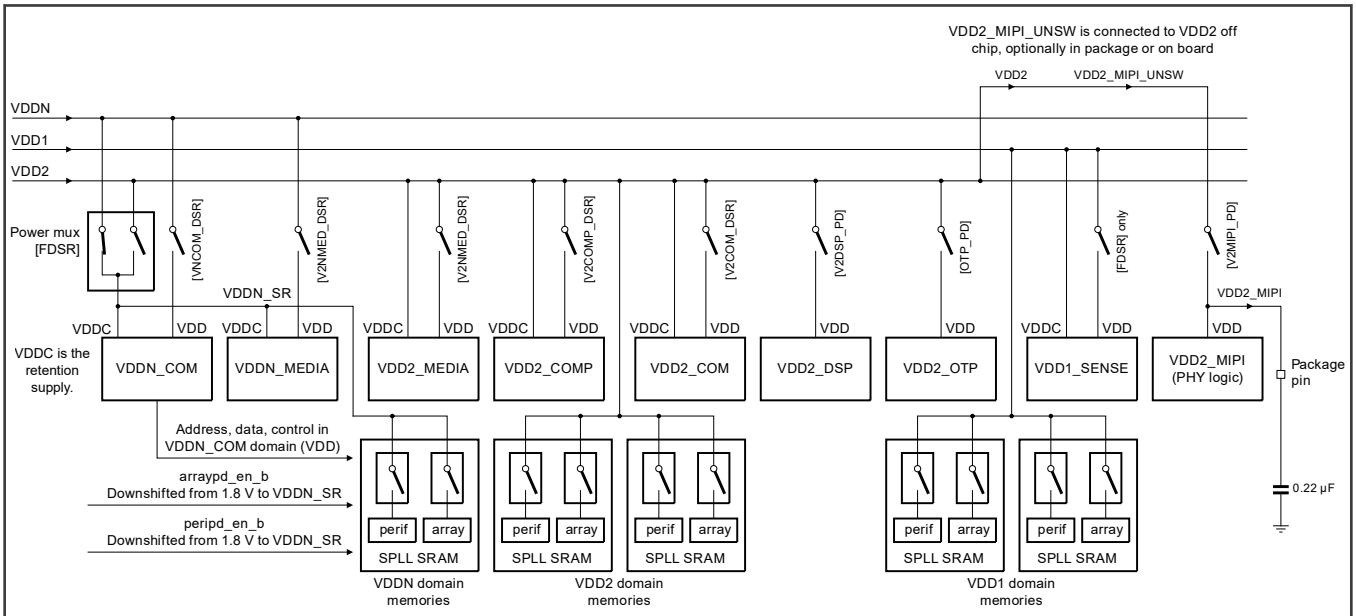


Figure 85. Power switches and power mux diagram

It is the software's responsibility to ensure that the following restrictions are maintained. The [FDPD], [DPD], and [FDSR] bits override the other PD*CFG power switch controls. When all of these 3 bits are clear, then the individual power gates may be enabled or disabled as indicated. Software is required to set the PD/DSR bits as shown in the table below when the bit in the left column is set. An 'X' in the other columns indicates when a bit may be either asserted or negated.

Table 227. Power switch restrictions

Aggregated PD*CFG bit is set and domain power-gated	Required value of other PD*CFG bits and supplies													
	[FDPD]	[DPD]	[FDSR]	[V2COM_DSR]	[V2COMP_DSR]	[V2DSP_PD]	[V2NMED_DSR]	[VNCOM_DSR]	[OTP_PD]	VDD1	VDD2	VDDN	VDD1V8	VDD1V8_AO
[FDPD]	-	X	X	X	X	X	X	X	X	X	X	X	X	On
[DPD]	0	-	X	X	X	X	X	X	X	X	X	X	On	On

Table continues on the next page...

Table 227. Power switch restrictions (continued)

Aggregated PD*CFG bit is set and domain power-gated	Required value of other PD*CFG bits and supplies													
	[FDPD]	[DPD]	[FDSR]	[V2COM_DSR]	[V2COMP_DSR]	[V2DSP_PD]	[V2NMED_DSR]	[VNCOM_DSR]	[OTP_PD]	VDD1	VDD2	VDDN	VDD1V8	VDD1V8_AO
[FDSR]	0	0	-	X	X	X	X	X	X	On	On	X	On	On
[V2COM_DSR]	0	0	0	-	1	1	1	1	1	On	On	On	On	On
[V2COMP_DSR]	0	0	0	X	-	1	X	X	X	On	On	On	On	On
[V2DSP_PD]	0	0	0	X	X	-	X	X	X	On	On	On	On	On
[V2NMED_DSR]	0	0	0	X	X	X	-	X	X	On	On	On	On	On
[VNCOM_DSR]	0	0	0	X	X	X	1	-	X	On	On	On	On	On
[OTP_PD]	0	0	0	X	X	X	X	X	-	On	On	On	On	On

27.3.1.5 Logic body bias generators

Body biasing consists of applying a voltage to the target transistors' back terminal to control their threshold voltage. Asymmetric forward body bias (AFBB) allows for higher performance while minimizing the increase in leakage current. Reverse body bias (RBB) reduces the leakage current consumption but also performance so is used only for logic that is not clocked.

The logic body bias is applied per main voltage domain (VDD1, VDD2, and VDDN) to allow each to be independently biased based on their supply voltage levels and performance needs.

AFBB is enabled at power up. The applied voltage is fixed.

RBB mode can be enabled via PDSLEEPCFG0 for Deep Sleep modes. Asynchronous activity is allowed while RBB is enabled. The RBB voltage can be selected in BBCTRL for each domain. Generally the application should use 1.3 V to get the smallest amount of leakage, but it is possible that the 1.0 V setting achieves lower total power for applications that frequently toggle between RBB and AFBB because the charge needed to switch modes is smaller with a smaller RBB voltage.

NOTE

For applications where one CPU domain never uses one of the voltage domains, and so the other CPU domain should have complete control, the first domain should configure the body bias for that domain to be RBB (the lowest power mode) so that aggregation will always use the second domain's PD*CFG to control that voltage domain's body bias. For example, if the Compute CPU never uses any VDD1 logic, then the Compute slot should be configured with PDRUNCFG0[AFBB1_PD] = 1, PDRUNCFG0[RBB1_PD] = 0, and the same for PDSLEEPCFG0, so regardless of the activity in the Compute domain, the Sense domain has complete control of the VDD1 body bias.

NOTE

See Enabling body bias for low-power modes for more information on controlling the Logic Body Bias.

27.3.1.6 SRAM reverse body bias generators

Similar to the logic body bias generators, there are two SRAM body bias generators, one for SRAMs in VDD1 and the other for SRAMs in VDD2. These should only be enabled when none of the memories in that voltage domain are being accessed, which is typically when that voltage domain is in a low-power mode. They are fixed voltage and require no configuration other than turning them on via PD*CFG0 registers. To enable SRAM RBB for a voltage domain's SRAMs, clear the associated [RBBSRn_PD].

When the generators are powered down, the SRAMs operate in normal body bias (NBB) mode.

27.3.1.7 PMC temperature sensor

To enable the temperature sensor, the sensor must be enabled by clearing PDRUNCFG1[TEMP_PD] and then setting CTRL[APPLYCFG] to activate PDRUNCFG1 change.

NOTE

The temperature sensor is automatically disabled and powered down when the chip is in DPD or FDPD mode.

27.3.1.7.1 Using the temperature sensor

To configure system for measurement:

- Enable PMC temperature sensor (PDRUNCFG1[TEMP_PD] = 0), set CTRL[APPLYCFG] to trigger the state machine update, wait for STATUS[BUSY] to go low, then wait another 40 µs for the temperature sensor to settle.
- Select ADC differential input channel connected to PMC temperature sensor.
- Configure ADC sampling time to (at least) 15 µs.

Measurement procedure:

1. Set TSENSOR[TSENSM] = 000b, perform AD conversion, and record C1_000
2. Set TSENSOR[TSENSM] = 001b, perform AD conversion, and record C1_001
3. Set TSENSOR[TSENSM] = 011b, perform AD conversion, and record C1_011
4. Set TSENSOR[TSENSM] = 010b, perform AD conversion, and record C1_010
5. Set TSENSOR[TSENSM] = 110b, perform AD conversion, and record C1_110
6. Set TSENSOR[TSENSM] = 111b, perform AD conversion, and record C1_111
7. Set TSENSOR[TSENSM] = 101b, perform AD conversion, and record C1_101
8. Set TSENSOR[TSENSM] = 100b, perform AD conversion, and record CM_100
9. Set TSENSOR[TSENSM] = 101b, perform AD conversion, and record C2_101
10. Set TSENSOR[TSENSM] = 111b, perform AD conversion, and record C2_111
11. Set TSENSOR[TSENSM] = 110b, perform AD conversion, and record C2_110
12. Set TSENSOR[TSENSM] = 010b, perform AD conversion, and record C2_010
13. Set TSENSOR[TSENSM] = 011b, perform AD conversion, and record C2_011
14. Set TSENSOR[TSENSM] = 001b, perform AD conversion, and record C2_001
15. Set TSENSOR[TSENSM] = 000b, perform AD conversion, and record C2_000

Calculate die-temperature using the formula:

$$T[\text{in } °C] = n^{-1} * 370.98 * (CM_TEMP/CM_VREF) - 273.15$$

Where:

- n = 1.0036
- CM_TEMP = (2*C1_000 - C1_010 + 2*C2_000 - C2_010 + 4*CM_100 - C1_110 - C2_110) / 4
- CM_VREF = CM_CTAT + (953.36 + TSENS_CAL) * CM_TEMP * 2^{-11}
  - CM_CTAT = (2*C1_001 - C1_011 + 2*C2_001 - C2_011 + 2*C1_101 - C1_111 + 2*C2_101 - C2_111) / 4
  - TSENS_CAL is an 8-bit signed calibration constant retrieved from the OCOTP fuse word index [2464].

NOTE

- Multiplications and divisions by 2 or 4 may be performed through simple shift-register operations.
- See ADC chapter for information on how to configure the ADC.
- The PMC references are put into high-power mode whenever the PMC temperature sensor is enabled.

27.3.2 PMC control

27.3.2.1 Operating modes

The PMC operating mode depends on a few inputs from the chip, a few register settings in PMC, and changes mode when either Sense or Compute (or both) CPUs enter or wake up from Deep Sleep mode, the Sense or Compute CPU is in Deep Sleep and its SLEEPCON enters or exits a hardware wake event (e.g. a DMA access), a warm or cold reset occurs, or the chip clears PRSTCTL0[CPU1] in RSTCTL3.

A hardware wake in a given domain's SLEEPCON causes PMC to enter Active mode for that domain, using PDRUNCFG instead of PDSLEEPCFG while the hardware wake event is active, then revert to PDSLEEPCFG afterwards. The domain's CPU remains in Deep Sleep during the hardware wake event.

It is possible for both CPUs to change their Deep Sleep state on the same clock cycle and PMC will correctly change modes using the aggregated values of the PD*CFG registers based on both CPUs' new states.

Table 228. PMC operating modes

Mode	Compute CPU Deep Sleep	Sense CPU Deep Sleep	Sense domain reset	Warm reset	Cold reset	Aggregated PDSLEEPCFG			Mode description
						[FDSR]	[DPD]	[FDPD]	
VDD1V8_AO POR	-	-	-	-	VDD1V8_AO POR	-	-	-	All logic is reset.
VDD1V8 POR	-	-	-	-	VDD1V8_POR or FDPD wake-up	-	-	-	Resets all PMC logic, except Always-on logic, including some FLAGS register bits.
Cold reset	-	-	-	-	VDD1, VDD2, or VDDN POR, LVD, HVD, or aGDET reset or VDD1V8 HVD	-	-	-	Resets most PMC logic except for the FLAGS and PWRFLAGS registers
Compute Active / Sense Reset	0	-	1	0	0	-	-	-	Compute PDRUNCFG controls PMC
Compute Active / Sense Active	0	0	0	0	0	-	-	-	Aggregated Compute PDRUNCFG + Sense PDRUNCFG controls PMC
Compute Active / Sense Deep Sleep	0	1	0	0	0	-	-	-	Aggregated Compute PDRUNCFG + Sense PDSLEEPCFG controls PMC

Table continues on the next page...

Table 228. PMC operating modes (continued)

Mode	Compute CPU Deep Sleep	Sense CPU Deep Sleep	Sense domain reset	Warm reset	Cold reset	Aggregated PDSLEEPCFG			Mode description
						[FDSR]	[DPD]	[FDPD]	
Compute Deep Sleep / Sense Active	1	0	0	0	0	-	-	-	Aggregated Compute PDSLEEPCFG + Sense PDRUNCFG controls PMC
Compute Deep Sleep / Sense Deep Sleep	1	1	0	0	0	0	0	0	Aggregated Compute PDSLEEPCFG + Sense PDSLEEPCFG controls PMC
FDSR	1	1	0	0	0	1	0	0	All low-voltage logic power-gated, most state retained. SRAM arrays optionally retained.
DPD	1	1	0	0	0	-	1	0	All chip's low-voltage logic powered off
FDPD	1	1	0	0	0	-	-	1	All chip's low-voltage and VDD1V8 logic powered off

NOTE

When PRSTCTL0[CPU1] in RSTCTL3 = 1 , the aggregated PD*CFG is the same as the Compute PD*CFG. The Sense domain's PD*CFG is ignored.

If there is a mode change in progress, or the PMC state machines are busy due to software setting CTRL[APPLYCFG], the mode change will occur after the current change completes. Software can poll CTRL[APPLYCFG] or the STATUS[BUSY] to see if an operation is in progress.

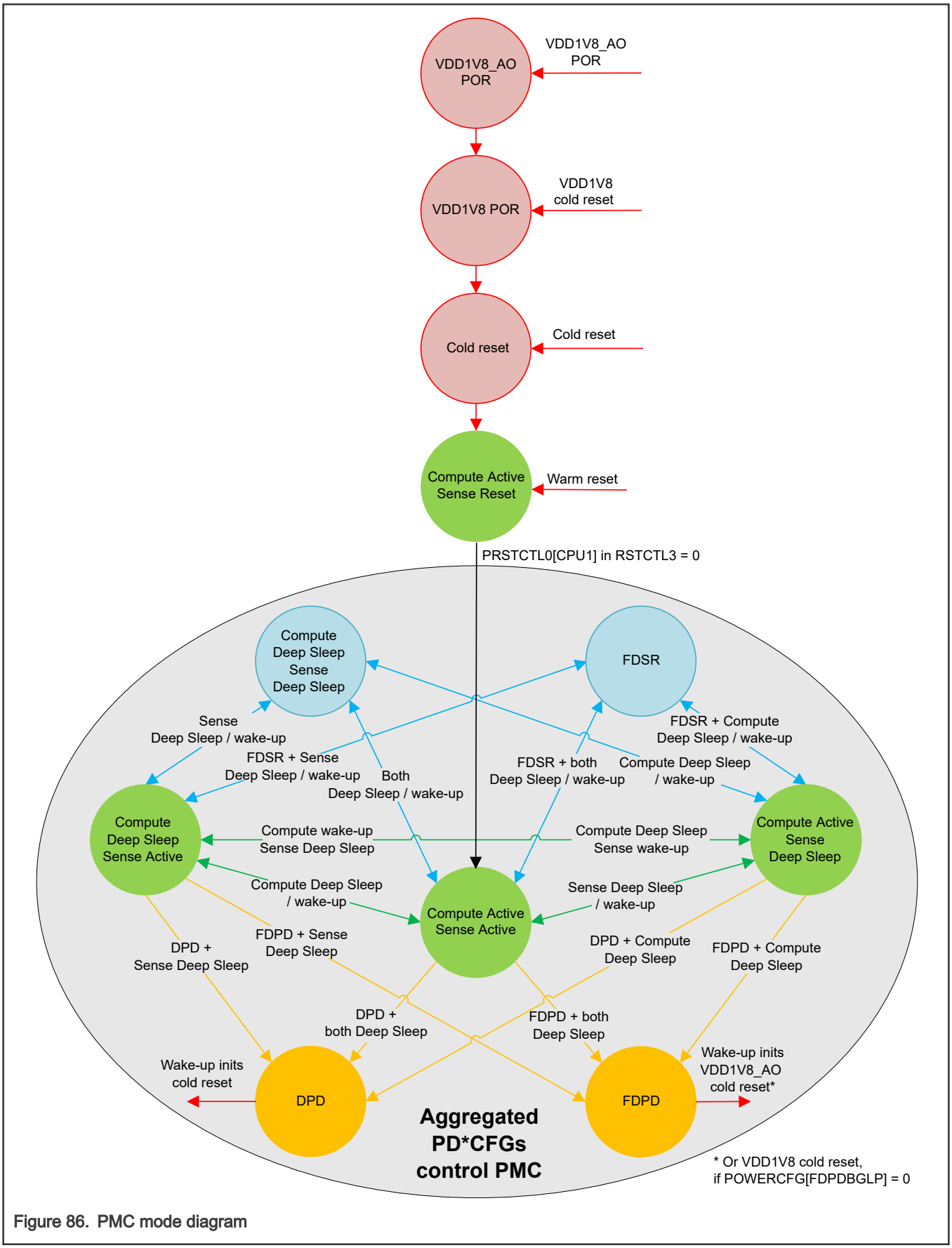


Figure 86. PMC mode diagram

27.3.2.2 PD*CFG and aggregation

PD*CFG refers to PDRUNCFG and PDSLEEPCFG registers that control the PMC block power modes, chip power modes, and core domain voltage levels. There are two sets of PDRUNCFG and PDSLEEPCFG registers, one set for the Compute domain and one for the Sense domain. For each domain, either the PDRUNCFG or the PDSLEEPCFG registers are selected based on that domain's CPU state. When a CPU is in Deep Sleep mode then it's PDSLEEPCFG registers are selected, otherwise it's PDRUNCFG registers are selected.

The selected registers for the Compute domain and the Sense domain are ***aggregated*** and used to control the power states of the PMC submodules, the voltage levels in the core domains, and the power switches for the core sub-domains and SRAMs. The aggregation compares the Compute and Sense domain PD*CFG inputs and selects the higher power state for each control field. For example, the Compute domain CPU is in Deep Sleep and its PDSLEEPCFG0[DCDC_VSEL] = 0, and the Sense domain CPU is active and its PDRUNCFG0[DCDC_VSEL] = 1, then the aggregated DCDC voltage select = 1 because that is higher power, allowing the Sense domain to run at the power level it needs while the Compute domain is in Deep Sleep and simplifying the software coordination needed between the two CPUs.

The aggregation generally uses the lower binary value of the bit fields as the higher power mode, since most fields are "power down" bits, but there are some exceptions in PD*CFG0: [DCDC_VSEL], [LDO1_VSEL], [LDO2_VSEL], [RBB1_PD], [RBB2_PD], [RBBN_PD], [RBBSR1_PD], [RBBSR2_PD] all chose the higher binary value because that is the higher power mode for these fields.

A new aggregated value is calculated any time the PMC state machines are triggered. This occurs when either CPUs' Deep Sleep state changes, the SLEEPCON for a domain in Deep Sleep receives a DMA wake-up request, PRSTCTL0[CPU1] in RSTCTL3 changes state, a reset occurs, or either CPU's CTRL[APPLYCFG] is set to capture and apply the software updates to its PDRUNCFG registers. PDRUNCFG or PDSLEEPCFG is captured only for the domain that has such an event. Unless both domains have an event at the same time, the other domain's old capture value is used for aggregation.

As shown in the figure below, the output of the aggregation scheme is readable via the PDCFGSTATUSn register.

For applications where one CPU domain never uses any resources configurable in PD*CFG and so the other CPU domain should have complete control, the first domain should configure that resource to be in the lowest power mode so that aggregation will always use the second domain's PD*CFG to control that resource.

> For example, if the Compute CPU never uses any VDD1 logic, then the Compute slot should be configured with PDRUNCFG0[AFBB1_PD] = 1, PDRUNCFG0[RBB1_PD] = 0, and the same for PDSLEEPCFG, so regardless of the activity in the Compute domain, the Sense domain has complete control of the VDD1 body bias. If the Compute domain does use VDD1 when the Compute CPU is active, then it should configure PDRUNCFG0[AFBB1_PD] = 0, PDRUNCFG0[RBB1_PD] = 1, and configure PDSLEEPCFG0[AFBB1_PD] = 1, PDSLEEPCFG0[RBB1_PD] = 0 so that when the Compute domain goes to Deep Sleep, the Sense domain can put VDD1 into RBB mode.

NOTE

When PRSTCTL0[CPU1] in RSTCTL3 = 1, the aggregated output is just the Compute domain's selected PDRUNCFG or PDSLEEPCFG and the default Sense domain's PDRUNCFG and PDSLEEPCFG are ignored.

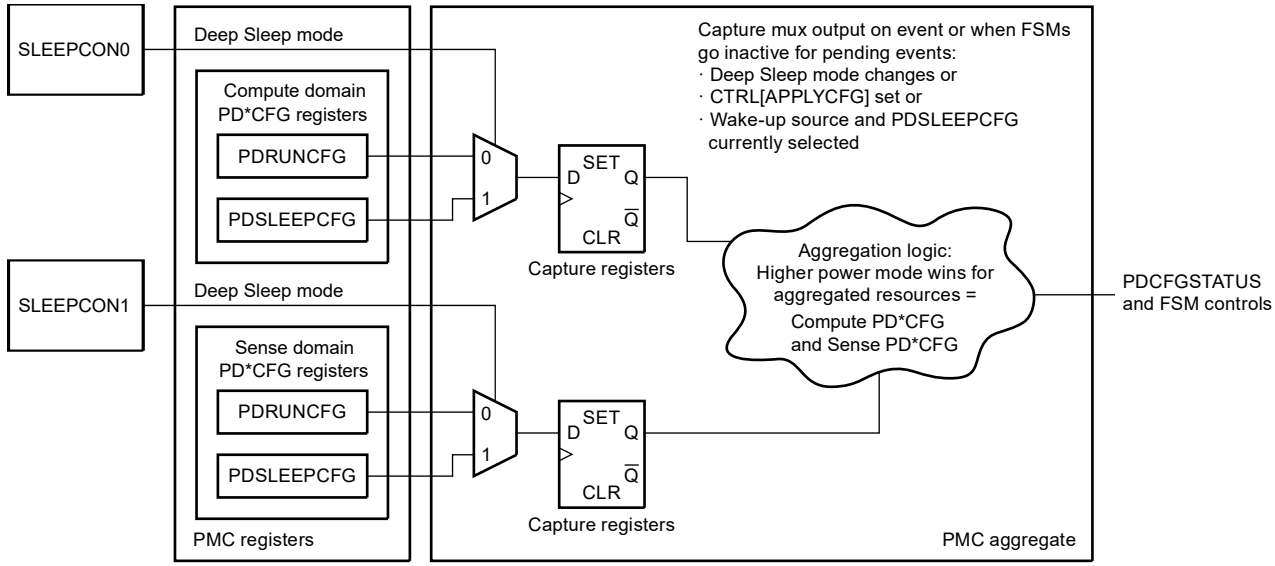


Figure 87. PMC PD*CFG aggregation scheme

27.3.2.3 Control functions

This section describes some of the control functions that may not be obvious from the register descriptions.

27.3.2.3.1 Autowake

This feature allows PMC to wake up some amount of time after the domain enters Deep Sleep. It is enabled for a domain by setting INTRCTRL[AUTOWKIE] and activated when that domain enters Deep Sleep. When PMC has completed the Deep Sleep mode change, the Autowake timer counts down from the value set in WAKEUP[WAKETIME] and when it reaches zero it asserts that domain's PMC interrupt, triggering a wake-up. The timer is clocked by PMC's internal 16 MHz clock. The domain's SLEEPCON and CPU should be configured to wake up and respond to the PMC interrupt. This feature is usable for wake-up from Deep Sleep and FDSR modes.

Autowake has one timer and one time out value, WAKEUP[WAKETIME], for each CPU slot. Both timers can get enabled if both slots go to Deep Sleep mode at the same time, else only 1 will run at a time. New requests blocked until after the Autowake time out. On wake-up via Autowake a pending Deep Sleep entry or wake-up request from the other slot is possible and its PDSLEEPCFG or PDRUNCFG will be aggregated along with the PDRUNCFG of the first slot. If both timers are enabled at the same time, wake-up will occur after BOTH timers time out (i.e. the max timeout value between the two is used).

27.3.2.3.2 Regulator and LVD level control

PD*CFG0[DCDC_VSEL] selects one of two voltage levels for the DCDC regulation voltage and for the VDDN LVD trip level. The actual level values are stored in DCDCVSEL (two values) and LVDVDDNCTRL (two values). Similarly, PD*CFG0[LDO1_VSEL] and PD*CFG0[LDO2_VSEL] select one of the four values stored in LDOVDD1VSEL, LDOVDD2VSEL, LVDVDD1CTRL, and LVDVDD2CTRL. This allows the PDRUNCFG, PDSLEEPCFG, and aggregation logic to select valid LVD and VDD combinations while adjusting the supplies when activity levels change.

27.3.2.3.3 Low-power mode details

27.3.2.3.3.1 Full Deep Sleep Retention (FDSR)

This mode reduces chip's power consumption to the minimum possible level while retaining state. In the default configuration, the DCDC and LDOs are powered down and the VDD1 and VDD2 rails are supplied by SCPC1 and SCPC2. At high temperature, these regulators may not supply sufficient current, so an alternate configuration using the DCDC and the LDOs to power VDD1

and VDD2 can be used. Alternatively, an external PMIC can be used to power VDDN, VDD1, and VDD2 in all modes. In this case, the VDDN supply can be powered down during FDSR mode.

When using the default configuration, the SCPC overcurrent flag triggers when excess load pulls down VDD1 or VDD2. If INTRCTRL[SCP1OCIE] and [SCP2OCIE] are set, then a wake-up and a PMC interrupt occur. The LVDs are masked in this mode because the minimum LVD setting is 0.5 V and the SCPC OC flag serves the same purpose.

At high temperature, the logic leakage can be too high for the SCPC to provide (>400 µA), so this mechanism allows software to go back to FDSR using the LDO + DCDC if desired.

Similarly, the logic leakage can also be too high for DCDC ULP mode to provide at high temperature, in this case, software can use DCDC LP mode.

27.3.2.3.3.2 Full Deep Power Down (FDPD)

Any Always-on wake-up source (the RTC, the PMIC_IRQN pin, or the RESETN pin) will wake up the chip from FDPD mode.

The default configuration keeps the references in high-power mode during FDPD mode so wakeup is triggered when the VDD1V8 supply rises through the VDD1V8 POR trip level.

NOTE

In this case, the VDD1V8 supply must drop below the POR trip level so the POR asserts and then later rise above the trip level so the POR will negate, triggering a wake-up.

To save power when the rising VDD1V8 wake-up feature isn't needed, the references can be put in low-power mode during FDPD by setting POWERCFG[FDPDBGLP]. This causes the references to enter high-power mode only after a wake-up event occurs, but the part will not wake up when VDD1V8 ramps up from below the VDD1V8 POR trip level to above.

In either configuration the wake-up event triggers a VDD1V8 POR cold reset.

27.3.2.4 Always-on domain logic

The Always-on domain logic includes the VDD1V8_AO POR, the PMIC_MODE pin controls, the logic used to trigger wake-up when PMIC_IRQN, RESETN, or RTC wake-up events occur, and the FLAGS fields that are retained through cold resets and VDD1V8 PORs. Access to these functions is accomplished via the register interface, with no special software control needed.

27.3.3 Clocking

PMC is clocked by the internal 16 MHz oscillator. The chip synchronizes all PMC bus accesses to this clock. All counters and state machine timing in PMC are clocked by the 16 MHz oscillator, except for the cyclic wakeup timer to the DCDC which is clocked by a 64 kHz oscillator.

27.3.4 Reset

PMC both generates and responds to a variety of reset sources.

27.3.4.1 PMC generated resets

PMC generates resets from the voltage monitors (POR, LVD, HVD, aGDET) which detect under or overvoltage conditions on the supply rails they monitor as described in Supply voltage monitors.

Table 229. Reset sources and effects

Source	Detector power mode	Enabled	Reset type	Description
POR VDD1V8_AO	Always on	Always	Cold VDD1V8_AO	All chip's logic is reset, including PMC flags.

Table continues on the next page...

Table 229. Reset sources and effects (continued)

Source	Detector power mode	Enabled	Reset type	Description
POR VDD1V8	On when VDD1V8 is powered	Always	Cold VDD1V8	Triggers when VDD1V8 drops below the VDD1V8 POR falling trip level. Resets all core logic and most PMC registers except some FLAGS and PWRFLAGS. **NOTE** When in FDPD, the VDD1V8 POR can be on to use as a wake-up source when VDD1V8 ramps back up, or can be disabled to save power if VDD1V8 may not drop below the POR threshold and other wake-up sources are used. This feature is controlled by POWERCFG[FDPDBGLP].
POR VDD1	PD*CFG1[POR1_LP]	Always	Cold	Triggers when VDD1 drops below the POR level set in PORCTRL register. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
POR VDD2	PD*CFG1[POR2_LP]	Always	Cold	Triggers when VDD2 drops below the POR level set in PORCTRL register. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
POR VDDN	PD*CFG1[PORN_LP]	Always	Cold	Triggers when VDDN drops below the POR level set in PORCTRL register. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
LVD VDD1	PD*CFG1[LVD1_LP]	CTRL[LVD1RE]	Cold	Triggers when VDD1 drops below the LVD level selected by the aggregated PD*CFG0[LDO1_VSEL] and defined in LDOVDD1VSEL register. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
LVD VDD2	PD*CFG1[LVD2_LP]	CTRL[LVD2RE]	Cold	Triggers when VDD2 drops below the LVD level selected by the aggregated PD*CFG0[LDO2_VSEL] and defined in LDOVDD2VSEL register. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
LVD VDDN	PD*CFG1[LVDN_LP]	CTRL[LVDNRE]	Cold	Triggers when VDDN drops below the LVD level selected by the aggregated PD*CFG0[DCDC_VSEL] and defined in LVDVDDNCTRL register. Resets all core logic

Table continues on the next page...

Table 229. Reset sources and effects (continued)

Source	Detector power mode	Enabled	Reset type	Description
				and most PMC registers except FLAGS, PWRFLAGS.
HVD VDD1V8	PD*CFG1[HVD1V8_PD]	CTRL[HVD1V8RE]	Cold	Triggers when VDD1V8 rises above the HVD level for VDD1V8. Resets all core logic and most PMC registers except FLAGS, PWRFLAGS.
HVD VDD1	PD*CFG1[HVD1_PD]	CTRL[HVDV1RE]	Cold	Triggers when VDD1 rises above the HVD level for VDD1. Resets all core logic and most PMC registers except some FLAGS and PWRFLAGS.
HVD VDD2	PD*CFG1[HVD2_PD]	CTRL[HVDV2RE]	Cold	Triggers when VDD2 rises above the HVD level for VDD2. Resets all core logic and most PMC registers except some FLAGS and PWRFLAGS.
HVD VDDN	PD*CFG1[HVDN_PD]	CTRL[HVDVNRE]	Cold	Triggers when VDDN rises above the HVD level for VDDN. Resets all core logic and most PMC registers except some FLAGS and PWRFLAGS.
aGDET VDD1	PD*CFG1[AGDET1_PD]	CTRL[AGDET1RE]	Cold	Triggers when a negative spike on VDD1 or positive spike on VSS exceeds the glitch detection threshold for VDD1
aGDET VDD2	PD*CFG1[AGDET2_PD]	CTRL[AGDET2RE]	Cold	Triggers when a negative spike on VDD2 or positive spike on VSS exceeds the glitch detection threshold for VDD2

27.3.4.2 PMC reset inputs

PMC receives a warm reset from the chip. Some PD*CFG bit fields, the voltage selection registers, DCDCVSEL, LDOVDDnVSEL, LVDVDDnCTRL, BBCTRL, TSENSOR, WAKEUP, and PDWAKECFG are reset, PMC transitions to Control Active/Sense Reset mode, so that PMC is correctly configured for the chip to boot up.

NOTE

The power switches for logic and memory are only reset by cold resets to maintain state to allow debugging of warm reset events.

The RESETN pin is intercepted by PMC for use as a wake-up source, and to record the pin reset event in FLAGS[RESETNF], but the chip asserts the warm reset input to PMC.

27.3.4.3 RESETN bidirectional pin control

PMC activates the RESETN strong pull down driver when any chip reset occurs. It asserts RESETN until PMC has completed its transition to Compute Active, Sense Reset mode, then releases the pull down and allows the RESETN weak pull up or any external driver in the application to control RESETN.

27.3.5 Interrupts

PMC has 2 interrupts, one for the Compute domain CPU and one for the Sense domain CPU, that are available to the CPUs' NVICs and SLEEPCONs. The sources for these interrupts include the PMIC_IRQN pin, the Autowake feature, Compute CPU Deep Sleep entry, Sense CPU Deep Sleep entry, the voltage monitors, the SCPC and DCDC overcurrent warnings, and the body bias blocks' error warnings. Each source for the two interrupts are enabled by each domain's INTRCTRL register. Only events that have a corresponding INTRCTRL enable can generate interrupts.

In Deep Sleep and FDSR modes, PMC interrupt for each slot will only wake up the CPU that has the source enabled in its slot's INTRCTRL. That domain's input to the aggregation logic will switch to its PDRUNCFG. The other domain's input to the aggregation logic remains unchanged. In DPD and FDPD modes the PMC interrupt event will trigger a cold reset, thus waking up and resetting both domains.

An event on any of the sources also asserts the corresponding FLAGS, whether the source is enabled in INTRCTRL or not. The FLAGS are also implemented as two registers, one for each CPU domain. They are "sticky" in that they are set by the event, but are cleared by software.

There are other flags in FLAGS and PWRFLAGS that are not related to a PMC interrupt source to provide the application with information about previous states for use after a wake-up or reset events.

The STATUS registers show the current state of the CPUs' Deep Sleep mode, PMIC_IRQN pin, RTC wake-up outputs, voltage monitors, and if the PMC sequencing state machines are BUSY.

The following table shows the events and sources that have status and flags that software can interrogate or enable for reset or interrupts. See the associated block descriptions and registers for more information.

Table 230. Event, status, flag, interrupt, and reset sources and controls

Source or event description	Wake-up or reset source for modes	Registers				
		STATUS	FLAGS	PWRFLAGS	CTRL (reset enable)	INTRCTRL (interrupt enable)
Chip entered Deep Power Down mode	-	-	[DEEPPDF]	-	-	-
Compute domain entered Deep Sleep	Sense Deep Sleep	[DSCOMP]	[DSCOMPF]	-	-	[DSCOMPIE]
Sense domain entered Deep Sleep	Compute Deep Sleep	[DSSENS]	[DSSENSF]	-	-	[DSSENSIE]
RESETN pin asserted	All modes	-	[RESETNF]	-	-	-
PMIC_IRQN pin asserted	All modes	[INTN]	[INTNF]	-	-	[INTNIE]
Autowake occurred for this domain	Deep Sleep, FDSR	-	[AUTOWKF]	-	-	[AUTOWKIE]
RTC wake or alarm received for this domain	All modes	[RTCWKUP]	[RTCF]	-	-	-
VDD1V8 HVD triggered	DPD, FDSR, Deep Sleep, Active	[HVD1V8]	[HVD1V8F]	-	[HVD1V8RE]	[HVD1V8IE]
VDDN HVD triggered	FDSR, Deep Sleep, Active	[HVDVDDN]	[HVDVDDNF]	-	[HVDVNRE]	[HVDNIE]
VDD2 HVD triggered	FDSR, Deep Sleep, Active	[HVDVDD2]	[HVDVDD2F]	-	[HVDV2RE]	[HVD2IE]
VDD1 HVD triggered	FDSR, Deep Sleep, Active	[HVDVDD1]	[HVDVDD1F]	-	[HVDV1RE]	[HVD1IE]

Table continues on the next page...

Table 230. Event, status, flag, interrupt, and reset sources and controls (continued)

Source or event description	Wake-up or reset source for modes	Registers				
		STATUS	FLAGS	PWRFLAGS	CTRL (reset enable)	INTRCTRL (interrupt enable)
VDDN LVD triggered	FDSR, Deep Sleep, Active	[LVDVDDN]	[LVDVDDNF]	-	[LVDNRE]	[LVDNIE]
VDD2 LVD triggered	FDSR, Deep Sleep, Active	[LVDVDD2]	[LVDVDD2F]	-	[LVD2RE]	[LVD2IE]
VDD1 LVD triggered	FDSR, Deep Sleep, Active	[LVDVDD1]	[LVDVDD1F]	-	[LVD1RE]	[LVD1IE]
VDD2 aGDET triggered	Deep Sleep, Active	-	[AGDET2F]	-	[AGDET2RE]	[AGDET2IE]
VDD1 aGDET triggered	Deep Sleep, Active	-	[AGDET1F]	-	[AGDET1RE]	[AGDET1IE]
VDD1V8_AO POR reset occurred	All modes	-	[PORAO18F]	-	-	-
VDD1V8 POR reset occurred	All modes[1]	-	[POR1V8F]	-	-	-
VDDN POR reset occurred	FDSR, Deep Sleep, Active	-	[PORVDDNF]	-	-	-
VDD2 POR reset occurred	FDSR, Deep Sleep, Active	-	[PORVDD2F]	-	-	-
VDD1 POR reset occurred	FDSR, Deep Sleep, Active	-	[PORVDD1F]	-	-	-
DCDC low-power mode overcurrent warning asserted	FDSR, Deep Sleep, Active	-	[DCLPOCF]	-	-	[DCLPOCIE]
DCDC high-power mode overcurrent warning asserted	FDSR, Deep Sleep, Active	-	[DCHPOCF]	-	-	[DCHPOCIE]
SCPC2 overcurrent warning asserted	FDSR, Deep Sleep, Active	-	[SCP2OCF]	-	-	[SCP2OCIE]
SCPC1 overcurrent warning asserted	FDSR, Deep Sleep, Active	-	[SCP1OCF]	-	-	[SCP1OCIE]

Table continues on the next page...

Table 230. Event, status, flag, interrupt, and reset sources and controls (continued)

Source or event description	Wake-up or reset source for modes	Registers				
		STATUS	FLAGS	PWRFLAGS	CTRL (reset enable)	INTRCTRL (interrupt enable)
VDDN Body Bias error occurred	FDSR, Deep Sleep, Active	-	[BBNEF]	-	-	[BBNEIE]
VDD2 Body Bias error occurred	FDSR, Deep Sleep, Active	-	[BB2EF]	-	-	[BB2EIE]
VDD1 Body Bias error occurred	FDSR, Deep Sleep, Active	-	[BB1EF]	-	-	[BB1EIE]
VDD2 SRAM Body Bias error occurred	FDSR, Deep Sleep, Active	-	[BBSR2EF]	-	-	[BBSR2EIE]
VDD1 SRAM Body Bias error occurred	FDSR, Deep Sleep, Active	-	[BBSR1EF]	-	-	[BBSR1EIE]
Intrusion and Tamper Response Controller reset occurred	None	-	[ITRCRF]	-	-	-
PMC state machine trigger control and status	None	[BUSY]	-	-	[APPLYCFG]	-
Both Compute and Sense domains were in Deep Sleep Mode at the same time.	None	-	[DSALLF]	-	-	-
VDD2_OTP was powered down	None	-	-	[V2OTPF]	-	-
VDD2_MIPI was powered down	None	-	-	[V2MIPIF]	-	-
VDD2_DSP was powered down	None	-	-	[V2DSPF]	-	-
VDDN_COM was in DSR	None	-	-	[VNCOMF]	-	-
VDD2_COM was in DSR	None	-	-	[V2COMF]	-	-
VDD2_MEDIA and VDDN_MEDIA were in DSR	None	-	-	[V2NMEDF]	-	-
VDD2_COMP was in DSR	None	-	-	[V2COMPF]	-	-
VDD1_SENSE was in DSR, which means FDSR mode was entered.	None	-	-	[V1SENSF]	-	-

1. VDD1V8 POR can be used to wake up from FDPD mode only if POWERCFG[FDPDBGLP] = 0.

27.4 External signals

PMC uses the following external signals

Table 231. PMC external signals

Signal	Direction	Description
RESETN	Open drain IO	Reset pin, active low. Used as a wake-up source for all power modes. Used by the chip to generate a chip wide warm reset. RESETN assertions set FLAGS[RESETNF]. PMC will assert the RESETN pin low on any reset event and hold it until PMC has completed transition to the Compute Active / Sense Reset state.
PMIC_IRQN	Input	Interrupt pin, active low. Used as a wake-up source for all power modes when enabled by INTRCTRL[INTNIE]. Triggers an interrupt for the CPU
PMIC_MODE[1:0]	Output	PMIC mode control pins used to activate PMIC power mode changes (e.g. supply voltage changes, external regulator power mode changes, etc.) when the aggregated value of PD*CFG0[PMICMODE] changes due to a software write to PDRUNCFG and CTRL[APPLYCFG] or either CPU entering or exiting Deep Sleep, or a wake-up or reset event.
PRSTCTL0[CPU1] in RSTCTL3	Input	Indicates that the Sense domain CPU is held in reset, causing the aggregation logic to ignore the Sense PD*CFG. When this input negates PMC triggers a mode change to update the PMC state to incorporate the aggregated Compute and Sense domain PD*CFG registers.
RTC wake-up	Input	The interrupt outputs from the RTC for the Compute and Sense domains. Used by PMC to trigger an interrupt when the associated INTRCTRL bit is set to wake up from power modes where the core logic is not available (FDSR, DPD, FDPD), allowing PMC to apply power to the core logic. When asserted, the event is captured in FLAGS. When the INTRCTRL enable bit is set and the RTC alarm or wakeup asserts, the PMC interrupt will remain asserted until the core clears the RTCF flag or the INTRCTRL RTCIE bit.

27.5 Initialization

This module comes up in a valid state and does not require initialization.

27.6 Application information

This section provides some information about how applications should use some of PMC features.

27.6.1 Enabling body bias for low-power modes

To enable RBB during low-power modes for the voltage domains (VDD1, VDD2, VDDN) that have their clocks stopped:

1. Enable the desired mode in PDSLEEPCFG:
   - RBB: clear PDSLEEPCFG0[RBBn_PD]] and set PDSLEEPCFG0[AFBBn_PD]]

   NOTE

   Software is responsible for ensuring the clocks are stopped during Deep Sleep. PMC will activate RBB regardless of the state of the clocks.

2. If only one CPU uses a voltage domain while that CPU is Active, then set the other CPU's PDRUNCFG0[AFBBn_PD] so that AFBB is disabled when the first CPU is in Deep Sleep mode, otherwise aggregation will keep AFBB on even though the first CPU doesn't use that voltage domain.

3. Enter Deep Sleep mode

27.6.2 Using a PMIC

PMC supports using an external PMIC to power VDD1V8, VDDN, VDD2, and VDD1, in a few different configurations. The external PMIC can be as simple as a single external 1.8 V regulator to a complex PMIC with multiple supplies that can be independently adjusted.

Support includes:

- PMIC_MODE pins to control the PMIC configuration automatically during mode changes, no I2C needed.
- [MODEDLY] timer to select how long to wait for a higher power PMIC configuration to take effect after PMIC_MODE changes state.
- PMIC_IRQN pin can be used by the PMIC (or any other external control) to interrupt the device during any power mode, including FDPD mode, triggering a wake-up and an interrupt to either or both CPUs.
- LVD handshaking holds off operation until the PMIC supplied voltages reach a new higher value during a mode change.
- Internal regulator power down controls in POWERCFG to disable the internal regulator for rails that are powered externally.

When using a PMIC to supply the internal logic rails, the PMIC should be configured at startup or cold reset to supply VDDN at 1.1 V nominal, and VDD1 and VDD2 at 1.0 V nominal.

27.6.2.1 Supported configurations

PMC supports these configurations for externally supplied voltages. There is a lot of different ways to use these configurations based on the capability of the PMIC.

27.6.2.1.1 Internal core regulators

This is the default configuration. The PMC reset values are sufficient.

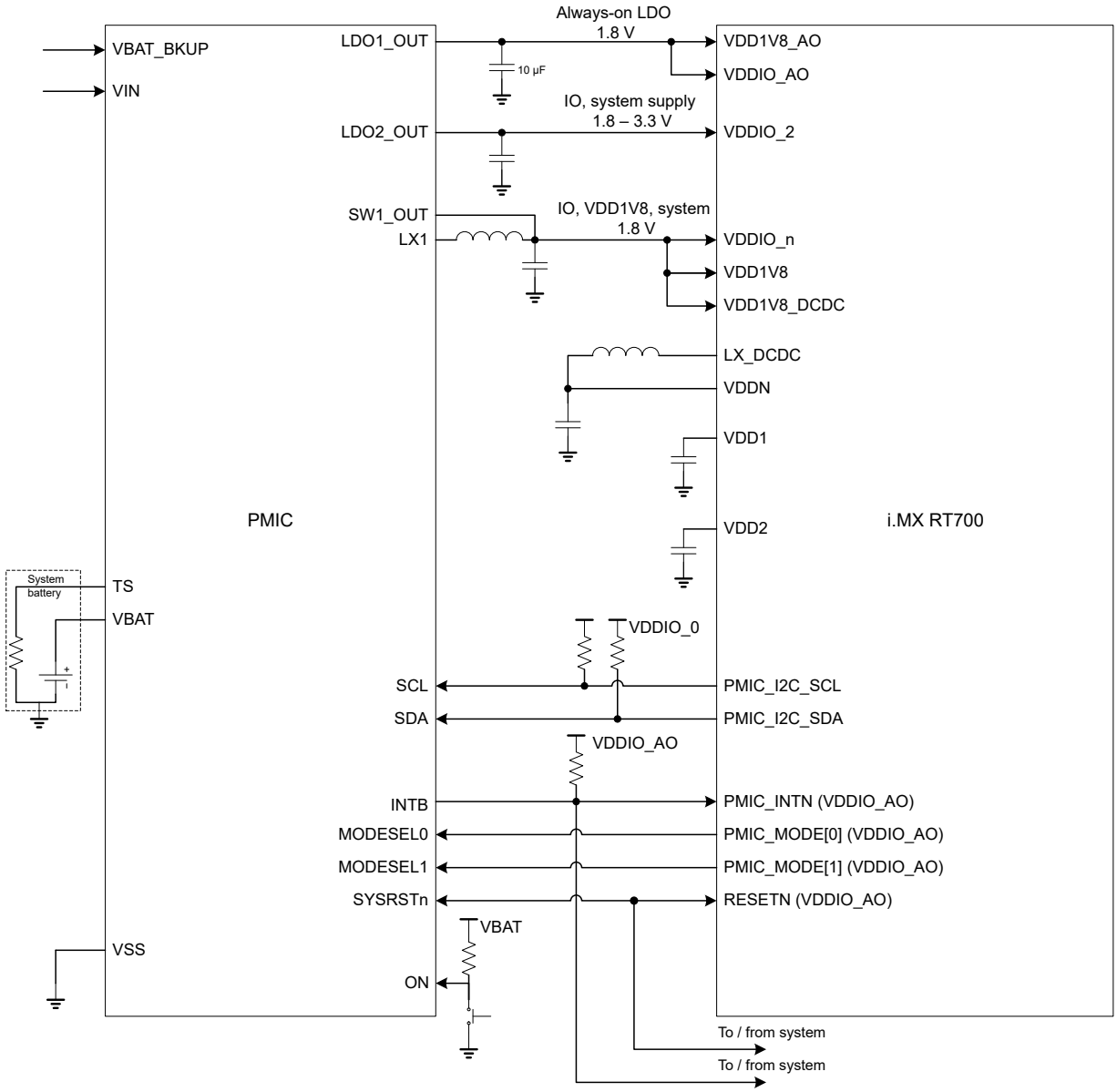


Figure 88. PMIC for 1.8 V and 3 V supplies only

27.6.2.1.2 External VDDN, internal VDD1 and VDD2

This is a slight variation from the default configuration, using the PMIC to supply VDDN while the internal LDOs power VDD1 and VDD2. The PMIC must ensure that VDDN ramps up to a voltage higher than the default VDDN LVD trip level, and provides enough head room to the VDD1 and VDD2 LDOs so they can ramp up correctly. PMIC_MODE can be used to change the voltage level of VDDN or power it down as needed. VDDN can be powered down in FDSR, DPD, and FDPD modes. The application must set [MODEDLY] so that VDDN ramps up to at least 1.0 V before the LDOs turn on when waking from FDSR mode. PMC will wait for VDDN to rise above the aggregated PD*CFG0[DCDC_VSEL] LVD trip before turning on the LDOs.

PMC should be configured to disable the DCDC after every cold reset by setting POWERCFG[DCDCPD].

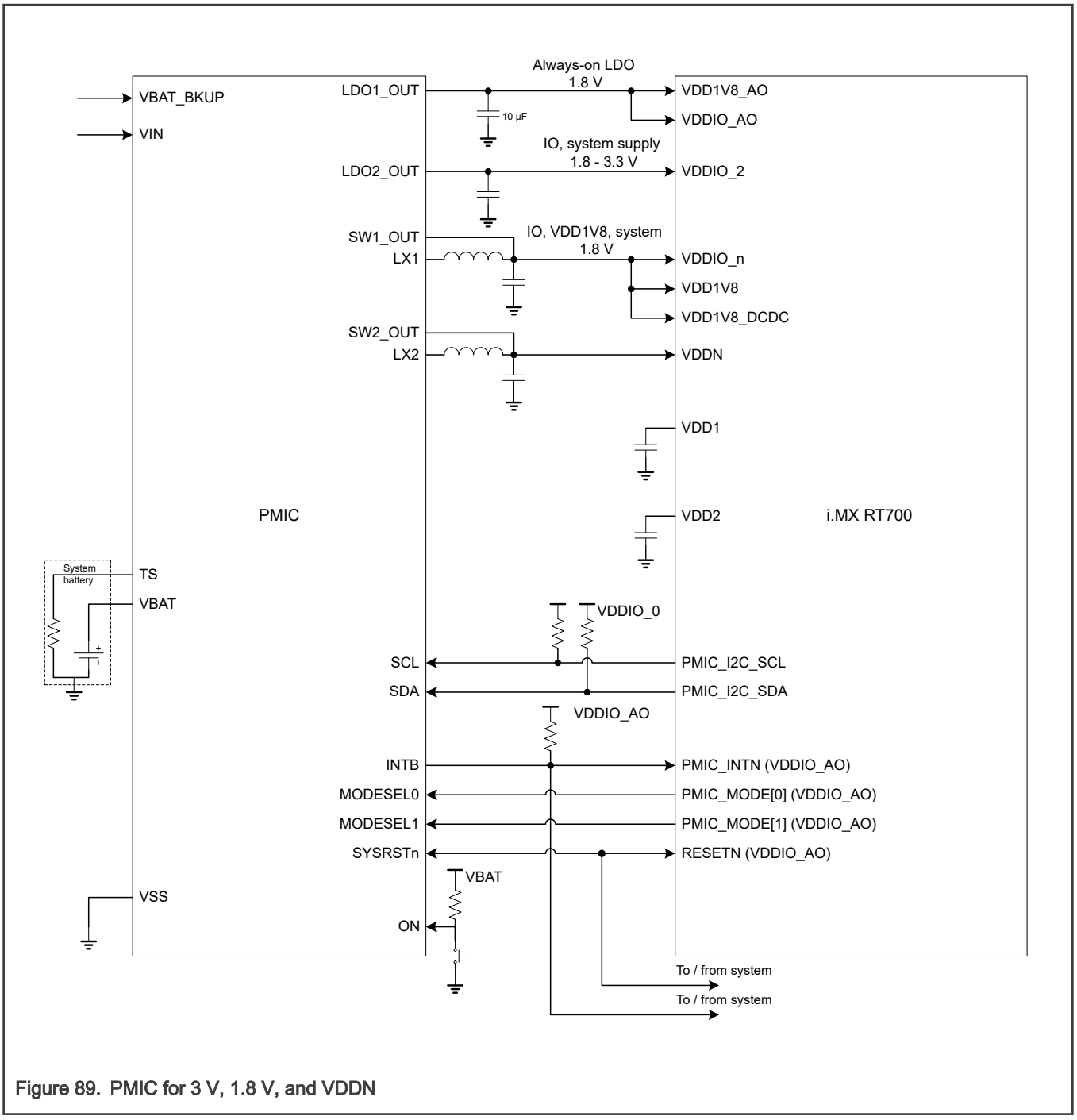


Figure 89. PMIC for 3 V, 1.8 V, and VDDN

27.6.2.1.3 External VDDN, VDD1, VDD2 with VDD1=VDD2

This configuration uses only external supplies, using the PMIC to supply VDDN, VDD1, and VDD2. The PMIC must ensure that VDDN, VDD1, and VDD2 ramp up to voltages between the minimum and maximum levels defined by the boot configuration. PMIC_MODE can be used to change the voltage level of VDDN, VDD1, VDD2, or power them down as needed. VDDN can be powered down in FDSR mode. VDD1, VDD2, and VDDN can all be powered down in DPD and FDPD modes. The application should set [MODEDLY] to allow for the supplies to ramp back up after PMIC_MODE changes. PMC will wait for VDDN, VDD1, and VDD2 to rise above the trip points selected by PD*CFG0[DCDC_VSEL], PD*CFG0[LDO1_VSEL], and PD*CFG0[LDO2_VSEL] before waking up the rest of the chip.

PMC should be configured to disable the DCDC and LDOs after every cold reset by setting POWERCFG[DCDCPD], POWERCFG[LDO1PD], and POWERCFG[LDO2PD].

If the SCPCs are used in FDSR mode, the PMIC should be configured to supply 0.5 V to VDD1 and VDD2 during FDSR mode because the on-board SCPCs may not be able to overdrive the connections on VDD1 and VDD2. Set POWERCFG[SCPCPD] to prevent the SCPCs from turning on in FDSR.

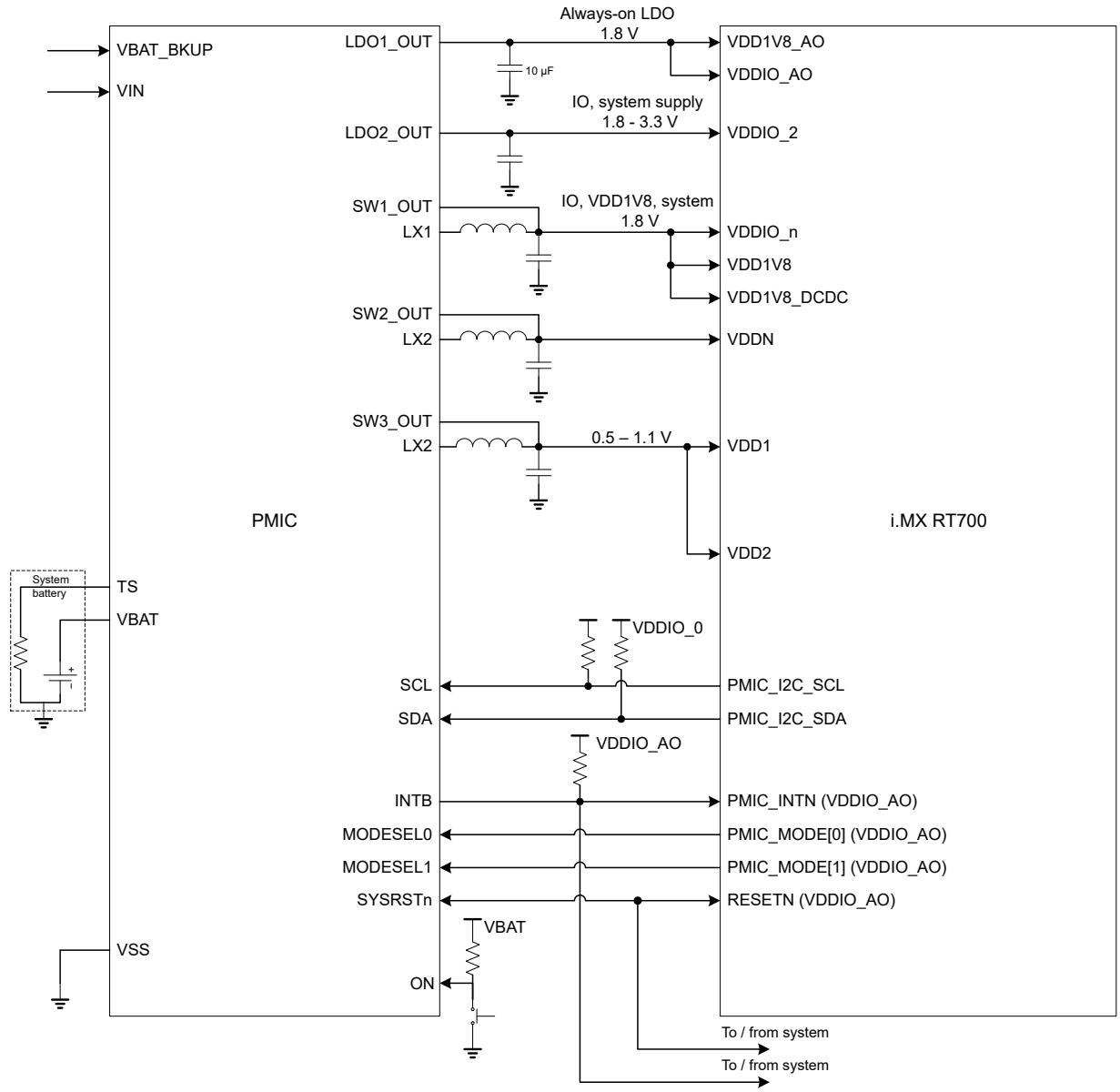


Figure 90. PMIC for 3.3 V, 1.8 V, VDDN, and shred for VDD1 and VDD2

27.6.2.1.4 External VDDN, VDD1, VDD2

This configuration uses only external supplies, using the PMIC to supply VDDN, VDD1, and VDD2, but provides more opportunities to reduce power when parts of the chip are active by decoupling the voltages supplied to VDD1 and VDD2, allowing one or the other to be reduced when activity in that domain is low. The PMIC must ensure that VDDN, VDD1, and VDD2 ramp up to

voltages between the minimum and maximum levels defined by the boot configuration. PMIC_MODE can be used to change the voltage level of VDDN, VDD1, VDD2, or power them down as needed. VDDN can be powered down in FDSR mode. VDD1, VDD2, and VDDN can all be powered down in DPD and FDPD modes. The application should set [MODEDLY] to allow for the supplies to ramp back up after PMIC_MODE changes. PMC will wait for VDDN, VDD1, and VDD2 to rise above the trip points selected by PD*CFG0[DCDC_VSEL], PD*CFG0[LDO1_VSEL], and PD*CFG0[LDO2_VSEL] before waking up the rest of the chip.

PMC should be configured to disable the DCDC and LDOs after every cold reset by setting POWERCFG[DCDCPD], POWERCFG[LDO1PD], and POWERCFG[LDO2PD].

If the SCPCs are used in FDSR mode, the PMIC should be configured to supply 0.5 V to VDD1 and VDD2 during FDSR mode because the on-board SCPCs may not be able to overdrive the connections on VDD1 and VDD2. Set POWERCFG[SCPCPD] to prevent the SCPCs from turning on in FDSR.

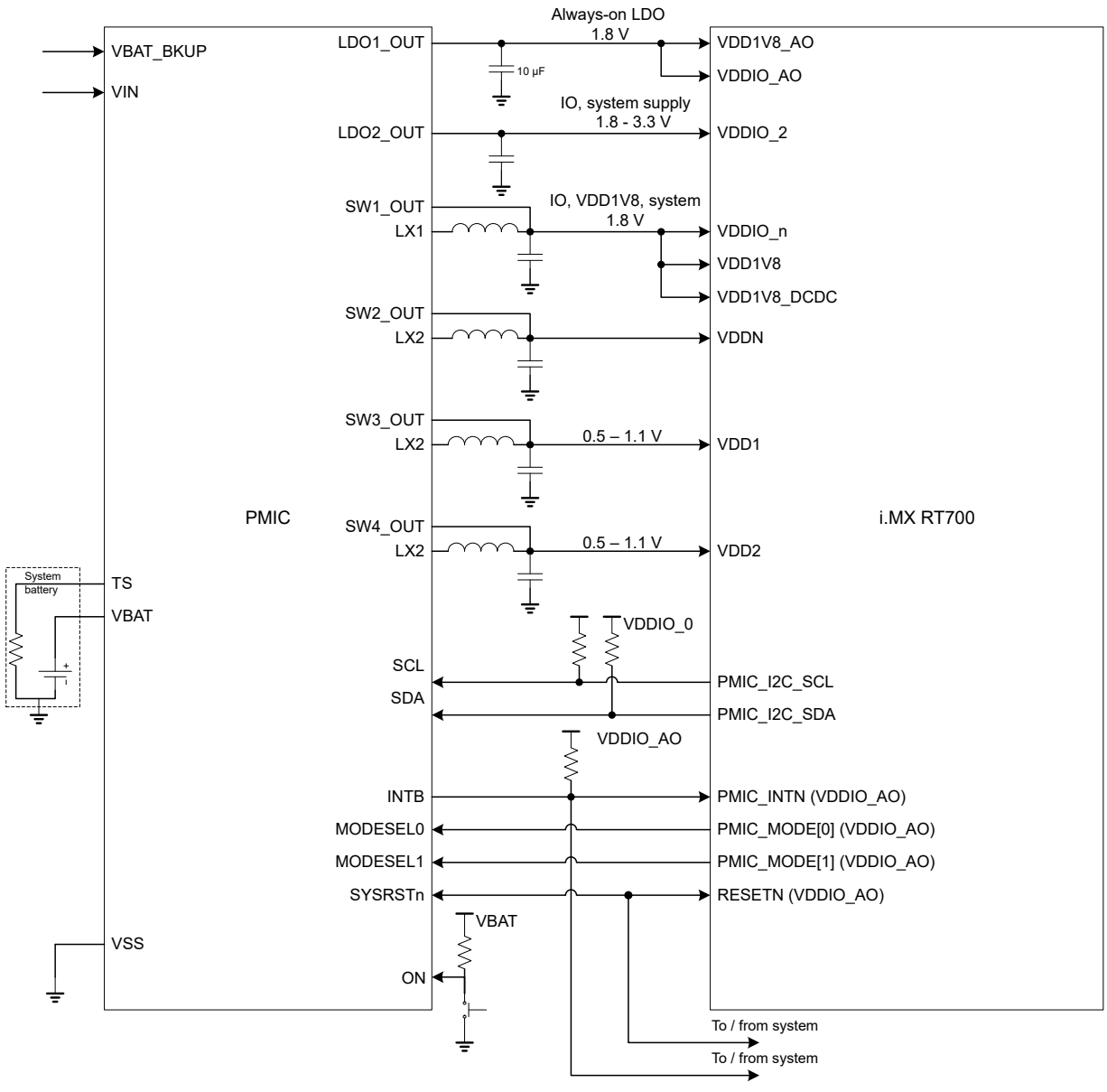


Figure 91. PMIC for 3.3 V, 1.8 V, VDDN, VDD2, and VDD1

27.7 Memory map and register definitions

PMC has 2 sets of registers, one is for the Compute domain, the other is for the Sense domain. Each domain accesses its set of registers via a separately mapped range of addresses, also referred to as the domain slot. Some registers and fields are shared allowing read and write access from either domain (e.g. LDOVDD1VSEL). Others are readable by both domains but writable by only one domain (e.g. LDOVDD2VSEL). Others are separate storage elements, accessible only via that slot (e.g. PDRUNCFG0). See individual register descriptions for more information.

In this chapter, the register descriptions are divided into two sections, one for each slot: ***PMC0*** for Compute domain and ***PMC1*** for Sense domain.

27.7.1 PMC register descriptions

27.7.1.1 PMC register description memory map

This section includes the memory map and detailed description of all registers and their fields of PMC0 in Compute domain.

PMC.PMC0 base address: 4020_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PMC Version and Feature ID (VERID)	32	R	0000_0000h
4h	PMC Status (STATUS)	32	R	0000_0000h
8h	PMC Flags in Compute Domain (FLAGS)	32	RW	001C_F800h
Ch	Control (CTRL)	32	RW	001C_0000h
10h	Interrupt Control in Compute Domain (INTRCTRL)	32	RW	0000_0000h
20h	DCDC Voltage Selection (DCDCVSEL)	32	RW	0000_6050h
24h	LDO VDD2 Voltage Selection (LDOVDD2VSEL)	32	RW	2C24_1C14h
28h	LDO VDD1 Voltage Selection (LDOVDD1VSEL)	32	RW	2C24_1C14h
2Ch	VDDN LVD Control (LVDVDDNCTRL)	32	RW	0000_2C28h
30h	VDD2 LVD Control (LVDVDD2CTRL)	32	RW	281E_140Ah
34h	VDD1 LVD Control (LVDVDD1CTRL)	32	RW	281E_140Ah
38h	Power-on Reset Control (PORCTRL)	32	RW	0008_0808h
40h	Body Bias Control (BBCTRL)	32	RW	0001_0101h
44h	Temperature Sensor Control (TSENSOR)	32	RW	0000_0000h
50h	Wake-up Configuration for Compute Domain (WAKEUP)	32	RW	0000_0000h
64h	Power Configuration (POWERCFG)	32	RW	6000_0070h
68h	3 V Capable VDDIO Range Select (PADVRANGE)	32	RW	0000_0000h
6Ch	IO Bank State Configuration (PADCFG)	32	RW	0000_0000h
A0h	PD Run Configuration 0 in Compute Domain (PDRUNCFG0)	32	RW	F577_6740h
A4h	PD Run Configuration 1 in Compute Domain (PDRUNCFG1)	32	RW	7FFE_4003h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
A8h	PD Run Configuration 2 in Compute Domain (PDRUNCFG2)	32	RW	FFFF_FFFEh
ACh	PD Run Configuration 3 in Compute Domain (PDRUNCFG3)	32	RW	FFFF_FFFEh
B0h	PD Run Configuration 4 in Compute Domain (PDRUNCFG4)	32	RW	FFBF_F1FFh
B4h	PD Run Configuration 5 in Compute Domain (PDRUNCFG5)	32	RW	FFBF_F1FFh
B8h	PD Sleep Configuration 0 in Compute Domain (PDSLEEPCFG0)	32	RW	F577_6740h
BCh	PD Sleep Configuration 1 in Compute Domain (PDSLEEPCFG1)	32	RW	7FFE_4003h
C0h	PD Sleep Configuration 2 in Compute Domain (PDSLEEPCFG2)	32	RW	FFFF_FFFEh
C4h	PD Sleep Configuration 3 in Compute Domain (PDSLEEPCFG3)	32	RW	FFFF_FFFEh
C8h	PD Sleep Configuration 4 in Compute Domain (PDSLEEPCFG4)	32	RW	FFBF_F1FFh
CCh	PD Sleep Configuration 5 in Compute Domain (PDSLEEPCFG5)	32	RW	FFBF_F1FFh
D0h	PD Configuration Status 0 (PDCFGSTATUS0)	32	R	F577_6740h
D4h	PD Configuration Status 1 (PDCFGSTATUS1)	32	R	7FFE_4003h
D8h	PD Configuration Status 2 (PDCFGSTATUS2)	32	R	FFFF_FFFEh
DCh	PD Configuration Status 3 (PDCFGSTATUS3)	32	R	FFFF_FFFEh
E0h	PD Configuration Status 4 (PDCFGSTATUS4)	32	R	FFBF_F1FFh
E4h	PD Configuration Status 5 (PDCFGSTATUS5)	32	R	FFBF_F1FFh
E8h	PD Wake-up Configuration for Compute Domain (PDWAKECFG)	32	RW	0000_0000h
ECh	Power Domain Flags for Compute Domain (PWRFLAGS)	32	RW	0000_0074h

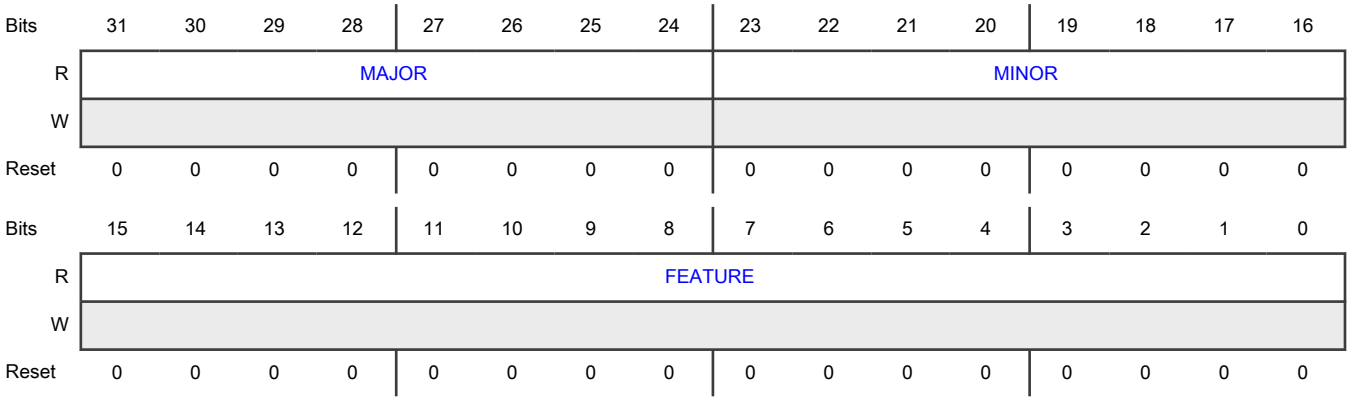
27.7.1.2 PMC Version and Feature ID (VERID)

Offset

Register	Offset
VERID	0h

Function
Indicates the version integrated for this instance on the chip.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Reads the major version number for the module specification.
23-16 MINOR	Minor Version Number Reads the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Reads the feature set number.

27.7.1.3 PMC Status (STATUS)

Offset

Register	Offset
STATUS	4h

Function

Indicates current status of PMC sources, not latched, but synchronized to PMC clock. See PMC Flags for latched versions of sources that are enabled to generate resets or interrupts.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	DSSENS	0	INTN	0	RTCWKUP	HVD1V8	HVDVDDN	HVDVDD2	HVDVDD1	LVDVDDN	LVDVDD2	LVDVDD1	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															BUSY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 DSSENS	CPU1 Deep Sleep Indicates that CPU1 is in Deep Sleep mode. 0b - CPU1 is in Active mode 1b - CPU1 is in Deep Sleep mode
28 —	Reserved
27 INTN	PMIC_IRQN Interrupt Input Shows the interrupt input pin status. 0b - Negated. Pin is high 1b - Asserted. Pin is low
26 —	Reserved
25 RTCWKUP	RTC Wake-up Shows RTC wake-up status. 0b - Compute RTC wake-up is not active 1b - Compute RTC wake-up is asserted
24	VDD1V8 High-Voltage Detector

Table continues on the next page...

Table continued from the previous page...

Field	Function
HVD1V8	Shows VDD1V8 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
23 HVDVDDN	VDDN High-Voltage Detector Shows VDDN high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
22 HVDVDD2	VDD2 High-Voltage Detector Shows VDD2 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
21 HVDVDD1	VDD1 High-Voltage Detector Shows VDD1 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
20 LVDVDDN	VDDN Low-Voltage Detector Shows VDDN low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
19 LVDVDD2	VDD2 Low-Voltage Detector Shows VDD2 low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
18 LVDVDD1	VDD1 Low-Voltage Detector Shows VDD1 low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
17-1 —	Reserved
0 BUSY	Finite State Machine Status Shows if PMC is currently processing a mode transition.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - PMC state machines are idle 1b - PMC state machine is busy

27.7.1.4 PMC Flags in Compute Domain (FLAGS)

Offset

Register	Offset
FLAGS	8h

Function

Indicates PMC event flags in Compute domain. This register is cleared by software, and set by hardware when event occurs. The flags in Sense and Compute domains are cleared respectively by writing 1 to corresponding bit fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEEPPDF	DSCOMPF	DSSENSF	RESETNF	INTNF	AUTOWKF	RTCF	HVD1V8F	HVDVDDNF	HVDVDD2F	HVDVDD1F	LVDVDDNF	LVDVDD2F	LVDVDD1F	AGDET2F	AGDET1F
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PORAO18F	POR1V8F	PORVDDNF	PORVDD2F	PORVDD1F	DCLPOCF	DCHPOCF	SCP2OCF	SCP1OCF	BBNEF	BB2EF	BB1EF	BBSR2EF	BBSR1EF	ITRCRF	DSALLF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DEEPPDF	DPD Wake-up Flag Indicates the wake-up status of PMC in DPD (Deep Power Down) mode. This bit is reset by VDD1V8_AO POR. 0b - DPD wake-up cleared 1b - DPD wake-up asserted
30 DSCOMPF	CPU0 Deep Sleep Flag Indicates that CPU0 entered Deep Sleep mode. This bit is set at Deep Sleep mode entry.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU0 Deep Sleep cleared 1b - CPU0 Deep Sleep asserted
29 DSSENSF	CPU1 Deep Sleep Mode Flag Indicates that CPU1 entered Deep Sleep mode. This bit is set at Deep Sleep mode entry, and can be cleared while CPU1 is still in Deep Sleep mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU1 Deep Sleep cleared 1b - CPU1 Deep Sleep asserted
28 RESETNF	Reset Pin Flag Indicates reset pin status. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Reset pin status cleared 1b - Reset pin status asserted
27 INTNF	PMIC_IRQN Interrupt Pin Flag Indicates interrupt pin status. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Interrupt pin status cleared 1b - Interrupt pin status asserted
26 AUTOWKF	Auto Wake-up Flag Indicates auto wake-up interrupt status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Auto wake-up status cleared 1b - Auto wake-up status asserted
25 RTCF	RTC Flag Indicates RTC alarm or wake was asserted. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - RTC status cleared 1b - RTC status asserted
24	VDD1V8 High-Voltage Detector Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
HVD1V8F	Indicates VDD1V8 high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
23 HVDVDDNF	VDDN High-Voltage Detector Flag Indicates VDDN high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
22 HVDVDD2F	VDD2 High-Voltage Detector Flag Indicates VDD2 high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
21 HVDVDD1F	VDD1 High-Voltage Detector Flag Indicates VDD1 high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
20 LVDVDDNF	VDDN Low-Voltage Detector Flag Indicates VDDN low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
19 LVDVDD2F	VDD2 Low-Voltage Detector Flag Indicates VDD2 low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
18 LVDVDD1F	VDD1 Low-Voltage Detector Flag Indicates VDD1 low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
17 AGDET2F	VDD2 Glitch Detector Flag Indicates VDD2 glitch detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Glitch detector status cleared 1b - Glitch detector status asserted
16 AGDET1F	VDD1 Glitch Detector Flag Indicates VDD1 glitch detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Glitch detector status cleared 1b - Glitch detector status asserted
15 PORAO18F	VDD1V8_AO Power-on Reset Flag Indicates VDD1V8_AO power-on reset status. This bit is set by VDD1V8_AO POR. 0b - POR status cleared 1b - POR status asserted
14 POR1V8F	VDD1V8 Power-on Reset Flag Indicates VDD1V8 power-on reset status. This bit is set by VDD1V8 POR. 0b - POR status cleared 1b - POR status asserted
13 PORVDDNF	VDDN Power-on Reset Flag Indicates VDDN power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
12 PORVDD2F	VDD2 Power-on Reset Flag Indicates VDD2 power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
11	VDD1 Power-on Reset Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
PORVDD1F	Indicates VDD1 power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
10 DCLPOCF	DCDC Low-Power Mode Overcurrent Flag Indicates the DCDC had an overcurrent event while in low-power mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
9 DCHPOCF	DCDC High-Power Mode Overcurrent Flag Indicates the DCDC had an overcurrent event while in high-power mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
8 SCP2OCF	SCPC VDD2 Overcurrent Flag Indicates that an SCPC2 overcurrent event was detected. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
7 SCP1OCF	SCPC VDD1 Overcurrent Flag Indicates that an SCPC1 overcurrent event was detected. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
6 BBNEF	VDDN Body Bias Error Flag Indicates body bias error in VDDN domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
5	VDD2 Body Bias Error Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
BB2EF	Indicates body bias error in VDD2 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
4 BB1EF	VDD1 Body Bias Error Flag Indicates body bias error in VDD1 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
3 BBSR2EF	VDD2 SRAM Body Bias Error Flag Indicates SRAM body bias error in VDD2 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No SRAM body bias error 1b - SRAM body bias error detected
2 BBSR1EF	VDD1 SRAM Body Bias Error Flag Indicates SRAM body bias error in VDD1 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No SRAM body bias error 1b - SRAM body bias error detected
1 ITRCRF	ITRC RAM ZEROIZE Flag Indicates ITRC triggered RAM ZEROIZE. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Not detected 1b - ITRC RAM ZEROIZE detected
0 DSALLF	Deep Sleep Flag Indicates whether CPU0 and CPU1 are in Deep Sleep mode at the same time. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU0 and CPU1 are not in Deep Sleep mode at the same time 1b - CPU0 and CPU1 have both been in Deep Sleep mode at the same time

27.7.1.5 Control (CTRL)

Offset

Register	Offset
CTRL	Ch

Function

Enables controls for triggering configuration updates and enabling voltage monitor resets.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							HVD1V8RE	HVDVNRE	HVDV2RE	HVDV1RE	LVDNRE	LVD2RE	LVD1RE	AGDET2RE	AGDET1RE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															APPLYCFG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-25 —	Reserved
24 HVD1V8RE	VDD1V8 High-Voltage Detector Reset Enable Enables reset for VDD1V8 high-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
23 HVDVNRE	VDDN High-Voltage Detector Reset Enable Enables reset for VDDN high-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
22 HVDV2RE	VDD2 High-Voltage Detector Reset Enable Enables reset for VDD2 high-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
21 HVDV1RE	VDD1 High-Voltage Detector Reset Enable Enables reset for VDD1 high-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
20 LVDNRE	VDDN Low-Voltage Detector Reset Enable Enables reset for VDDN low-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
19 LVD2RE	VDD2 Low-Voltage Detector Reset Enable Enables reset for VDD2 low-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
18 LVD1RE	VDD1 Low-Voltage Detector Reset Enable Enables reset for VDD1 low-voltage detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
17 AGDET2RE	VDD2 Glitch Detector Reset Enable Enables reset for VDD2 glitch detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
16 AGDET1RE	VDD1 Glitch Detector Reset Enable Enables reset for VDD1 glitch detector. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Disables 1b - Enables
15-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 APPLYCFG	Apply Configure for PMC0 Applies updated PDRUNCFG bits (SRAM power gates, RBB, AFBB, LVD, and HVD control bits) and/or RUNCTRL setting. Automatically clears when operation is done. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Operation is done 1b - Start configuration change, or operation has not completed

27.7.1.6 Interrupt Control in Compute Domain (INTRCTRL)

Offset

Register	Offset
INTRCTRL	10h

Function
Enables interrupt control for PMC interrupt sources.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DSSENSIE	0	INTNIE	AUTOWKIE	0	HVD1V8IE	HVDNIE	HVD2IE	HVD1IE	LVDNIE	LVD2IE	LVD1IE	AGDET2IE	AGDET1IE
W		0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DCLPOCIE	DCHPOCIE	SCP2OCIE	SCP1OCIE	BBNEIE	BB2EIE	BB1EIE	BBSR2EIE	BBSR1EIE	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 —	Reserved
29 DSSENSIE	CPU1 Deep Sleep Interrupt Enable Enables interrupt for CPU1 entered into Deep Sleep mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
28 —	Reserved
27 INTNIE	PMIC_IRQN PAD Interrupt Enable Enables PMIC_IRQN PAD interrupt. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
26 AUTOWKIE	Auto Wake-up Interrupt Enable Enables auto wake-up interrupt. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
25 —	Reserved
24 HVD1V8IE	VDD1V8 High-Voltage Detector Interrupt Enable Enables interrupt for VDD1V8 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
23 HVDNIE	VDDN High-Voltage Detector Interrupt Enable Enables interrupt for VDDN high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
22 HVD2IE	VDD2 High-Voltage Detector Interrupt Enable Enables interrupt for VDD2 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
21 HVD1IE	VDD1 High-Voltage Detector Interrupt Enable Enables interrupt for VDD1 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
20 LVDNIE	VDDN Low-Voltage Detector Interrupt Enable Enables interrupt for VDDN low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
19 LVD2IE	VDD2 Low-Voltage Detector Interrupt Enable Enables interrupt for VDD2 low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
18 LVD1IE	VDD1 Low-Voltage Detector Interrupt Enable Enables interrupt for VDD1 low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
17 AGDET2IE	VDD2 Glitch Detector Interrupt Enable Enables interrupt for VDD2 glitch detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
16	VDD1 Glitch Detector Interrupt Enable Enables interrupt for VDD1 glitch detector. This bit is reset by cold reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
AGDET1IE	This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
15-11 —	Reserved
10 DCLPOCIE	DCDC Low-Power Overcurrent Interrupt Enable Enables interrupt for DCDC overcurrent in low-power mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
9 DCHPOCIE	DCDC High-Power Overcurrent Interrupt Enable Enables interrupt for DCDC overcurrent in high-power mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
8 SCP2OCIE	SCPC VDD2 Overcurrent Interrupt Enable Enables interrupt for SCPC overcurrent in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
7 SCP1OCIE	SCPC VDD1 Overcurrent Interrupt Enable Enables interrupt for SCPC overcurrent in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
6 BBNEIE	VDDN Body Bias Error Interrupt Enable Enables interrupt for body bias error in VDDN domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
5	VDD2 Body Bias Error Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
BB2EIE	Enables interrupt for body bias error in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
4 BB1EIE	VDD1 Body Bias Error Interrupt Enable Enables interrupt for body bias error in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
3 BBSR2EIE	VDD2 SRAM Body Bias Error Interrupt Enable Enables interrupt for SRAM body bias error in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
2 BBSR1EIE	VDD1 SRAM Body Bias Error Interrupt Enable Enables interrupt for SRAM body bias error in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
1-0 —	Reserved

27.7.1.7 DCDC Voltage Selection (DCDCVSEL)

Offset

Register	Offset
DCDCVSEL	20h

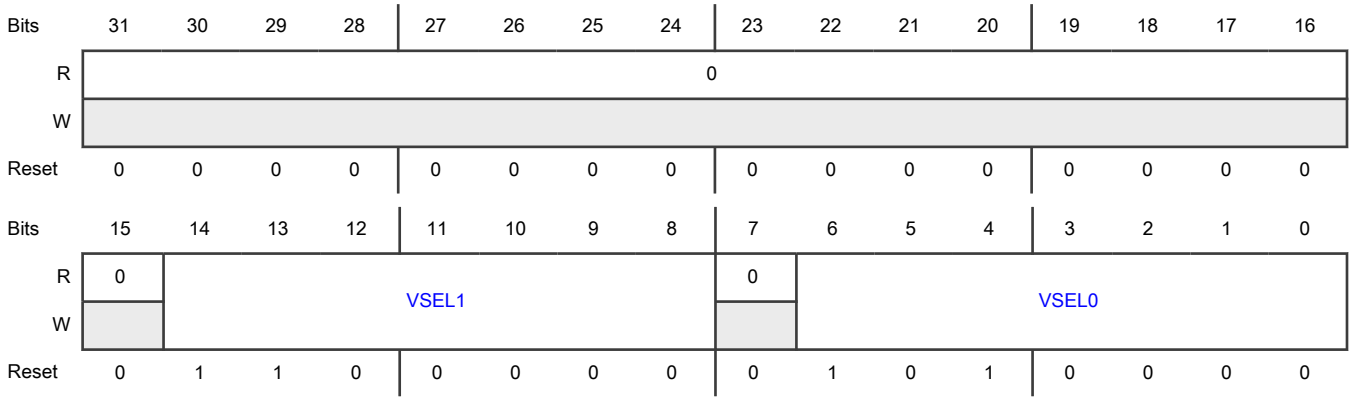
Function

Configures DCDC voltage levels for selection by PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL].

NOTE

To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Diagram



Fields

Field	Function
31-15 —	Reserved
14-8 VSEL1	DCDC Voltage Level 1 Sets the DCDC voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 1. This is the level selected by PD*CFG after reset, where • Voltage = 0.5 V + 6.25 mV * value of [VSEL1] • Reset value = 1.1 V • Written value is clipped to 68h • Maximum value = 1.15 V **NOTE** The value of [VSEL1] must be set larger than that of [VSEL0]. The Sense slot version of this field is read-only. This bit is reset by warm reset. 000_0000b - 0.50 V 010_0000b - 0.70 V 100_0000b - 0.90 V 110_0000b - 1.10 V 110_1000b - 1.15 V (maximum) 111_1111b - 1.15 V (clipped)
7 —	Reserved
6-0	DCDC Voltage Level 0

Table continues on the next page...

Table continued from the previous page...

Field	Function
VSEL0	Sets the DCDC voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 0. • Voltage = 0.5 V + 6.25 mV * value of [VSEL0] • Reset value = 1.0 V • Written value is clipped to 68h • Maximum value = 1.15 V **NOTE** The value of [VSEL0] must be set smaller than that of [VSEL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset. 000_0000b - 0.50 V 010_0000b - 0.70 V 100_0000b - 0.90 V 110_0000b - 1.10 V 110_1000b - 1.15 V (maximum) 111_1111b - 1.15 V (clipped)

27.7.1.8 LDO VDD2 Voltage Selection (LDOVDD2VSEL)

Offset

Register	Offset
LDOVDD2VSEL	24h

Function

Configures LDO VDD2 voltage levels for selection by PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		VSEL3						0		VSEL2					
W																
Reset	0	0	1	0	1	1	0	0	0	0	1	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		VSEL1						0		VSEL0					
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0

Fields

Field	Function
31-30 —	Reserved
29-24 VSEL3	LDO VDD2 Voltage Level 3 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL3] • Reset value = 1.0 V • Written value is clipped to 38h **NOTE** The value of [VSEL3] must be set larger than that of [VSEL2]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
23-22 —	Reserved
21-16 VSEL2	LDO VDD2 Voltage Level 2 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 2. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL2] • Reset value = 0.9 V • Written value is clipped to 38h **NOTE** The value of [VSEL2] must be set larger than that of [VSEL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
15-14 —	Reserved
13-8 VSEL1	LDO VDD2 Voltage Level 1 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 1. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL1] • Reset value = 0.8 V • Written value is clipped to 38h **NOTE** The value of [VSEL1] must be set larger than that of [VSEL0]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG] or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
7-6 —	Reserved
5-0	LDO VDD2 Voltage Level 0

Table continues on the next page...

Table continued from the previous page...

Field	Function
VSEL0	Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 0. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL0] • Reset value = 0.7 V • Written value is clipped to 38h **NOTE** The value of [VSEL0] must be set smaller than that of [VSEL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG] or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)

27.7.1.9 LDO VDD1 Voltage Selection (LDOVDD1VSEL)

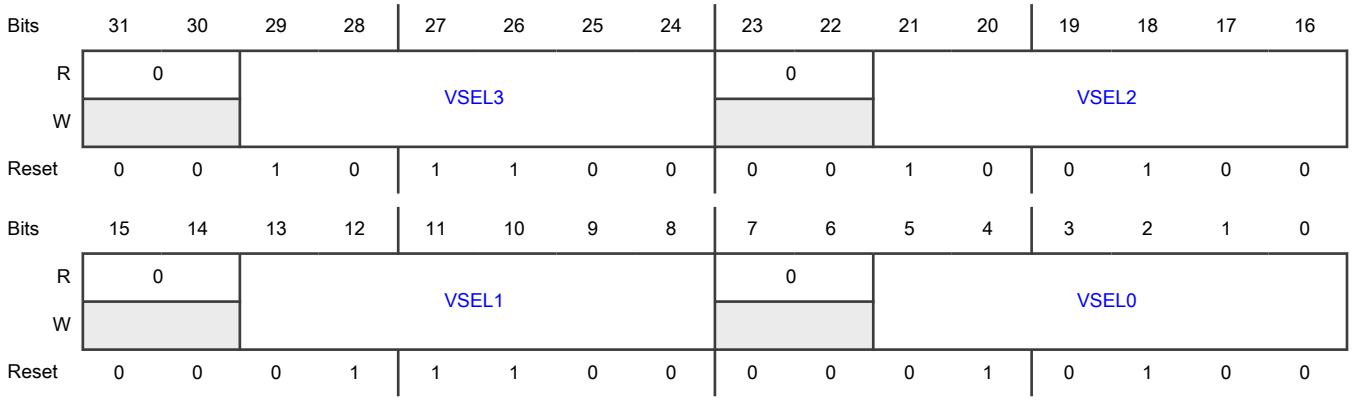
Offset

Register	Offset
LDOVDD1VSEL	28h

Function

Configures LDO VDD1 voltage levels for selection by PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL]. Fields of this register are shared between PMC0 and PMC1. Both Compute and Sense slots can write to this register.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 VSEL3	LDO VDD1 Voltage Level 3 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL3] • Reset value = 1.0 V • Written value is clipped to 38h **NOTE** The value of [VSEL3] must be set larger than that of [VSEL2]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
23-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
21-16 VSEL2	LDO VDD1 Voltage Level 2 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 2. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL2] • Reset value = 0.9 V • Written value is clipped to 38h **NOTE** The value of [VSEL2] must be set larger than that of [VSEL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
15-14 —	Reserved
13-8 VSEL1	LDO VDD1 Voltage Level 1 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 1. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL1] • Reset value = 0.8 V • Written value is clipped to 38h **NOTE** The value of [VSEL1] must be set larger than that of [VSEL0]. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
7-6 —	Reserved
5-0 VSEL0	LDO VDD1 Voltage Level 0 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 0. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL0] • Reset value = 0.7 V • Written value is clipped to 38h **NOTE** The value of [VSEL0] must be set smaller than that of [VSEL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)

27.7.1.10 VDDN LVD Control (LVDVDDNCTRL)

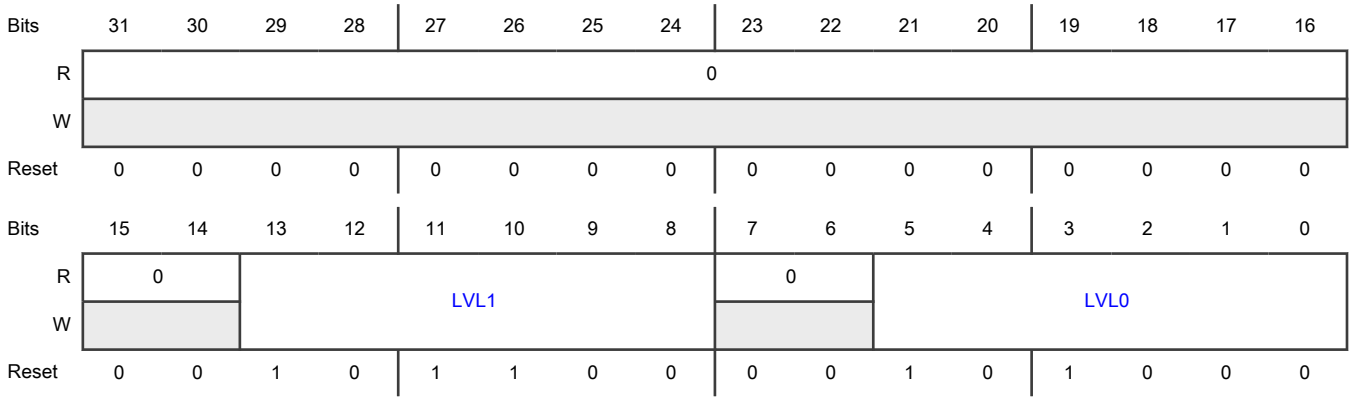
Offset

Register	Offset
LVDVDDNCTRL	2Ch

Function

Sets the VDDN LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL].

Diagram



Fields

Field	Function
31-14 —	Reserved
13-8 LVL1	LVD Level 1 Voltage Sets the VDDN LVD falling trip voltage when the aggregated value of aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 1. This is the level selected by PD*CFG after reset, where: • Falling trip = 0.5 + 10 mV * value of [LVL1] • Reset value = 0.94 V **NOTE** The value of [LVL1] must be set larger than that of [LVL0]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
7-6 —	Reserved
5-0	LVD Level 0 Voltage

Table continues on the next page...

Table continued from the previous page...

Field	Function
LVL0	Sets the VDDN LVD falling trip voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 0. • Falling trip = 0.5 + 10 mV * value of [LVL0] • Reset value = 0.9 V **NOTE** The value of [LVL0] must be set smaller than that of [LVL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.1.11 VDD2 LVD Control (LVDVDD2CTRL)

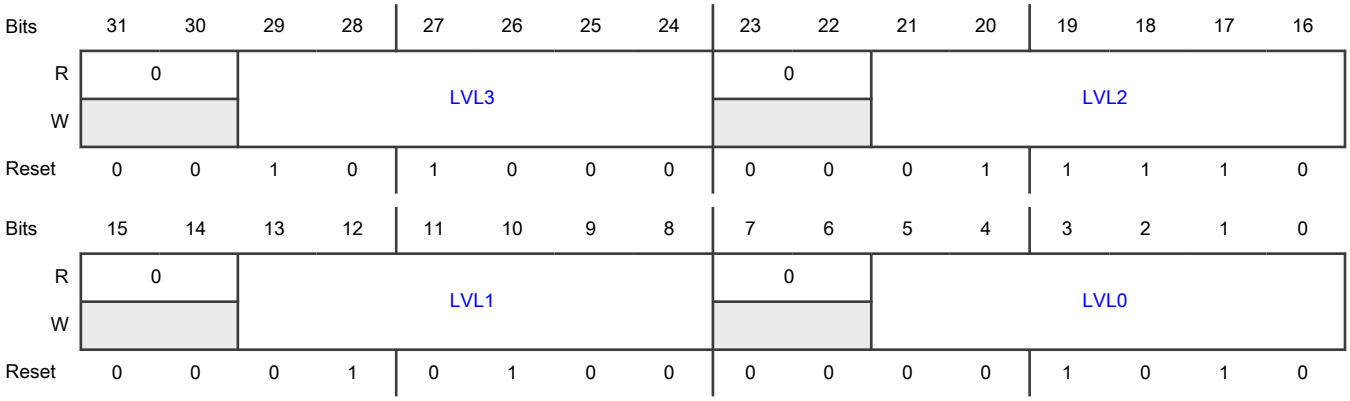
Offset

Register	Offset
LVDVDD2CTRL	30h

Function

Sets the VDD2 LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL].

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 LVL3	LVD Level 3 Voltage Sets the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Falling trip = 0.5 + 10 mV * value of [LVL3] • Reset value = 0.9 V **NOTE** The value of [LVL3] must be set larger than that of [LVL2]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
23-22 —	Reserved
21-16 LVL2	LVD Level 2 Voltage Sets the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 2. • Falling trip = 0.5 + 10 mV * value of [LVL2] • Reset value = 0.8 V **NOTE** The value of [LVL2] must be set larger than that of [LVL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
15-14 —	Reserved
13-8 LVL1	LVD Level 1 Voltage Sets the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 1. • Falling trip = 0.5 + 10 mV * value of [LVL1] • Reset value = 0.7 V **NOTE** The value of [LVL1] must be set larger than that of [LVL0]. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
7-6 —	Reserved
5-0 LVL0	LVD Level 0 Voltage Sets the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 0. • Falling trip = 0.5 + 10 mV * value of [LVL0] • Reset value = 0.6 V **NOTE** The value of [LVL0] must be set smaller than that of [LVL1]. The Sense slot version of this field is read-only. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.1.12 VDD1 LVD Control (LVDVDD1CTRL)

Offset

Register	Offset
LVDVDD1CTRL	34h

Function

Sets the VDD1 LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		LVL3						0		LVL2					
W																
Reset	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		LVL1						0		LVL0					
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0

Fields

Field	Function
31-30 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
29-24 LVL3	Level 3 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Falling trip = 0.5 + 10 mV * value of [LVL3] **NOTE** The value of [LVL3] must be set larger than that of [LVL2]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
23-22 —	Reserved
21-16 LVL2	Level 2 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 2. • Falling trip = 0.5 + 10 mV * value of [LVL2] **NOTE** The value of [LVL2] must be set larger than that of [LVL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-14 —	Reserved
13-8 LVL1	Level 1 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 1. • Falling trip = 0.5 + 10 mV * value of [LVL1] **NOTE** The value of [LVL1] must be set larger than that of [LVL0]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.
7-6 —	Reserved
5-0 LVL0	Level 0 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 0. • Falling trip = 0.5 + 10 mV * value of [LVL0] **NOTE** The value of [LVL0] must be set smaller than that of [LVL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.1.13 Power-on Reset Control (PORCTRL)

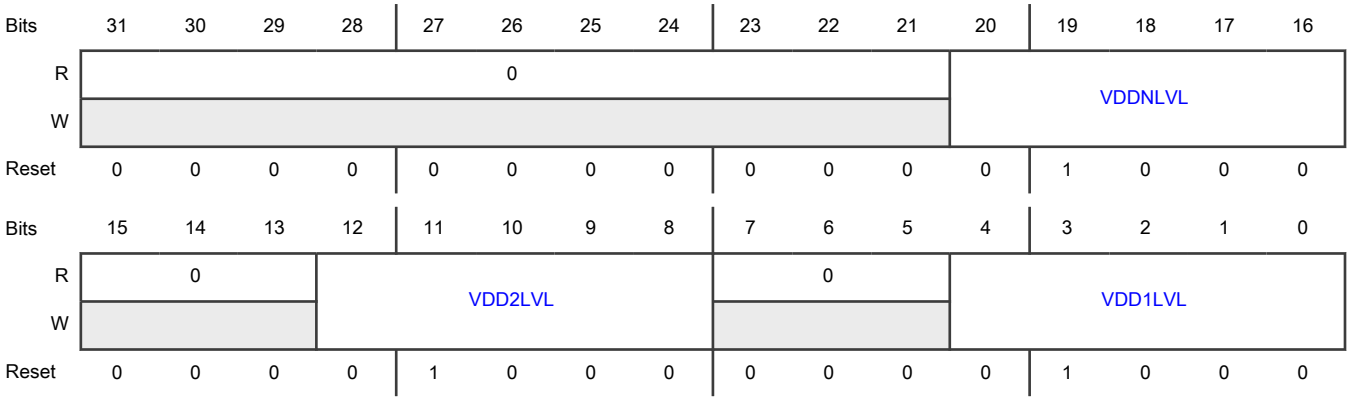
Offset

Register	Offset
PORCTRL	38h

Function

Sets the POR falling trip voltages in VDD1, VDD2, and VDDN domains. When the supply voltage falls below the selected trip voltage, it triggers a cold reset.

Diagram



Fields

Field	Function
31-21 —	Reserved
20-16 VDDNLVL	POR Level in VDDN Domain Sets the POR falling trip voltage in VDDN domain. The Sense slot version of this field is read-only. • Falling trip = 0.4 + 10 mV * value of [VDDNLVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V 0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V
15-13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
12-8 VDD2LVL	POR Level in VDD2 Domain Sets the POR falling trip voltage in VDD2 domain. The Sense slot version of this field is read-only. • Falling trip = 0.4 + 10 mV * value of [VDD2LVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V 0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V
7-5 —	Reserved
4-0 VDD1LVL	POR Level in VDD1 Domain Sets the POR falling trip voltage in VDD1 domain. The Sense slot version of this field is read-only. • Falling trip = 0.4 + 10 mV * value of [VDD1LVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V 0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V

27.7.1.14 Body Bias Control (BBCTRL)

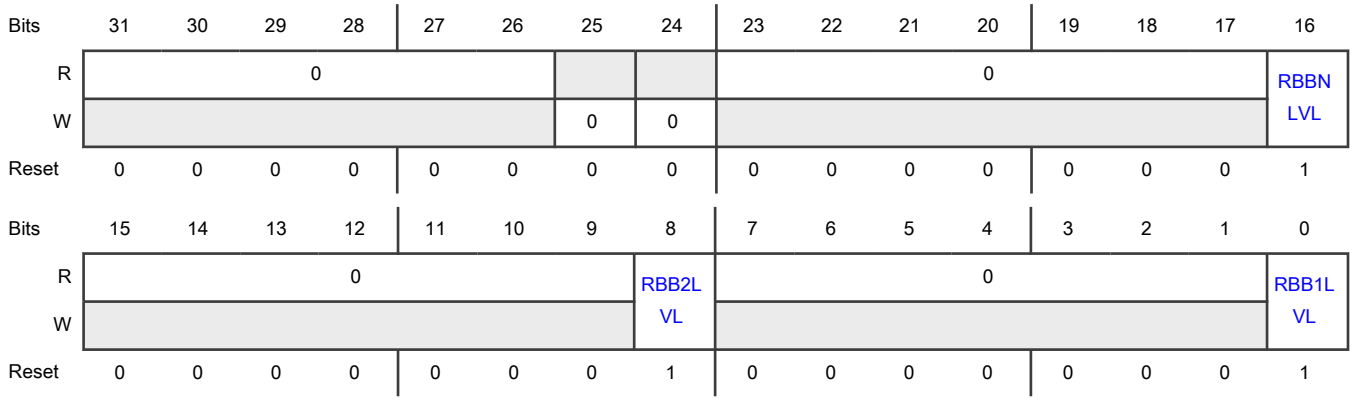
Offset

Register	Offset
BBCTRL	40h

Function

Controls the configuration of the body bias blocks when they are enabled by PD*CFG.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 —	Reserved
24 —	Reserved
23-17 —	Reserved
16 RBBNLVL	Reverse Body Bias Voltage for VDDN Domain Sets RBB Voltage for VDDN domain. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V
15-9 —	Reserved
8	Reverse Body Bias Voltage for VDD2 Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
RBB2LVL	Sets RBB Voltage for VDD2 domain. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V
7-1 —	Reserved
0 RBB1LVL	Reverse Body Bias Voltage for VDD1 Domain Sets RBB Voltage for VDD1 domain. The Sense slot version of this field is read-only. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V

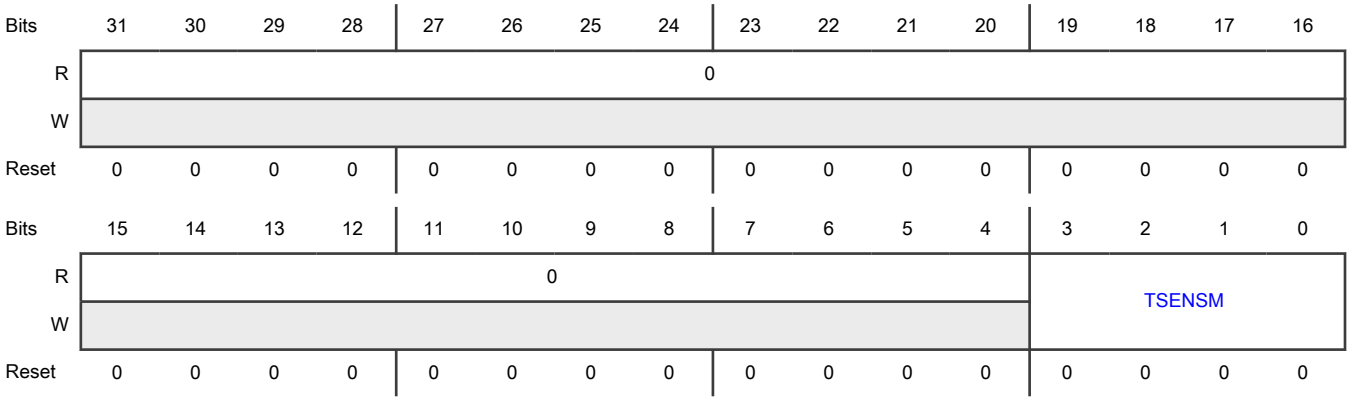
27.7.1.15 Temperature Sensor Control (TSENSOR)

Offset

Register	Offset
TSENSOR	44h

Function
Controls the PMC temperature sensor measurement mode. See PMC temperature sensor for details.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TSENSM	Temperature Sensor Mode Select Selects the temperature sensor mode according to the procedure described in PMC temperature sensor. All modes must be selected in sequence during the temperature measurement procedure to dynamically cancel various internal error sources. **NOTE** This field is shared between PMC0 and PMC1. Software should only do temperature measurement from one slot at a time. 0000b - Perform AD conversion and record C*_000 0001b - Perform AD conversion and record C*_001 0010b - Perform AD conversion and record C*_010 0011b - Perform AD conversion and record C*_011 0100b - Perform AD conversion and record CM_100 0101b - Perform AD conversion and record C*_101 0110b - Perform AD conversion and record C*_110 0111b - Perform AD conversion and record C*_111

27.7.1.16 Wake-up Configuration for Compute Domain (WAKEUP)

Offset

Register	Offset
WAKEUP	50h

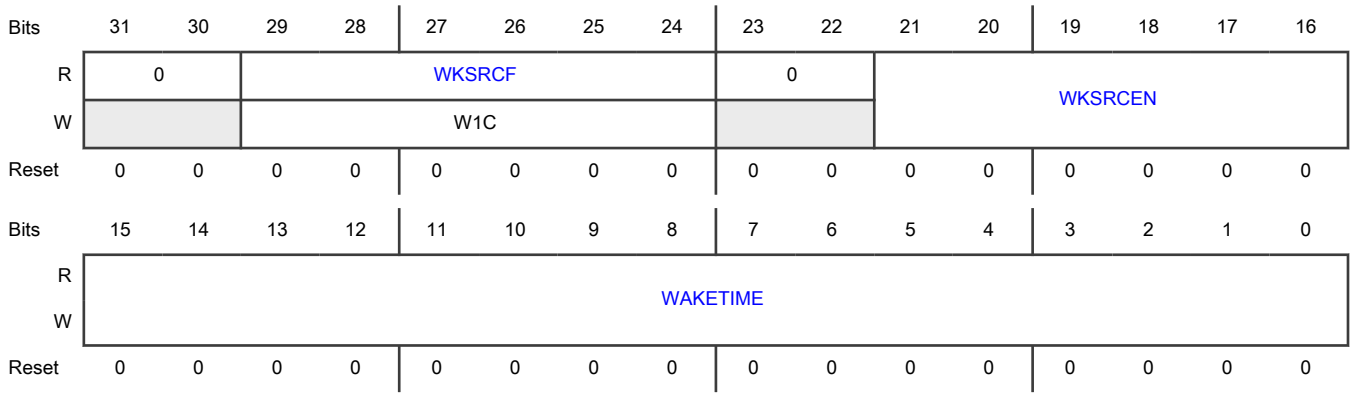
Function

Sets wake-up sources for PMC when VDD2_COMP and VDD2_COM domains are in DSR (Deep Sleep Retention) mode, and sets the configuration of the general purpose wake-up counter.

When VDD2_COMP and VDD2_COM domains are both in DSR mode, but VDD1_SENSE domain is not (i.e. the chip is not in FDSR mode), SLEEPCON0 is powered down, so it cannot wake up from sources in the VDD1_SENSE domain. When wake-up is enabled here, PMC can wake up from some VDD1_SENSE domain sources.

The wake-up timer is enabled by INTRCTRL[AUTOWKIE], and will wake up the associated PMC slot when the associated CPU enters the Deep Sleep mode. When the counter times out, PMC wakes up and generates an interrupt for that slot.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 WKSRCF	Wake-up Source Flags Indicates if wake-ups have been triggered by the respective source. Bit 29: ADC0 Bit 28: ACMP Bit 27: MU1 Bit 26: MU0 Bit 25: MICFIL Bit 24: OSTIMER This bit is reset by warm reset. 00_0000b - No wake-up from source 00_0001b - Source has triggered a wake-up
23-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
21-16 WKSRCEN	Wake-up Source Enable Enables wake-up from Sense domain MU, OSTIMER, or other sources. Bit 21: ADC0 Bit 20: ACMP Bit 19: MU1 Bit 18: MU0 Bit 17: MICFIL[1] Bit 16: OSTIMER This bit is reset by warm reset. 00_0000b - Wake-up disabled 00_0001b - Wake-up enabled
15-0 WAKETIME	Wake-up Timer Sets timer to wake up from Deep Sleep mode after mode sequencing is completed. See Autowake for more information. Time = value of [WAKETIME] / 16 MHz = value of [WAKETIME] / 16 in µs This field has two copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0000_0000_0000_0000b - 0 0000_0000_0000_0001b - 62.5 ns = 1 / (16e + 6) 0000_0000_0000_0010b - 125 ns = 2 / (16e + 6) ...

1. There are two MICFIL interrupts:
   - Interrupt for read data or error
   - Interrupt for hardware voice activity detector (HWVAD) or HWVAD error

   Only the MICFIL interrupt for read data or error can be used as the wake-up source.

27.7.1.17 Power Configuration (POWERCFG)

Offset

Register	Offset
POWERCFG	64h

Function

Configures internal regulators and internal references during FDSR (Full Deep Sleep Retention) mode and FDPD (Full Deep Power Down) modes, or when external supplies are used. Also configures the wait time for PMIC or external supplies to respond to changes on the PMIC_MODE pins.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MODEDLY				0			FDPD BGLP	0							
W																
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			DCDC PD	0		LDO2 PD	LDO1 PD	SCPC PD	FDDC PD	FDLD O2PD	FDLD O1PD	0		FDSC P2PD	FDSC P1PD
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Fields

Field	Function
31-28 MODEDLY	Mode Delay Sets delay to wait for external PMIC mode changes to complete after any mode value change occurs on the PMIC_MODE[1:0] pins. The PMIC_MODE pins change state whenever the aggregated PD*CFG0[PMICMODE] changes. • Time = Period of 4 clocks * 2 ^ value of [MODEDLY] • Reset value = 16 µs The Sense slot version of this field is read-only. This bit is reset by cold reset. Values shown below are examples based on calculations with the period of 4 clocks (250 ns) at a frequency of 16 MHz. 0000b - 250 ns 0001b - 500 ns 0010b - 1000 ns 0011b - 2000 ns 0100b - 4000 ns 0101b - 8000 ns 0110b - 16 µs 0111b - 32 µs 1000b - 64 µs 1001b - 128 µs 1010b - 256 µs 1011b - 512 µs 1100b - 1024 µs 1101b - 2048 µs 1110b - 4096 µs

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1111b - 8192 µs
27-25 —	Reserved
24 FDPDBGLP	Force Band Gap during FDPD Mode to LP Mode In FDPD mode, force band gap to low-power mode. Rising VDD1V8 will not wake up the chip from FDPD mode when this bit is set. If clear, band gap will remain in high-power mode and consume 85 µA. The Sense slot version of this field is read-only. This bit is reset by cold reset. **NOTE** This bit is cleared on wake-up from FDPD mode and any other cold reset! If the rising VDD1V8 is not used as the wake-up source then this bit should be set before entering FDPD every time. 0b - Enables band gap to high-power mode during FDPD mode and wakes up if VDD1V8 POR is asserted then negated 1b - Enables band gap to low-power mode during FDPD mode and will not wake up on VDD1V8 POR negation. RTC, PMIC_IRQN, or reset is used as wake-up
23-13 —	Reserved
12 DCDCPD	DCDC Power Down Disables DCDC when using a PMIC or other external supply to drive VDDN. The Sense slot version of this field is read-only. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDDN never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes. **NOTE** If this bit is set, then software should also write 11b to PMC0.PDRUNCFG0[DCDC_MODE], PMC0.PDSLEEPCFG0[DCDC_MODE], PMC1.PDRUNCFG0[DCDC_MODE], and PMC1.PDSLEEPCFG0[DCDC_MODE]. to minimize power when the other blocks of the PMC are all in low power mode and/or powered down. 0b - Not powered down, except in FDSR mode when [FDDCPD] = 1, or in DPD or FDPD mode 1b - Powered down (until the next cold reset)
11-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
9 LDO2PD	LDO VDD2 Power Down Disables LDO VDD2 when using a PMIC or other external supply to drive VDD2. The Sense slot version of this field is read-only. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDD2 never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes. 0b - Not powered down, except in FDSR mode when [FDLDO2PD] = 1, or in DPD or FDPD mode 1b - Powered down (until the next cold reset)
8 LDO1PD	LDO VDD1 Power Down Disables LDO VDD1 when using a PMIC or other external supply to drive VDD1. The Sense slot version of this field is read-only. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDD1 never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes. 0b - Not powered down, except in FDSR mode when [FDLDO1PD] = 1, or in DPD or FDPD mode 1b - Powered down (until the next cold reset)
7 SCPCPD	SCPC Power Down Disables both SCPCs when a PMIC is driving. **NOTE** This field should only be set if FDSR mode is configured to always use the LDOs, or in applications that always use external supplies for VDD1 and VDD2. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Not powered down in FDSR mode, except in FDSR mode when [FDSCPnPD] = 1. SCPC might be powered down in FDSR depending on [FDSCPnPD] field 1b - Powered down (until the next cold reset)
6 FDDCPD	DCDC Power Down in FDSR Mode Disables DCDC in FDSR mode. This bit is overridden by [DCDCPD] = 1. The Sense slot version of this field is read-only. This bit is reset by cold reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not powered down 1b - Powered down
5 FDLDO2PD	LDO VDD2 Power Down in FDSR Mode Disables LDO VDD2 in FDSR mode. This bit is overridden by [LDO2PD] = 1. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Not powered down 1b - Powered down
4 FDLDO1PD	LDO VDD1 Power Down in FDSR Mode Disables LDO VDD1 in FDSR mode. This bit is overridden by [LDO1PD] = 1. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Not powered down 1b - Powered down
3-2 —	Reserved
1 FDSCP2PD	SCPC VDD2 Power Down in FDSR Mode Disables SCPC VDD2 in FDSR mode. This bit is overridden by [SCPCPD] = 1. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Not powered down 1b - Powered down
0 FDSCP1PD	SCPC VDD1 Power Down in FDSR Mode Disables SCPC VDD1 in FDSR mode. This bit is overridden by [SCPCPD] = 1. The Sense slot version of this field is read-only. This bit is reset by cold reset. 0b - Not powered down 1b - Powered down

27.7.1.18 3 V Capable VDDIO Range Select (PADVRANGE)

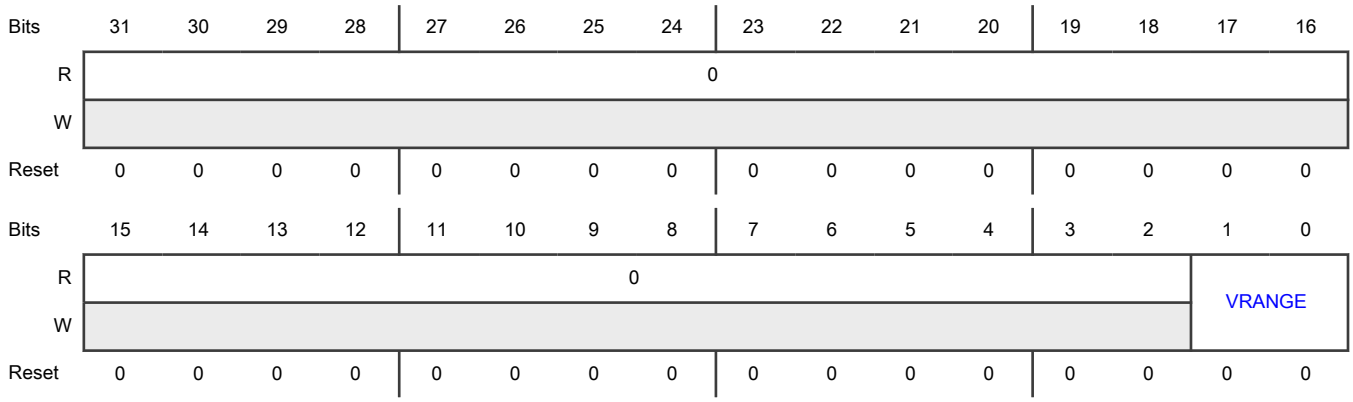
Offset

Register	Offset
PADVRANGE	68h

Function

Selects the allowed voltage range for the 3 V capable IOs which have a detector to Sense the value of supply, but this consumes current. To reduce power the IOs can be configured for 1.8 V or 3.3 V operation, powering down the detector, but requiring the application to restrict the IO supply voltage to the selected range. The detector is enabled after any cold reset.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 VRANGE	3 V Capable IO Bank VDDIO Range Select Selects the VDDIO range configuration. The Sense slot version of this field is read-only. This bit is reset by cold reset. 00b - Continuous mode. Allows 1.71 - 3.6 V range but consumes static current. 01b - 1.71 - 1.98 V, supply detector off 10b - 3.00 - 3.6 V, supply detector off 11b - Reserved

27.7.1.19 IO Bank State Configuration (PADCFG)

Offset

Register	Offset
PADCFG	6Ch

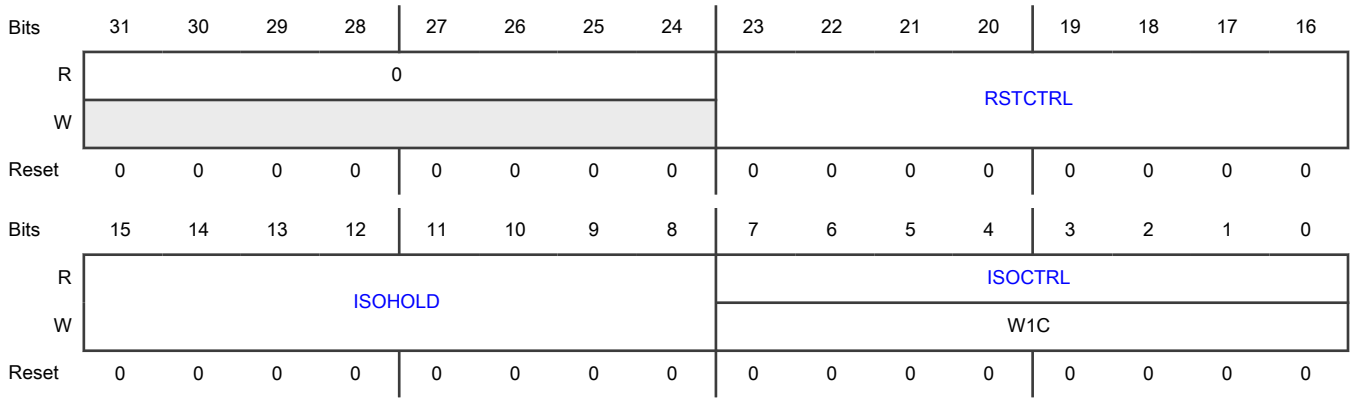
Function

Controls IO bank reset and state retain. Each bit field has the same IO bank distribution map.

Table 232. IO bank mapping

Bit	IO bank(s)	Domain
Bit 7	PMIC_I2C	VDD1_SENSE
Bit 6	Port 8/9/10	VDD1_SENSE
Bit 5	Port 7	VDDN_COM
Bit 4	Port 6	VDDN_COM
Bit 3	Port 5	VDDN_COM
Bit 2	Port 4	VDDN_COM
Bit 1	Port 2	VDD2_COM
Bit 0	Port 0/1/3	VDD2_COM

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 RSTCTRL	Reset Control Sets the IO bank reset which tristates the ports. Intended to be used prior to powering off the VDDIO supplies. This bit is reset by cold reset. 0000_0000b - IO banks are not reset 1xxx_xxxxb - PMIC_I2C reset = tristate x1xx_xxxxb - Port 8/9/10 reset = tristate xx1x_xxxxb - Port 7 reset = tristate xxx1_xxxxb - Port 6 reset = tristate

Table continues on the next page...

Table continued from the previous page...

Field	Function
	xxxx_1xxxb - Port 5 reset = tristate xxxx_x1xxb - Port 4 reset = tristate xxxx_xx1xb - Port 2 reset = tristate xxxx_xxx1b - Port 0/1/3 reset = tristate
15-8 ISOHOLD	IO Bank Isolation Hold Maintains IO bank isolation state after wake-up from FDSR/DPD modes. If clear, the IO bank's state will be retained during FDSR & DPD modes, but will be controllable after wake-up. Clear to 0 so that after PMC powers the controlling logic domain, the isolation is automatically removed and the port is once again controlled by the controlling domain. If the controlling domain is state-retained, then the port state will not change when the isolation is auto-negated. If set, the IO bank will remain in retain mode until ISOCTRL cleared by software. Set to 1 for banks that you want to remain isolated even after power is restored to the controlling logic. If the controlling logic's state is not retained, this enables the software to configure the controlling logic back to its previous state before clearing the isolation/retention to restore control back to the logic. This bit is reset by cold reset. 0000_0000b - IO bank is controllable on wake-up from FDSR/DPD mode 1xxx_xxxxb - PMIC_I2C is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 7 to regain control x1xx_xxxxb - Port 8/9/10 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 6 to regain control xx1x_xxxxb - Port 7 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 5 to regain control xxx1_xxxxb - Port 6 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 4 to regain control xxxx_1xxxb - Port 5 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 3 to regain control xxxx_x1xxb - Port 4 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 2 to regain control xxxx_xx1xb - Port 2 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 1 to regain control xxxx_xxx1b - Port 0/1/3 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 0 to regain control
7-0 ISOCTRL	IO bank State Retain Control Controls IO bank state retain, one bit for each VDDIO/VDD domain pair that consists of a controlling VDD logic domain (for example, VDD2_COM or VDD1_SENSE) and a VDDIO supply (VDDIO, VDDIO1, etc.). Hardware sets this bit when driving domain powers down (FDSR/DPD). This bit cannot be cleared if associated domain is not powered. This bit is reset by cold reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000_0000b - IO bank is controllable 1xxx_xxxxb - PMIC_I2C is in retain mode x1xx_xxxxb - Port 8/9/10 is in retain mode xx1x_xxxxb - Port 7 is in retain mode xxx1_xxxxb - Port 6 is in retain mode xxxx_1xxxb - Port 5 is in retain mode xxxx_x1xxb - Port 4 is in retain mode xxxx_xx1xb - Port 2 is in retain mode xxxx_xxx1b - Port 0/1/3 is in retain mode

27.7.1.20 PD Run Configuration 0 in Compute Domain (PDRUNCFG0)

Offset

Register	Offset
PDRUNCFG0	A0h

Function

Configures power down for CPU0 in active mode. This drives the aggregation input for the power down and other controls of PMC functions when the Compute domain is in active mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Compute domain CTRL[APPLYCFG] is set, triggering the state machines or CPU0 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W	1	1														
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	V2DSP_PD	VNCOM_D...	V2COM_D...	V2NMED_...	V2COMP_...	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 RBBSR2_PD	Power Down SRAM RBB in VDD2 Domain SRAM RBB should be activated only when no SRAMs are accessed in the VDD2 domain. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain SRAM RBB should be activated only when no SRAMs are accessed in the VDD1 domain. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
27 AFBBN_PD	Power Down AFBB in VDDN Domain Controls AFBB mode for VDDN domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
26	Power Down RBB in VDDN Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
RBBN_PD	Controls RBB mode for VDDN domain. RBB mode should only be enabled when VDDN domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBBN_PD] = 1, power on. Else power down 1b - Power down
25 AFBB2_PD	Power Down AFBB in VDD2 Domain Controls AFBB mode for VDD2 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
24 RBB2_PD	Power Down RBB in VDD2 Domain Controls RBB mode for VDD2 domain. RBB mode should only be enabled when VDD2 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBB2_PD] = 1, power on. Else power down 1b - Power down
23 AFBB1_PD	Power Down AFBB in VDD1 Domain Controls AFBB mode for VDD1 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 Controls RBB mode for VDD1 domain. RBB mode should only be enabled when VDD1 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBB1_PD] = 1, power on. Else power down 1b - Power down
21-20 LDO2_VSEL	Select LDO VDD2 Voltage Selects one of four voltages in LDOVDD2VSEL register. Also selects from one of four LVD trip points in LVDVDD2CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level 01b - Selects LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level 10b - Selects LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level 11b - Selects LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level
19-18 LDO2_MODE	LDO VDD2 Mode Selects regulation mode of LDO VDD2. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Select Selects one of four voltages in LDOVDD1VSEL register. Also selects from one of four LVD trip points in LVDVDD1CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level 01b - Selects LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level 10b - Selects LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level 11b - Selects LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level
15-14 LDO1_MODE	LDO VDD1 Mode Selects regulation mode of LDO VDD1. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
13 DCDC_VSEL	Select DCDC Voltage Selects PMC DCDC voltage and LVD level VDDN supply rail from DCDCVSEL registers. DCDCVSEL[VSEL1] is the higher voltage for aggregation. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has 2 copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Selects DCDCVSEL[VSEL0] level 1b - Selects DCDCVSEL[VSEL1] level
12-11 DCDC_MODE	DCDC Power Mode Selects DCDC power mode. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has 2 copies separately accessed by the Compute and Sense slots. This field is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** When using an external supply to power VDDN and POWERCFG[DCDCPD] = 1b, software must also set field DCDC_MODE = 11b in PMC0.PDRUNCFG0, PMC0.PDSLEEPCFG0, PMC1.PDRUNCFG0, and PMC1.PDSLEEPCFG to minimze power when all other PMC subblocks go to low power or power down mode. 00b - Enables DCDC in high-power mode 01b - Enables DCDC in low-power mode 10b - Enables DCDC in ultra-low-power mode 11b - Enables DCDC in ultra-low-power mode
10 V2MIPI_PD	Power Down MIPI PHY in VDD2 Domain Controls power down of the MIPI PHY. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 V2DSP_PD	Power Down VDD2_DSP Disables VDD2_DSP (HiFi4). Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until after the HiFi4 is powered on. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** VDD2_DSP must be powered down if VDD2_COMP is in DSR mode. 0b - Power on 1b - Power down
8 VNCOM_DSR	Power Switch and DSR for VDDN_COM Enables DSR mode for VDDN_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. You must write 0 to this bit and apply it before entering DPD mode. Write 0 to the bit in either Compute or Sense slot since it is aggregated low. You can write 1 to this bit again after DPD wakeup. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDDN_COM domain 1b - Enables VDDN_COM domain in DSR mode
7 V2COM_DSR	Power Switch and DSR Enable for VDD2_COM **NOTE** Read only. Value used for aggregation. Enables DSR mode for VDD2_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. 0b - Powers on VDD2_COM 1b - Enables DSR mode in VDD2_COM
6 V2NMED_DSR	Power Switch and DSR for VDD2 and VDDN Media Enables DSR mode for VDD2 and VDDN media domains. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	You must write 0 to this bit and apply it before entering DPD mode. Write 0 to the bit in either Compute or Sense slot since it is aggregated low. You can write 1 to this bit again after DPD wakeup. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2 and VDDN media domains 1b - Enables VDD2 and VDDN media domains in DSR mode
5 V2COMP_DSR	Power Switch and DSR Enable in VDD2_COMP Domain **NOTE** Read only. Value used for aggregation. Enables DSR mode for VDD2_COMP domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. 0b - Powers on VDD2_COMP 1b - Enables DSR mode in VDD2_COMP
4 FDPD	Full Deep Power Down (FDPD) Mode **NOTE** Read only. Value used for aggregation. Activates isolation and corrects sequencing for FDPD mode. The chip will only go to FDPD mode if both domains enter Deep Sleep with this bit set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. 0b - Disables 1b - Enables FDPD mode
3 DPD	Deep Power Down (DPD) Mode **NOTE** Read only. Value used for aggregation. Activates isolation and corrects sequencing for DPD mode. The chip will only go to DPD mode if both domains enter Deep Sleep with this bit set, and at least one domain has [FDPD] cleared. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables DPD mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 FDSR	Full Deep Sleep Retention (FDSR) Mode **NOTE** Read only. Value used for aggregation. Puts all power switches into DSR or power down. All core domains go to DSR mode or power down mode depending on domain. Muxes VDDN_SR to VDD2 before ramping down supply voltages and turning off VDDN. This is the only way to put VDD1_SENSE domain into DSR mode and control the associated power switch. The chip will only go to FDSR mode if both domains enter Deep Sleep with this bit set, and [DPD] and [FDPD] are not set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. 0b - All power switch states controlled by their respective PD*CFG bits 1b - All core domains in DSR or Power Down mode
1-0 PMICMODE	PMIC Power Mode Select Sets value of PMIC_MODE output pins. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Drives PMIC_MODE pins to 0h 01b - Drives PMIC_MODE pins to 1h 10b - Drives PMIC_MODE pins to 2h 11b - Drives PMIC_MODE pins to 3h

27.7.1.21 PD Run Configuration 1 in Compute Domain (PDRUNCFG1)

Offset

Register	Offset
PDRUNCFG1	A4h

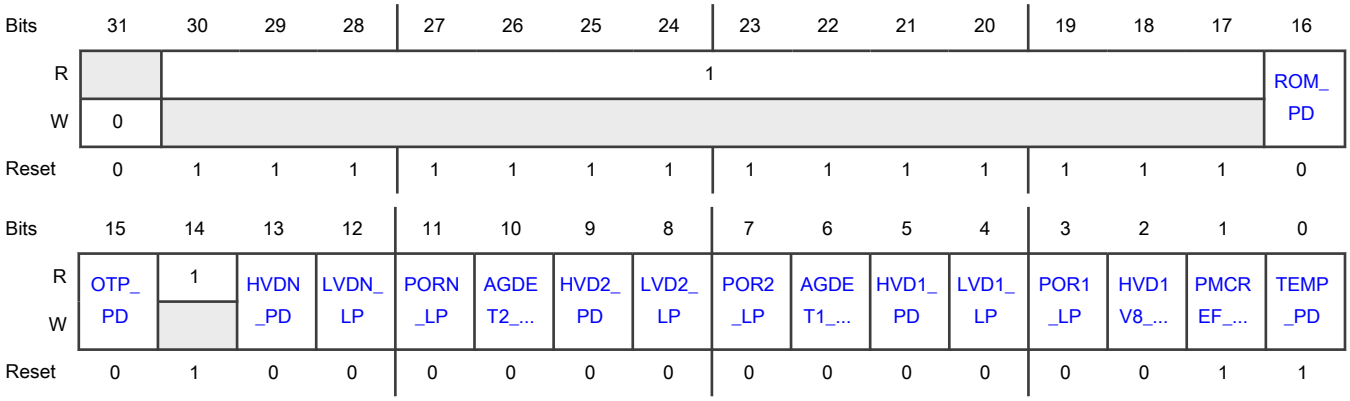
Function

Configures power down for CPU0 in active mode. This drives the aggregation input for the power down and other controls of PMC functions when Compute domain is in active mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Compute domain CTRL[APPLYCFG] is set, triggering the state machines or CPU0 enters and then exits the Deep Sleep mode.

Diagram



Fields

Field	Function
31 —	Reserved
30-17 —	Reserved
16 ROM_PD	Power Down ROM Controls the power switch for the ROM. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 OTP_PD	Power Down OTP Controls the power switch for OTP0 VDD2 Core. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	Power Down HVD in VDDN Domain Controls the power mode of the VDDN high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
12 LVDN_LP	LVD Low-Power Mode in VDDN Domain Sets LVD in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode
11 PORN_LP	POR Low-Power Mode in VDDN Domain Sets POR in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets POR in low-power mode
10 AGDET2_PD	Power Down aGDET in VDD2 Domain Powers down the VDD2 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 HVD2_PD	Power Down HVD in VDD2 Domain Controls the power mode of the VDD2 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
8 LVD2_LP	LVD Low-Power Mode in VDD2 Domain Sets LVD in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode
7 POR2_LP	POR Low-Power Mode in VDD2 Domain Sets POR in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets POR in low-power mode
6	Power Down aGDET in VDD1 Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
AGDET1_PD	Powers down the VDD1 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 HVD1_PD	Power Down HVD in VDD1 Domain Controls the power mode of the VDD1 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
4 LVD1_LP	LVD Low-Power Mode in VDD1 Domain Sets LVD in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode
3 POR1_LP	POR Low-Power Mode in VDD1 Domain Sets POR in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets POR in low-power mode
2 HVD1V8_PD	HVD Power Down Controls the power switch for the VDD1V8 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
1 PMCREF_LP	PMC References in Low-Power Mode Sets PMC references in low-power mode if high power is not needed by any regulators, HVDs, POR, LVD, AFBB, or the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets PMC references in high-power mode 1b - Sets PMC references in low-power mode if not overridden by any *_PD bits
0 TEMP_PD	Power Down PMC Temperature Sensor Controls the power switch for the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down

27.7.1.22 PD Run Configuration 2 in Compute Domain (PDRUNCFG2)

Offset

Register	Offset
PDRUNCFG2	A8h

Function

Configures main memory array and periphery power down for CPU0 in active mode. This drives aggregation input for the power down control of the arrays and peripheries of the main SRAM partitions when CPU0 is in active mode. If a partition's aggregated PD*CFG2[SRAMn] is 0, then the array is powered and the periphery power is controlled by PD*CFG3. If PD*CFG2[SRAMn] is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until Compute domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU0 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-1 SRAMn	Power Down RAM Partition n Array Powers down the array and periphery of the associated memory, regardless of PD*CFG3 state. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Array power on, periphery power controlled by PD*CFG3 1b - Power down
0 —	Reserved

27.7.1.23 PD Run Configuration 3 in Compute Domain (PDRUNCFG3)

Offset

Register	Offset
PDRUNCFG3	ACh

Function

Powers down the selected main memory partitions' periphery logic to save power, while maintaining the contents if the corresponding PD*CFG2[SRAMn] is configured to keep the array on.

This register is reset by cold reset.

NOTE

SRAM0 to SRAM17 are the VDD2_COM main memories. SRAM18 to SRAM29 are the VDD1_SENSE main memories.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down***
- Changes to this register do not affect the memories until Compute domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU0 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-1 SRAMn	Power Down RAM Partition n Periphery Powers down SRAM partition n peripheral. SRAM partition n peripheral is powered on only when the corresponding array SRAM n is powered on. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on when the corresponding array SRAM 1 is 0. 1b - Power down
0 —	Reserved

27.7.1.24 PD Run Configuration 4 in Compute Domain (PDRUNCFG4)

Offset

Register	Offset
PDRUNCFG4	B0h

Function

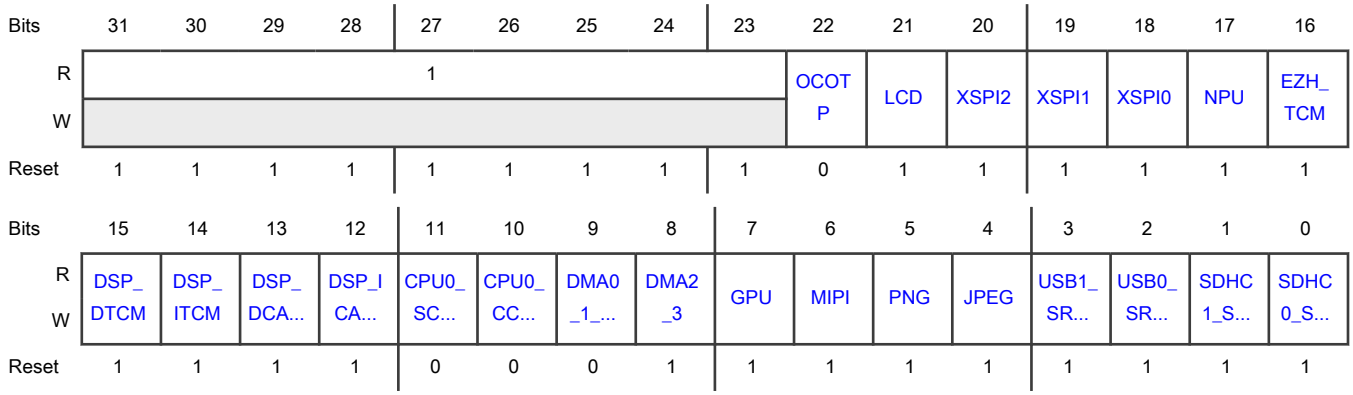
Configures peripheral memory array and peripheral power down for CPU0 in active mode. This drives aggregation input for the power down control of the arrays and peripheries of the Peripherals" SRAM partitions when CPU0 is in active mode. If a partition's aggregated PD*CFG4[PD] bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4[PD] bit is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until Compute domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU0 enters and then exits Deep Sleep mode.

Diagram



Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[LCD] or PDRUNCFG5[LCD] or both are set the LCDIF controller RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2 and MMU2 RAMs array. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs array. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_DTCM] or PDRUNCFG5[DSP_DTCM] or both are set the HiFi4 data TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_ITCM] or PDRUNCFG5[DSP_ITCM] or both are set the HiFi4 instruction TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_DCACHE] or PDRUNCFG5[DSP_DCACHE] or both are set the HiFi4 data cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_ICACHE] or PDRUNCFG5[DSP_ICACHE] or both are set the HiFi4 instruction cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1 and PKC periphery, and ETF RAMs array. This RAM does not support a power down of DMA and PKC array, so PDRUNCFG4 and PDRUNCFG5 both control power down of DMA and PKC periphery. If either PDRUNCFG4[DMA0_1_P_E] or PDRUNCFG5[DMA0_1_P_E] or both are set the DMA0-1 and PKC RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DMA2_3] or PDRUNCFG5[DMA2_3] or both are set the DMA2 and DMA3 RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs array.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.1.25 PD Run Configuration 5 in Compute Domain (PDRUNCFG5)

Offset

Register	Offset
PDRUNCFG5	B4h

Function

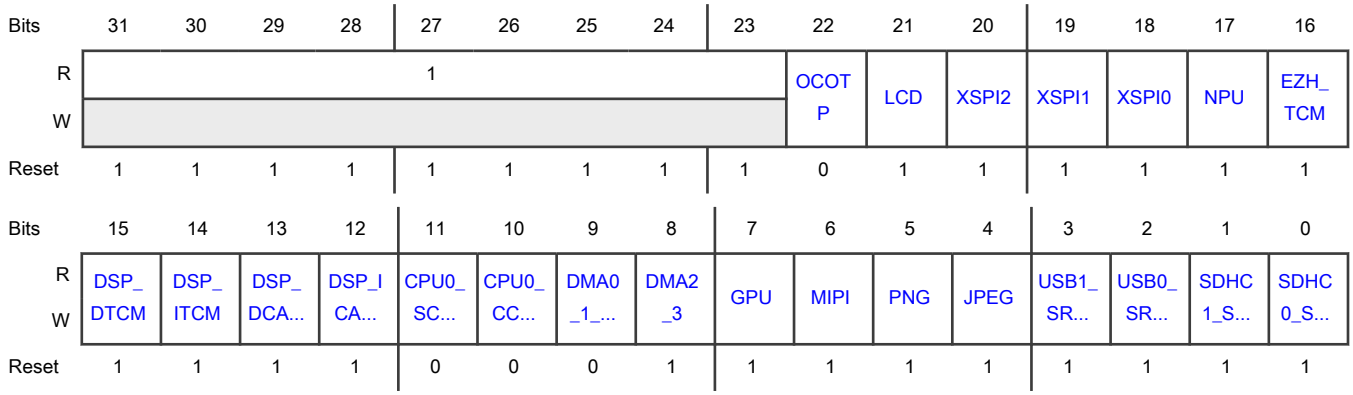
Configures peripheral memory periphery power down for CPU0 in active mode. This drives aggregation input for the power down control of the peripheries of the Peripherals" SRAM partitions when CPU0 is in active mode. If a partition's aggregated PD*CFG4 bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4 bit is 1, then the periphery is powered off and the array power is determined by PD*CFG4. When the power domain of the peripheral in DSR mode, the peripheries for all SRAMs in that domain are automatically powered down.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down***
- Changes to this register do not affect the memories until Compute domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU0 enters and then exits Deep Sleep mode.

Diagram



Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2, MMU2, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1, PKC, and ETF RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.1.26 PD Sleep Configuration 0 in Compute Domain (PDSLEEPCFG0)

Offset

Register	Offset
PDSLEEPCFG0	B8h

Function

Configures power down for CPU0 in Deep Sleep mode. This drives the aggregation input for the power down and other controls of PMC functions when Compute domain is in Deep Sleep mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Compute domain enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W	1	1														
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	V2DSP_PD	VNCOM_D...	V2COM_D...	V2NMED_...	V2COMP_...	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 —	Reserved
29 RBBSR2_PD	Power Down SRAM RBB in VDD2 Domain SRAM RBB should be activated only when no SRAMs in the domain are accessed. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain SRAM RBB should be activated only when no SRAMs in the domain are accessed. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
27 AFBBN_PD	Power Down AFBB in VDDN Domain Controls AFBB mode for VDDN domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
26 RBBN_PD	Power Down RBB in VDDN Domain Controls RBB mode for VDDN domain. RBB mode should only be enabled when VDDN domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBBN_PD] = 1, power on. Else power down 1b - Power down
25 AFBB2_PD	Power Down AFBB in VDD2 Domain Controls AFBB mode for VDD2 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
24 RBB2_PD	Power Down RBB in VDD2 Domain Controls RBB mode for VDD2 domain. RBB mode should only be enabled when VDD2 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBB2_PD] = 1, power on. Else power down 1b - Power down
23 AFBB1_PD	Power Down AFBB in VDD1 Domain Controls AFBB mode for VDD1 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 Domain Controls RBB mode for VDD1 domain. RBB mode should only be enabled when VDD1 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBB1_PD] = 1, power on. Else power down 1b - Power down
21-20 LDO2_VSEL	LDO VDD2 Voltage Select Selects one of four voltages in LDOVDD2VSEL register. Also selects from one of four LVD trip points in LVDVDD2CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level 01b - Selects LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level 10b - Selects LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level 11b - Selects LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level
19-18 LDO2_MODE	LDO VDD2 Regulator Mode Selects regulation mode of LDO VDD2. When high-power mode is selected, the PMC references automatically enter high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - High-power mode 10b,11b - Low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Select Selects one of four voltages in LDOVDD1VSEL register. Also selects from one of four LVD trip points in LVDVDD1CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level 01b - Selects LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level 10b - Selects LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level 11b - Selects LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level
15-14 LDO1_MODE	LDO VDD1 Regulator Mode Selects regulation mode of LDO VDD1. When high-power mode is selected, the PMC references automatically enter high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
13 DCDC_VSEL	DCDC Voltage Select Selects the PMC DCDC voltage and LVD level VDDN supply rail from the DCDCVSEL register. DCDCVSEL[VSEL1] is the higher voltage for aggregation. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Selects DCDCVSEL[VSEL0] level 1b - Selects DCDCVSEL[VSEL1] level
12-11 DCDC_MODE	DCDC Power Mode Selects DCDC power mode. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. This field is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** When using an external supply to power VDDN and POWERCFG[DCDCPD] = 1b, software must also set field DCDC_MODE = 11b in PMC0.PDRUNCFG0, PMC0.PDSLEEPCFG0, PMC1.PDRUNCFG0, and PMC1.PDSLEEPCFG to minimze power when all other PMC subblocks go to low power or power down mode. 00b - Enables DCDC in high-power mode 01b - Enables DCDC in low-power mode 10b - Enables DCDC in ultra-low-power mode 11b - Enables DCDC in ultra-low-power mode
10 V2MIPI_PD	Power Down MIPI PHY in VDD2 Domain Controls power down of the MIPI PHY. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down
9 V2DSP_PD	Power Down VDD2_DSP Disables VDD2_DSP (HiFi4). This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** VDD2_DSP must be powered down if VDD2_COMP is in DSR mode. **NOTE** Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until the HiFi4 is powered. 0b - Power on 1b - Power down
8 VNCOM_DSR	Power Switch and DSR Enable in VDDN_COM Domain Enables DSR mode for VDDN_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDDN_COM 1b - Enables DSR mode in VDDN_COM
7 V2COM_DSR	Power Switch and DSR Enable in VDD2_COM Domain Enables DSR mode for VDD2_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2_COM 1b - Enables DSR mode in VDD2_COM
6 V2NMED_DSR	Power Switch and DSR Enable in VDD2_MEDIA and VDDN_MEDIA Domains Enables DSR mode for VDD2_MEDIA and VDDN_MEDIA domains. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Merged VDD2 and VDDN media switch controls. 0b - Powers on VDD2_MEDIA and VDDN_MEDIA 1b - Enables DSR mode in VDD2_MEDIA and VDDN_MEDIA
5 V2COMP_DSR	Power Switch and DSR Enable in VDD2_COMP Domain Enables DSR mode for VDD2_COMP domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2_COMP 1b - Enables DSR mode in VDD2_COMP
4 FDPD	Full Deep Power Down (FDPD) Mode Activates isolation and corrects sequencing for FDPD mode. The chip will only go to FDPD mode if both domains enter Deep Sleep with this bit set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. 0b - Disables 1b - Enables FDPD mode
3 DPD	Deep Power Down (DPD) Mode Activates isolation and corrects sequencing for DPD mode. The chip will only go to DPD mode if both domains enter Deep Sleep with this bit set, and at least one domain has [FDPD] cleared. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. You must clear this bit whenever PDSLEEPCFG0[FDSR] is set. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Disables 1b - Enables DPD mode
2	Full Deep Sleep Retention (FDSR) Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
FDSR	Puts all power switches into DSR or power down. All core domains go to DSR mode or power down mode depending on domain. Muxes VDDN_SR to VDD2 before ramping down supply voltages and turning off VDDN. This is the only way to put VDD1_SENSE domain into DSR mode and control the associated power switch. The chip will only go to FDSR mode if both domains enter Deep Sleep with this bit set, and [DPD] and [FDPD] are not selected instead. You must clear this bit whenever PDSLEEPCFG0[DPD] is set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - All power switch states controlled by their respective PD*CFG bits 1b - All core domains in DSR or Power Down mode
1-0 PMICMODE	PMIC Mode Sets value of PMIC_MODE output pins. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Drives PMIC_MODE pins to 0h 01b - Drives PMIC_MODE pins to 1h 10b - Drives PMIC_MODE pins to 2h 11b - Drives PMIC_MODE pins to 3h

27.7.1.27 PD Sleep Configuration 1 in Compute Domain (PDSLEEPCFG1)

Offset

Register	Offset
PDSLEEPCFG1	BCh

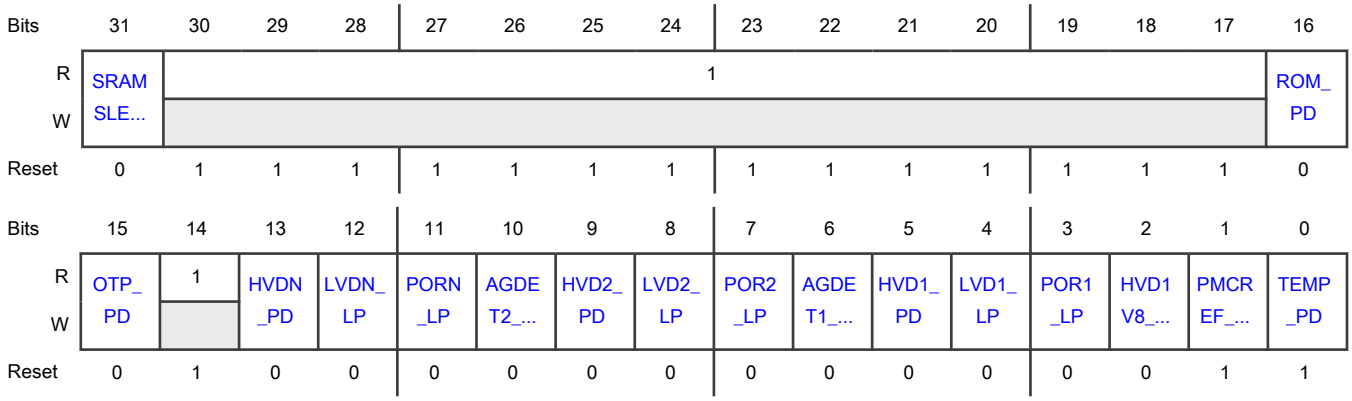
Function

Configures power down for CPU0 in Deep Sleep mode. This drives the aggregation input for the power down and other controls of PMC functions when Compute domain is in Deep Sleep mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Compute domain enters Deep Sleep mode.

Diagram



Fields

Field	Function
31 SRAMSLEEP	SRAM Sleep Mode Sets SRAM sleep mode. This is needed when VDD Core can be < 0.85 V to ensure contents retained. No power savings. Memories are not accessible in RAM sleep mode. It should only be set in PDSLEEPCFG. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Normal 1b - Sleep
30-17 —	Reserved
16 ROM_PD	ROM Power Down Controls the power switch for the ROM. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15	OTP Power Down in VDD2 Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
OTP_PD	Controls the power switch for OTP0 VDD2 Core. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	HVD Power Down in VDDN Domain Controls the power mode of the VDDN high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
12 LVDN_LP	LVD Low power in VDDN Domain Sets LVD in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
11 PORN_LP	POR Low Power in VDDN Domain Sets POR in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
10 AGDET2_PD	aGDET Power Down in VDD2 Domain Powers down the VDD2 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 HVD2_PD	HVD Power Down in VDD2 Domain Controls the power mode of the VDD2 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
8 LVD2_LP	LVD Low power in VDD2 Domain Sets LVD in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 POR2_LP	POR Low power in VDD2 Domain Sets POR in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
6 AGDET1_PD	aGDET Power Down in VDD1 Domain Powers down the VDD1 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 HVD1_PD	HVD Power Down in VDD1 Domain Controls the power mode of the VDD1 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
4 LVD1_LP	LVD Low power in VDD1 Domain Sets LVD in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
3 POR1_LP	POR Low Power in VDD1 Domain Sets POR in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
2 HVD1V8_PD	HVD VDD1V8 Power Down This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
1 PMCREF_LP	PMC References Low Power Sets PMC reference in low-power mode if high power is not needed by any regulators, HVDs, POR, LVD, AFBB, or the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Low-power mode if not overridden by any *_PD bits
0 TEMP_PD	PMC Temperature Sensor Power Down Controls the power switch for the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down

27.7.1.28 PD Sleep Configuration 2 in Compute Domain (PDSLEEPCFG2)

Offset

Register	Offset
PDSLEEPCFG2	C0h

Function

Configures power down for main memory array and periphery for CPU0 in Deep Sleep mode. This drives aggregation input for the power down control of the arrays and peripheries of the min SRAM partitions when CPU0 is in Deep Sleep mode. If a partition's aggregated PD*CFG2[SRAMn] is 0, then the array is powered and the periphery power is controlled by PD*CFG3. If PD*CFG2[SRAMn] is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until CPU0 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Array Power Down This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on RAM Partition 0 array 1b - Powers down RAM Partition 0 array and periphery

27.7.1.29 PD Sleep Configuration 3 in Compute Domain (PDSLEEPCFG3)

Offset

Register	Offset
PDSLEEPCFG3	C4h

Function

Powers down the selected main memory partitions' periphery logic to save power, while maintaining the contents if the corresponding PD*CFG2[SRAMn] is configured to keep the array on.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down*** .
- Changes to this register do not affect the memories until CPU0 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Periphery Power Down Periphery is powered only when the corresponding array SRAMn is powered. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.1.30 PD Sleep Configuration 4 in Compute Domain (PDSLEEPCFG4)

Offset

Register	Offset
PDSLEEPCFG4	C8h

Function

Configures power down for peripheral memory array and periphery for CPU0 in Deep Sleep mode. This drives aggregation input for the power down control of the arrays and peripheries of the peripherals" SRAM partitions when CPU0 is in Deep Sleep mode. If a partition's aggregated PD*CFG4[PD] bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4[PD] bit is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until CPU0 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[LCD] or PDSLEEPCFG5[LCD] or both are set the LCDIF controller RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2 and MMU2 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_DTCM] or PDSLEEPCFG5[DSP_DTCM] or both are set the HiFi4 data TCM RAM periphery will be powered down.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_ITCM] or PDSLEEPCFG5[DSP_ITCM] or both are set the HiFi4 instruction TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_DCACHE] or PDSLEEPCFG5[DSP_DCACHE] or both are set the HiFi4 data cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_ICACHE] or PDSLEEPCFG5[DSP_ICACHE] or both are set the HiFi4 instruction cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1 and PKC periphery, and ETF RAMs array. This RAM does not support a power down of DMA and PKC array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of DMA and PKC periphery. If either PDSLEEPCFG4[DMA0_1_P_E] or PDSLEEPCFG5[DMA0_1_P_E] or both are set the DMA0-1 and PKC RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Powers down DMA2 and DMA3 RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DMA2_3] or PDSLEEPCFG5[DMA2_3] or both are set the DMA2 and DMA3 RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0	uSDHC0 SRAM

Table continues on the next page...

Table continued from the previous page...

Field	Function
SDHC0_SRAM	Powers down uSDHC0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.1.31 PD Sleep Configuration 5 in Compute Domain (PDSLEEPCFG5)

Offset

Register	Offset
PDSLEEPCFG5	CCh

Function

Configures power down for peripheral memory periphery for CPU0 in Deep Sleep mode. This drives aggregation input for the power down control of the peripheries of the Peripherals" SRAM partitions when CPU0 is in Deep Sleep. If a partition's aggregated PD*CFG4 bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4 bit is 1, then the periphery is powered off and the array power is determined by PD*CFG4. When the power domain of the Peripheral in DSR mode the peripheries for all SRAMs in that domain are automatically powered down.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down*** .
- Changes to this register do not affect the memories until CPU0 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2, MMU2, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1, PKC, and ETF RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.1.32 PD Configuration Status 0 (PDCFGSTATUS0)

Offset

Register	Offset
PDCFGSTATUS0	D0h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG0 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep mode), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1	1	RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W																
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	V2DSP_PD	VNCOM_D...	V2COM_D...	V2NMED_...	V2COMP_...	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 RBBSR2_PD	SRAM RBB Power Down in VDD2 Domain Indicates SRAM RBB power down status in VDD2 domain. SRAM RBB should be activated only when no SRAMs in the domain are accessed. This bit is reset by warm reset. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain Indicates SRAM RBB power down status in VDD1 domain. SRAM RBB should be activated only when no SRAMs in the domain are accessed. This bit is reset by warm reset. 0b - Power on 1b - Power down
27	AFBB Power Down in VDDN domain Indicates AFBB power down status in VDDN domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
AFBBN_PD	This bit is reset by warm reset. 0b - Power on 1b - Power down
26 RBBN_PD	RBB Power Down in VDDN domain Indicates RBB power down status in VDDN domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBBN_PD] = 1, otherwise RBB is disabled 1b - Power down
25 AFBB2_PD	AFBB Power Down in VDD2 domain Indicates AFBB power down status in VDD2 domain. This bit is reset by warm reset. 0b - Power 1b - Power down
24 RBB2_PD	RBB Power Down in VDD2 domain Indicates RBB power down status in VDD2 domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBB2_PD] = 1, otherwise RBB is disabled 1b - Power down
23 AFBB1_PD	AFBB Power Down in VDD1 domain Indicates AFBB power down status in VDD1 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 domain Indicates RBB power down status in VDD1 domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBB1_PD] = 1, otherwise RBB is disabled 1b - Power down
21-20 LDO2_VSEL	LDO VDD2 Voltage Selection Indicates the selected level for LDO VDD2 and LVD from the LDOVDD2VSEL and LVDVDD2CTRL registers. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level selected (lowest voltage) 01b - LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level selected 10b - LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level selected 11b - LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level selected (highest voltage)
19-18 LDO2_MODE	LDO VDD2 Regulator Mode Indicates the effective regulation mode for LDO VDD2. This bit is reset by warm reset. 00b - LDO VDD1 is in bypass mode 01b - LDO VDD2 is in high-power mode 10b,11b - LDO VDD2 is in low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Selection Indicates the selected level for LDO VDD1 and LVD from the LDOVDD1VSEL and LVDVDD1CTRL registers. This bit is reset by warm reset. 00b - LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level selected (lowest voltage) 01b - LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level selected 10b - LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level selected 11b - LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level selected (highest voltage)
15-14 LDO1_MODE	LDO VDD1 Regulator Mode Indicates the effective regulation mode for LDO VDD1. This bit is reset by warm reset. 00b - LDO VDD1 is in bypass mode 01b - LDO VDD1 is in high-power mode 10b,11b - LDO VDD1 is in low-power mode
13 DCDC_VSEL	DCDC Voltage Selection Indicates the selection of DCDC voltage and VDDN LVD levels from the DCDCVSEL and LVDVDDNCTRL registers. This bit is reset by warm reset. This bit shows the effective aggregated value based on the CPU0 and CPU1 status and the PDRUNCFG and PDSLEEPCFG registers. 0b - [VSEL0] level selected 1b - [VSEL1] level selected
12-11	DCDC Power Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
DCDC_MODE	Indicates DCDC power mode. This field is reset by warm reset. 00b - DCDC in high-power mode 01b - DCDC in low-power mode 10b - DCDC in ultra-low-power mode 11b - DCDC in ultra-low-power mode
10 V2MIPI_PD	MIPI PHY Power Down Indicates the power down status of MIPI PHY. This bit is reset by warm reset. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down
9 V2DSP_PD	HiFi4 Power Down Indicates the power down status of HiFi4. This bit is reset by warm reset. **NOTE** Must be powered down if VDD2_COMP is in DSR mode. **NOTE** Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until the HiFi4 is powered. 0b - Power on 1b - Power down
8 VNCOM_DSR	DSR of VDDN_COM Domain Indicates the DSR status of VDDN_COM domain. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
7 V2COM_DSR	DSR of VDD2_COM Domain Indicates the DSR status of VDD2_COM domain. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - DSR mode
6 V2NMED_DSR	DSR of VDD2_MEDIA and VDDN_MEDIA Domains Indicates the DSR status of VDD2_MEDIA and VDDN_MEDIA domains. **NOTE** Merged VDD2 and VDDN media switch controls. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
5 V2COMP_DSR	DSR of VDD2_COMP Domain Indicates the DSR status of VDD2_COMP domain. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
4 FDPD	FDPD Mode Indicates FDPD status. This bit is reset by warm reset. 0b - Enables 1b - Activates isolation and correct sequencing for FDPD mode
3 DPD	DPD Mode Indicates DPD status. This bit is reset by warm reset. 0b - Enables 1b - Activates isolation and correct sequencing for DPD mode
2 FDSR	FDSR Mode Indicates the FDSR mode of all core domains. This bit is reset by warm reset. 0b - Power switches and DSR status are determined by the single power domain controls 1b - All core domains are in FDSR mode or power down
1-0 PMICMODE	PMIC_MODE Output Value This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates the aggregated PMIC_MODE output value driven to PMIC_MODE[1:0] pins, which is lowest value (highest power) mode selected by the active PD*CFG registers in Compute and Sense domains. PMIC_MODE output value is used to control external power supplies. 00b - PMIC_MODE[1:0] pins = 00 01b - PMIC_MODE[1:0] pins = 01 10b - PMIC_MODE[1:0] pins = 10 11b - PMIC_MODE[1:0] pins = 11

27.7.1.33 PD Configuration Status 1 (PDCFGSTATUS1)

Offset

Register	Offset
PDCFGSTATUS1	D4h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG1 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SRAM SLE...	1														ROM_ PD
W																
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OTP_ PD	1	HVDN _PD	LVDN_ LP	PORN _LP	AGDE T2_...	HVD2_ PD	LVD2_ LP	POR2 _LP	AGDE T1_...	HVD1_ PD	LVD1_ LP	POR1 _LP	HVD1 V8_...	PMCR EF_...	TEMP _PD
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Function
31	SRAM Sleep Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
SRAMSLEEP	Indicates the status of SRAM Sleep mode. This is needed when VDD Core can be < 0.85 V to ensure contents retained. No power savings. Memories are not accessible in RAM sleep mode. It should only be set in PDSLEEPCFG. This bit is reset by warm reset. 0b - Normal 1b - Sleep
30-17 —	Reserved
16 ROM_PD	ROM Power Down Indicates the power down status of ROM. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 OTP_PD	OTP Power Down Indicates the power down status of OTP VDD2. This bit is reset by warm reset. 0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	HVD Power Down in VDDN Domain Indicates the power down status of the high-voltage detect in VDDN domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 LVDN_LP	LVD Low Power in VDDN Domain Indicates the status of LVD low-power mode in VDDN domain. This bit is reset by warm reset. 0b - High power 1b - Low power
11 PORN_LP	POR Low Power in VDDN Domain Indicates the status of POR low-power mode in VDDN domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This bit is reset by warm reset. 0b - High power 1b - Low power
10 AGDET2_PD	aGDET Power Down in VDD2 Domain Indicates the power down status of the analog glitch detect in VDD2 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 HVD2_PD	HVD Power Down in VDD2 Domain Indicates the power down status of the high-voltage detect in VDD2 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
8 LVD2_LP	LVD Low Power in VDD2 Domain Indicates the status of LVD low-power mode in VDD2 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
7 POR2_LP	POR Low Power in VDD2 Domain Indicates the status of POR low-power mode in VDD2 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
6 AGDET1_PD	aGDET Power Down in VDD1 Domain Indicates the power down status of the analog glitch detect in VDD1 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 HVD1_PD	HVD Power Down in VDD1 Domain Indicates the power down status of the high-voltage detect in VDD1 domain. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
4 LVD1_LP	LVD Low Power in VDD1 Domain Indicates the status of LVD low-power mode in VDD1 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
3 POR1_LP	POR Low Power in VDD1 Domain Indicates the status of POR low-power mode in VDD1 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
2 HVD1V8_PD	HVD Power Down Indicates the power down status of HVD. This bit is reset by warm reset. 0b - Power on 1b - Power down
1 PMCREF_LP	PMC References Low Power Indicates the status of PMC reference low-power mode. This bit is reset by warm reset. 0b - High-power mode 1b - Low-power mode
0 TEMP_PD	PMC Temperature Sensor Power Down Indicates the power down status of PMC temperature sensor. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.1.34 PD Configuration Status 2 (PDCFGSTATUS2)

Offset

Register	Offset
PDCFGSTATUS2	D8h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG2 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Array Power Down Indicates the power down status of RAM Partition n Array. This bit is reset by warm reset. 0b - Array was powered on 1b - Array and periphery were powered down

27.7.1.35 PD Configuration Status 3 (PDCFGSTATUS3)

Offset

Register	Offset
PDCFGSTATUS3	DCh

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG3 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Periphery Power Down This bit is reset by warm reset. Indicates if the SRAM partition n periphery is powered or not. Periphery is powered only when the corresponding array SRAMn is powered. 0b - Power on 1b - Power down

27.7.1.36 PD Configuration Status 4 (PDCFGSTATUS4)

Offset

Register	Offset
PDCFGSTATUS4	E0h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG4 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Indicates the power down status of OCOTP shadow RAMs array. This bit is reset by warm reset. 0b - Array Powered 1b - Array Power Down
21 LCD	LCDIF Controller RAMs Array Indicates the power down status of LCDIF controller RAMs array. This bit is reset by warm reset. 0b - Power on

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Power down
20 XSPI2	XSPI2 and MMU2 RAMs Array Indicates the power down status of XSPI2 and MMU2 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache RAMs Array Indicates the power down status of XSPI1, MMU1, and cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache RAMs Array Indicates the power down status of XSPI0, MMU0, and cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
17 NPU	NPU RAMs Array Indicates the power down status of NPU RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM RAMs Array Indicates the power down status of EZH-V TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM RAMs Array Indicates the power down status of HiFi4 data TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
14 DSP_ITCM	HiFi4 Instruction TCM RAMs Array Indicates the power down status of HiFi4 instruction TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache RAMs Array Indicates the power down status of HiFi4 data cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache RAMs Array Indicates the power down status of HiFi4 instruction cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache RAMs Array Indicates the power down status of CPU0 system cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache RAMs Array Indicates the power down status of CPU0 code cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF RAMs Array Indicates the power down status of DMA0-1, PKC, and ETF RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
8	DMA2 and DMA3 RAMs Array Indicates the power down status of DMA2 and DMA3 RAMs array.

Table continues on the next page...

Table continued from the previous page...

Field	Function
DMA2_3	This bit is reset by warm reset. 0b - Power on 1b - Power down
7 GPU	VGPU RAMs Array Indicates the power down status of VGPU RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY RAMs Array Indicates the power down status of MIPI PHY RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 PNG	PNGDEC RAMs Array Indicates the power down status of PNGDEC RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC RAMs Array Indicates the power down status of JPEGDEC RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 RAMs Array Indicates the power down status of USB1 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 RAMs Array Indicates the power down status of USB0 RAMs array. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 RAMs Array Indicates the power down status of uSDHC1 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 RAMs Array Indicates the power down status of uSDHC0 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.1.37 PD Configuration Status 5 (PDCFGSTATUS5)

Offset

Register	Offset
PDCFGSTATUS5	E4h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG5 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Indicates the power down status of OCOTP shadow RAMs periphery. This bit is reset by warm reset. 0b - Array Powered 1b - Array Power Down
21 LCD	LCDIF Controller RAMs Periphery Indicates the power down status of LCDIF controller RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 RAMs Periphery Indicates the power down status of XSPI2, MMU2, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache RAMs Periphery Indicates the power down status of XSPI1, MMU1, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache RAMs Periphery Indicates the power down status of XSPI0, MMU0, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
17 NPU	NPU RAMs Periphery Indicates the power down status of NPU RAMs periphery. This bit is reset by warm reset. 0b - Power on

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Power down
16 EZH_TCM	EZH-V TCM RAMs Periphery Indicates the power down status of EZH-V TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM RAMs Periphery Indicates the power down status of HiFi4 data TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM RAMs Periphery Indicates the power down status of HiFi4 instruction TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache RAMs Periphery Indicates the power down status of HiFi4 data cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache RAMs Periphery Indicates the power down status of HiFi4 instruction cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache RAMs Periphery Indicates the power down status of CPU0 system cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 CPU0_CCACHE	CPU0 Code Cache RAMs Periphery Indicates the power down status of CPU0 code cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF RAMs Periphery Indicates the power down status of DMA0-1, PKC, and ETF RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 RAMs Periphery Indicates the power down status of DMA2 and DMA3 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
7 GPU	VGPU RAMs Periphery Indicates the power down status of VGPU RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY RAMs Periphery Indicates the power down status of MIPI PHY RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 PNG	PNGDEC RAMs Periphery Indicates the power down status of PNGDEC RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
4	JPEGDEC RAMs Periphery Indicates the power down status of JPEGDEC RAMs periphery.

Table continues on the next page...

Table continued from the previous page...

Field	Function
JPEG	This bit is reset by warm reset. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 RAMs Periphery Indicates the power down status of USB1 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 RAMs Periphery Indicates the power down status of USB0 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 RAMs Periphery Indicates the power down status of uSDHC1 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 RAMs Periphery Indicates the power down status of uSDHC0 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.1.38 PD Wake-up Configuration for Compute Domain (PDWAKECFG)

Offset

Register	Offset
PDWAKECFG	E8h

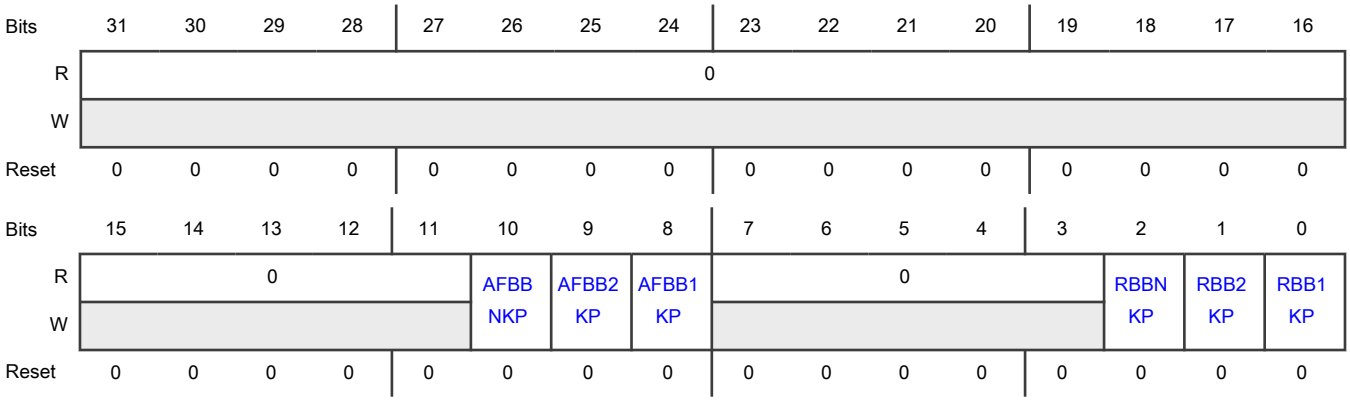
Function

Copies the selected values from the Compute slot PDSLEEPCFG0 to PDRUNCFG0 when CPU0 wakes up from Deep Sleep mode.

Power down wake-up is used when the PMC module switches from one body bias mode to another body bias mode in active mode, but it allows the change to occur while the affected domains are in Deep Sleep mode and the clocks are stopped to avoid timing issues during the transition. The clock controls need to be configured to stop in Deep Sleep mode in the domain's clock controls.

When PMC enters in Deep Sleep mode, the PDSLEEPCFG0 changes the body bias modes and the PDRUNCFG0 body bias power down controls are replaced with the PDSLEEPCFG0 controls selected in PDWAKECFG. This feature is used with INTRCTRL[AUTOWKIE] which enables the AUTOWAKE feature that wakes up PMC when the transition to Deep Sleep is completed and the updated body bias modes are active. The AUTOWAKE triggers a PMC interrupt which triggers a wake-up. Since PDRUNCFG0 was updated with the body bias state from PDSLEEPCFG0, the body bias for the selected domains is maintained.

Diagram



Fields

Field	Function
31-11 —	Reserved
10 AFBBNKP	AFBB Wake-up in VDDN Domain Copies PDSLEEPCFG0[AFBBN_PD] to PDRUNCFG0[AFBBN_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
9 AFBB2KP	AFBB Wake-up in VDD2 Domain Copies PDSLEEPCFG0[AFBB2_PD] to PDRUNCFG0[AFBB2_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
8 AFBB1KP	AFBB Wake-up in VDD1 Domain Copies PDSLEEPCFG0[AFBB1_PD] to PDRUNCFG0[AFBB1_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
7-3 —	Reserved
2 RBBNKP	RBB Wake-up in VDDN Domain Copies PDSLEEPCFG0[RBBN_PD] to PDRUNCFG0[RBBN_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
1 RBB2KP	RBB Wake-up in VDD2 Domain Copies PDSLEEPCFG0[RBB2_PD] to PDRUNCFG0[RBB2_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
0 RBB1KP	RBB Wake-up in VDD1 Domain Copies PDSLEEPCFG0[RBB1_PD] to PDRUNCFG0[RBB1_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep

27.7.1.39 Power Domain Flags for Compute Domain (PWRFLAGS)

Offset

Register	Offset
PWRFLAGS	ECh

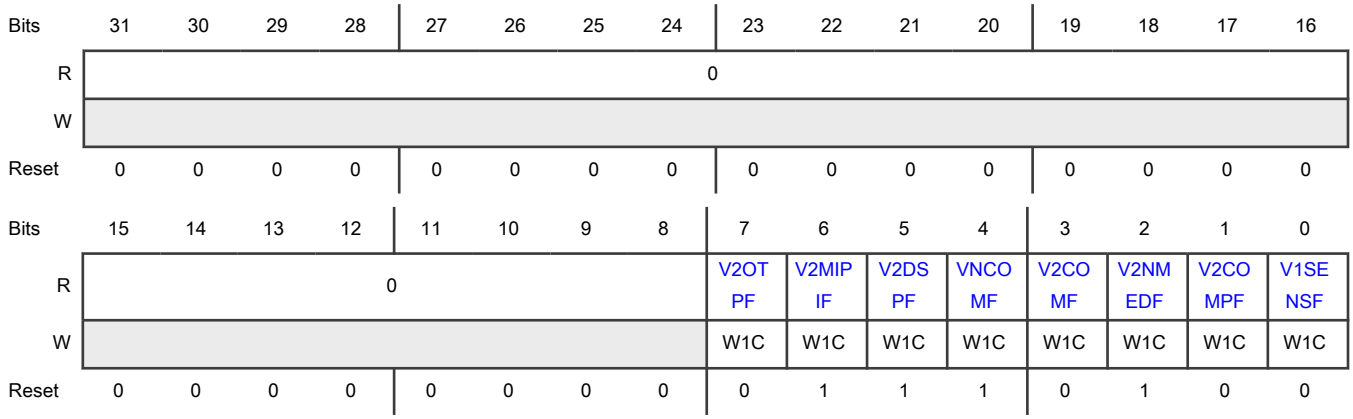
Function

Indicates which domains were put into Deep Sleep Retain mode or Power Down mode since the last time the flag was cleared by software. Flags are clearable only when the domain is powered.

NOTE

The Compute and Sense slots each have their own set of flags. Clearing a flag in the Compute slot will not clear the corresponding flag in the Sense slot.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 V2OTPF	VDD2_OTP Domain Flag Indicates if VDD2_OTP domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
6 V2MIPIF	VDD2_MIPI Domain Flag Indicates if VDD2_MIPI domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
5 V2DSPF	VDD2_DSP Domain Flag Indicates if VDD2_DSP domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
4 VNCOMF	VDDN_COM Domain Flag Indicates if VDDN_COM domain enters DSR mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
3 V2COMF	VDD2_COM Domain Flag Indicates if VDD2_COM domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
2 V2NMEDF	VDD2_MEDIA and VDDN_MEDIA Domain Flag Indicates if VDD2_MEDIA and VDDN_MEDIA domains enter DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
1 V2COMPF	VDD2_COMP Domain Flag Indicates if VDD2_COMP domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
0 V1SENSF	VDD1_SENSE Domain Flag Indicates if VDD1_SENSE domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred

27.7.2 PMC register descriptions

27.7.2.1 PMC register description memory map

This section includes the memory map and detailed description of all registers and their fields of PMC1 in Sense domain.

PMC.PMC1 base address: 4021_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PMC Version and Feature ID (VERID)	32	R	0000_0000h
4h	PMC Status (STATUS)	32	R	0000_0000h
8h	PMC Flags in Sense Domain (FLAGS)	32	RW	001C_F800h
Ch	Control (CTRL)	32	RW	001C_0000h
10h	Interrupt Control in Sense Domain (INTRCTRL)	32	RW	0000_0000h
20h	DCDC Voltage Selection (DCDCVSEL)	32	R	0000_6050h
24h	LDO VDD2 Voltage Selection (LDOVDD2VSEL)	32	R	2C24_1C14h
28h	LDO VDD1 Voltage Selection (LDOVDD1VSEL)	32	RW	2C24_1C14h
2Ch	VDDN LVD Control (LVDVDDNCTRL)	32	R	0000_2C28h
30h	VDD2 LVD Control (LVDVDD2CTRL)	32	R	281E_140Ah
34h	VDD1 LVD Control (LVDVDD1CTRL)	32	RW	281E_140Ah
38h	Power-on Reset Control (PORCTRL)	32	R	0008_0808h
40h	Body Bias Control (BBCTRL)	32	R	0001_0101h
44h	Temperature Sensor Control (TSENSOR)	32	RW	0000_0000h
50h	Wake-up Configuration for Sense Domain (WAKEUP)	32	RW	0000_0000h
64h	Power Configuration (POWERCFG)	32	R	6000_0070h
68h	3 V Capable VDDIO Range Select (PADVRANGE)	32	R	0000_0000h
6Ch	IO Bank State Configuration (PADCFG)	32	RW	0000_0000h
A0h	PD Run Configuration 0 in Sense Domain (PDRUNCFG0)	32	RW	F577_67E0h
A4h	PD Run Configuration 1 in Sense Domain (PDRUNCFG1)	32	RW	7FFF_C003h
A8h	PD Run Configuration 2 in Sense Domain (PDRUNCFG2)	32	RW	FFFF_FFFEh
ACh	PD Run Configuration 3 in Sense Domain (PDRUNCFG3)	32	RW	FFFF_FFFEh
B0h	PD Run Configuration 4 in Sense Domain (PDRUNCFG4)	32	RW	FFBF_FFFFh
B4h	PD Run Configuration 5 in Sense Domain (PDRUNCFG5)	32	RW	FFBF_FFFFh
B8h	PD Sleep Configuration 0 in Sense Domain (PDSLEEPCFG0)	32	RW	F577_67E0h
BCh	PD Sleep Configuration 1 in Sense Domain (PDSLEEPCFG1)	32	RW	7FFF_C003h
C0h	PD Sleep Configuration 2 in Sense Domain (PDSLEEPCFG2)	32	RW	FFFF_FFFEh
C4h	PD Sleep Configuration 3 in Sense Domain (PDSLEEPCFG3)	32	RW	FFFF_FFFEh
C8h	PD Sleep Configuration 4 in Sense Domain (PDSLEEPCFG4)	32	RW	FFBF_FFFFh
CCh	PD Sleep Configuration 5 in Sense Domain (PDSLEEPCFG5)	32	RW	FFBF_FFFFh
D0h	PD Configuration Status 0 (PDCFGSTATUS0)	32	R	F577_6740h
D4h	PD Configuration Status 1 (PDCFGSTATUS1)	32	R	7FFE_4003h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
D8h	PD Configuration Status 2 (PDCFGSTATUS2)	32	R	FFFF_FFFEh
DCh	PD Configuration Status 3 (PDCFGSTATUS3)	32	R	FFFF_FFFEh
E0h	PD Configuration Status 4 (PDCFGSTATUS4)	32	R	FFBF_F1FFh
E4h	PD Configuration Status 5 (PDCFGSTATUS5)	32	R	FFBF_F1FFh
E8h	PD Wake-up Configuration for Sense Domain (PDWAKECFG)	32	RW	0000_0000h
ECh	Power Domain Flags for Sense Domain (PWRFLAGS)	32	RW	0000_0074h

27.7.2.2 PMC Version and Feature ID (VERID)

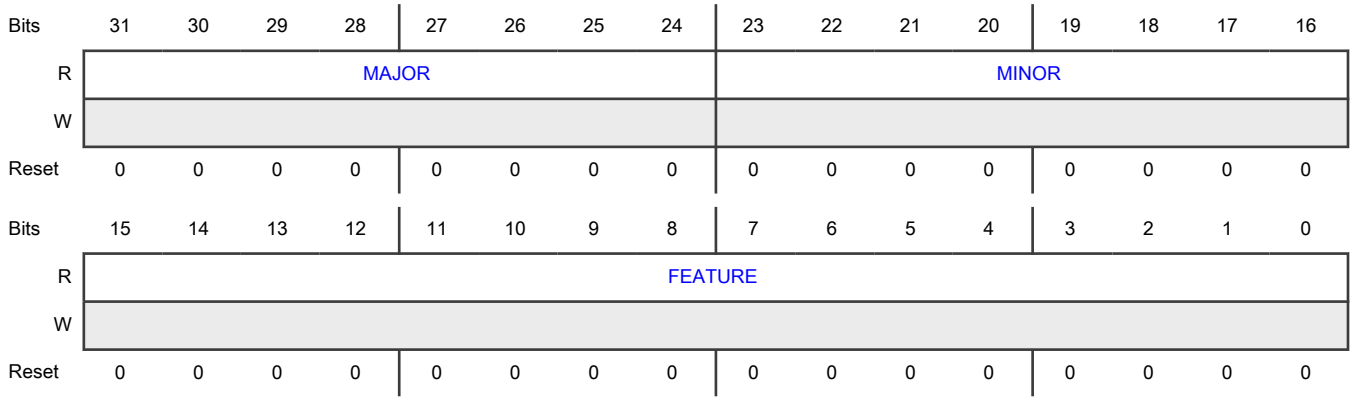
Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the chip.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Reads the major version number for the module specification.
23-16	Minor Version Number

Table continues on the next page...

Table continued from the previous page...

Field	Function
MINOR	Reads the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Reads the feature set number.

27.7.2.3 PMC Status (STATUS)

Offset

Register	Offset
STATUS	4h

Function

Indicates current status of PMC sources, not latched, but synchronized to PMC clock. See PMC Flags for latched versions of sources that are enabled to generate resets or interrupts.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	DSCOMP	0	0	INTN	0	RTCWKUP	HVD1V8	HVDVDDN	HVDVDD2	HVDVDD1	LVDVDDN	LVDVDD2	LVDVDD1	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															BUSY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 DSCOMP	CPU0 Deep Sleep Indicates that CPU0 is in Deep Sleep mode. 0b - CPU0 is in Active mode 1b - CPU0 is in Deep Sleep mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
29 —	Reserved
28 —	Reserved
27 INTN	PMIC_IRQN Interrupt Input Show the interrupt input pin status. 0b - Negated. Pin is high 1b - Asserted. Pin is low
26 —	Reserved
25 RTCWKUP	RTC Wake-up Shows RTC wake-up status. 0b - Sense RTC wake-up is negated 1b - Sense RTC wake-up is asserted
24 HVD1V8	VDD1V8 High-Voltage Detector Shows VDD1V8 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
23 HVDVDDN	VDDN High-Voltage Detector Shows VDDN high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
22 HVDVDD2	VDD2 High-Voltage Detector Shows VDD2 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
21 HVDVDD1	VDD1 High-Voltage Detector Shows VDD1 high-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
20	VDDN Low-Voltage Detector

Table continues on the next page...

Table continued from the previous page...

Field	Function
LVDVDDN	Shows VDDN low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
19 LVDVDD2	VDD2 Low-Voltage Detector Shows VDD2 low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
18 LVDVDD1	VDD1 Low-Voltage Detector Shows VDD1 low-voltage detector status. 0b - Detector is not tripped 1b - Detector is tripped
17-1 —	Reserved
0 BUSY	Finite State Machine Status Shows if PMC is currently processing a mode transition. 0b - PMC state machines are idle 1b - PMC state machine is busy

27.7.2.4 PMC Flags in Sense Domain (FLAGS)

Offset

Register	Offset
FLAGS	8h

Function

Indicates PMC event flags in Sense domain. This register is cleared by software, and set by hardware when event occurs. The flags in Sense and Compute domains are cleared respectively by writing 1 to corresponding bit fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEEPPDF	DSCOMPF	DSSENSF	RESETNF	INTNF	AUTOWKF	RTCF	HVD1V8F	HVDVDDNF	HVDVDD2F	HVDVDD1F	LVDVDDNF	LVDVDD2F	LVDVDD1F	AGDET2F	AGDET1F
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PORAO18F	POR1V8F	PORVDDNF	PORVDD2F	PORVDD1F	DCLPOCF	DCHPOCF	SCP2OCF	SCP1OCF	BBNEF	BB2EF	BB1EF	BBSR2EF	BBSR1EF	ITRCRF	DSALLF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DEEPPDF	DPD Wake-up Flag Indicates the wake-up status of PMC in DPD mode. This bit is reset by VDD1V8_AO POR. 0b - DPD wake-up cleared 1b - DPD wake-up asserted
30 DSCOMPF	CPU0 Deep Sleep Flag Indicates that CPU0 entered Deep Sleep mode. This bit is set at Deep Sleep mode entry, and can be cleared while CPU0 is still in Deep Sleep mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU0 Deep Sleep cleared 1b - CPU0 Deep Sleep asserted
29 DSSENSF	CPU1 Deep Sleep Mode Flag Indicates that CPU1 entered Deep Sleep mode. This bit is set at Deep Sleep mode entry. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU1 Deep Sleep cleared 1b - CPU1 Deep Sleep asserted
28 RESETNF	Reset Pin Flag Indicates reset pin status. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Reset pin status cleared 1b - Reset pin status asserted

Table continues on the next page...

Table continued from the previous page...

Field	Function
27 INTNF	PMIC_IRQN Interrupt Pin Flag Indicates interrupt pin status. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Interrupt pin status cleared 1b - Interrupt pin status asserted
26 AUTOWKF	Auto Wake-up Flag Indicates auto wake-up interrupt status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Auto wake-up status cleared 1b - Auto wake-up status asserted
25 RTCF	RTC Flag Indicates RTC alarm or wake was asserted. This bit is reset by VDD1V8_AO POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - RTC status cleared 1b - RTC status asserted
24 HVD1V8F	VDD1V8 High-Voltage Detector Flag Indicates VDD1V8 high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
23 HVDVDDNF	VDDN High-Voltage Detector Flag Indicates VDDN high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
22 HVDVDD2F	VDD2 High-Voltage Detector Flag Indicates VDD2 high-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
21	VDD1 High-Voltage Detector Flag Indicates VDD1 high-voltage detector status. This bit is reset by VDD1V8 POR.

Table continues on the next page...

Table continued from the previous page...

Field	Function
HVDVDD1F	This field has two copies separately accessed by the Compute and Sense slots. 0b - High-voltage detector status cleared 1b - High-voltage detector status asserted
20 LVDVDDNF	VDDN Low-Voltage Detector Flag Indicates VDDN low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
19 LVDVDD2F	VDD2 Low-Voltage Detector Flag Indicates VDD2 low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
18 LVDVDD1F	VDD1 Low-Voltage Detector Flag Indicates VDD1 low-voltage detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Low-voltage detector status cleared 1b - Low-voltage detector status asserted
17 AGDET2F	VDD2 Glitch Detector Flag Indicates VDD2 glitch detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Glitch detector status cleared 1b - Glitch detector status asserted
16 AGDET1F	VDD1 Glitch Detector Flag Indicates VDD1 glitch detector status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Glitch detector status cleared 1b - Glitch detector status asserted
15 PORAO18F	VDD1V8_AO Power-on Reset Flag Indicates VDD1V8_AO power-on reset status. This bit is set by VDD1V8_AO POR. 0b - POR status cleared 1b - POR status asserted

Table continues on the next page...

Table continued from the previous page...

Field	Function
14 POR1V8F	VDD1V8 Power-on Reset Flag Indicates VDD1V8 power-on reset status. This bit is set by VDD1V8 POR. 0b - POR status cleared 1b - POR status asserted
13 PORVDDNF	VDDN Power-on Reset Flag Indicates VDDN power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
12 PORVDD2F	VDD2 Power-on Reset Flag Indicates VDD2 power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
11 PORVDD1F	VDD1 Power-on Reset Flag Indicates VDD1 power-on reset status. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - POR status cleared 1b - POR status asserted
10 DCLPOCF	DCDC Low-Power Mode Overcurrent Flag Indicates the DCDC had an overcurrent event while in low-power mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
9 DCHPOCF	DCDC High-Power Mode Overcurrent Flag Indicates the DCDC had an overcurrent event while in high-power mode. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
8	SCPC VDD2 Overcurrent Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
SCP2OCF	Indicates that an SCPC2 overcurrent event was detected. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
7 SCP1OCF	SCPC VDD1 Overcurrent Flag Indicates that an SCPC1 overcurrent event was detected. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No overcurrent event detected 1b - Overcurrent event detected
6 BBNEF	VDDN Body Bias Error Flag Indicates body bias error in VDDN domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
5 BB2EF	VDD2 Body Bias Error Flag Indicates body bias error in VDD2 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
4 BB1EF	VDD1 Body Bias Error Flag Indicates body bias error in VDD1 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No body bias error 1b - Body bias error detected
3 BBSR2EF	VDD2 SRAM Body Bias Error Flag Indicates SRAM body bias error in VDD2 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - No SRAM body bias error 1b - SRAM body bias error detected
2 BBSR1EF	VDD1 SRAM Body Bias Error Flag Indicates SRAM body bias error in VDD1 domain. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No SRAM body bias error 1b - SRAM body bias error detected
1 ITRCRF	ITRC RAM ZEROIZE Flag Indicates ITRC triggered RAM ZEROIZE. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - Not detected 1b - ITRC RAM ZEROIZE detected
0 DSALLF	Deep Sleep Flag CPU0 and CPU1 are in Deep Sleep mode at the same time. This bit is reset by VDD1V8 POR. This field has two copies separately accessed by the Compute and Sense slots. 0b - CPU0 and CPU1 are not in Deep Sleep mode at the same time 1b - CPU0 and CPU1 have both been in Deep Sleep mode at the same time

27.7.2.5 Control (CTRL)

Offset

Register	Offset
CTRL	Ch

Function

Enables controls for triggering configuration updates and indicates voltage monitor resets.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							HVD1V8RE	HVDVNRE	HVDV2RE	HVDV1RE	LVDNRE	LVD2RE	LVD1RE	AGDET2RE	AGDET1RE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															APPLYCFG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-25 —	Reserved
24 HVD1V8RE	VDD1V8 High-Voltage Detector Reset Enable Indicates reset for VDD1V8 high-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
23 HVDVNRE	VDDN High-Voltage Detector Reset Enable Indicates reset for VDDN high-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
22 HVDV2RE	VDD2 High-Voltage Detector Reset Enable Indicates reset for VDD2 high-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
21 HVDV1RE	VDD1 High-Voltage Detector Reset Enable Indicates reset for VDD1 high-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
20 LVDNRE	VDDN Low-Voltage Detector Reset Enable Indicates reset for VDDN low-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
19 LVD2RE	VDD2 Low-Voltage Detector Reset Enable Indicates reset for VDD2 low-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
18	VDD1 Low-Voltage Detector Reset Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LVD1RE	Indicates reset for VDD1 low-voltage detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
17 AGDET2RE	VDD2 Glitch Detector Reset Enable Indicates reset for VDD2 glitch detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
16 AGDET1RE	VDD1 Glitch Detector Reset Enable Indicates reset for VDD1 glitch detector. The Compute slot version of this field is read-write. This bit is reset by cold reset. 0b - Disabled 1b - Enabled
15-1 —	Reserved
0 APPLYCFG	Apply Configure for PMC1 Applies updated PDRUNCFG bits (SRAM power gates, RBB, AFBB, LVD, and HVD control bits) and/or RUNCTRL setting. Automatically clears when operation is done. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Operation is done 1b - Start configuration change, or operation has not completed

27.7.2.6 Interrupt Control in Sense Domain (INTRCTRL)

Offset

Register	Offset
INTRCTRL	10h

Function
Enables interrupt control for PMC interrupt sources.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	DSCOMPIE		0	INTNIE	AUTOWKIE	0	HVD1V8IE	HVDNIE	HVD2IE	HVD1IE	LVDNIE	LVD2IE	LVD1IE	AGDET2IE	AGDET1IE
W			0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DCLPOCIE	DCHPOCIE	SCP2OCIE	SCP1OCIE	BBNEIE	BB2EIE	BB1EIE	BBSR2EIE	BBSR1EIE	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 DSCOMPIE	CPU0 Deep Sleep Interrupt Enable Enables interrupt for CPU0 entered into Deep Sleep mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
29 —	Reserved
28 —	Reserved
27 INTNIE	PMIC_IRQN PAD Interrupt Enable Enables PMIC_IRQN PAD interrupt. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
26 AUTOWKIE	Auto Wake-up Interrupt Enable Enables auto wake-up interrupt. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 —	Reserved
24 HVD1V8IE	VDD1V8 High-Voltage Detector Interrupt Enable Enables interrupt for VDD1V8 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
23 HVDNIE	VDDN High-Voltage Detector Interrupt Enable Enables interrupt for VDDN high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
22 HVD2IE	VDD2 High-Voltage Detector Interrupt Enable Enables interrupt for VDD2 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
21 HVD1IE	VDD1 High-Voltage Detector Interrupt Enable Enables interrupt for VDD1 high-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
20 LVDNIE	VDDN Low-Voltage Detector Interrupt Enable Enables interrupt for VDDN low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
19 LVD2IE	VDD2 Low-Voltage Detector Interrupt Enable Enables interrupt for VDD2 low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
18 LVD1IE	VDD1 Low-Voltage Detector Interrupt Enable Enables interrupt for VDD1 low-voltage detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
17 AGDET2IE	VDD2 Glitch Detector Interrupt Enable Enables interrupt for VDD2 glitch detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
16 AGDET1IE	VDD1 Glitch Detector Interrupt Enable Enables interrupt for VDD1 glitch detector. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
15-11 —	Reserved
10 DCLPOCIE	DCDC Low-Power Overcurrent Interrupt Enable Enables interrupt for DCDC overcurrent in low-power mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
9 DCHPOCIE	DCDC High-Power Overcurrent Interrupt Enable Enables interrupt for DCDC overcurrent in high-power mode. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
8 SCP2OCIE	SCPC VDD2 Overcurrent Interrupt Enable Enables interrupt for SCPC overcurrent in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
7 SCP1OCIE	SCPC VDD1 Overcurrent Interrupt Enable Enables interrupt for SCPC overcurrent in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
6 BBNEIE	VDDN Body Bias Error Interrupt Enable Enables interrupt for body bias error in VDDN domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
5 BB2EIE	VDD2 Body Bias Error Interrupt Enable Enables interrupt for body bias error in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
4 BB1EIE	VDD1 Body Bias Error Interrupt Enable Enables interrupt for body bias error in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
3 BBSR2EIE	VDD2 SRAM Body Bias Error Interrupt Enable Enables interrupt for SRAM body bias error in VDD2 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables
2 BBSR1EIE	VDD1 SRAM Body Bias Error Interrupt Enable Enables interrupt for SRAM body bias error in VDD1 domain. This bit is reset by cold reset. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
1-0 —	Reserved

27.7.2.7 DCDC Voltage Selection (DCDCVSEL)

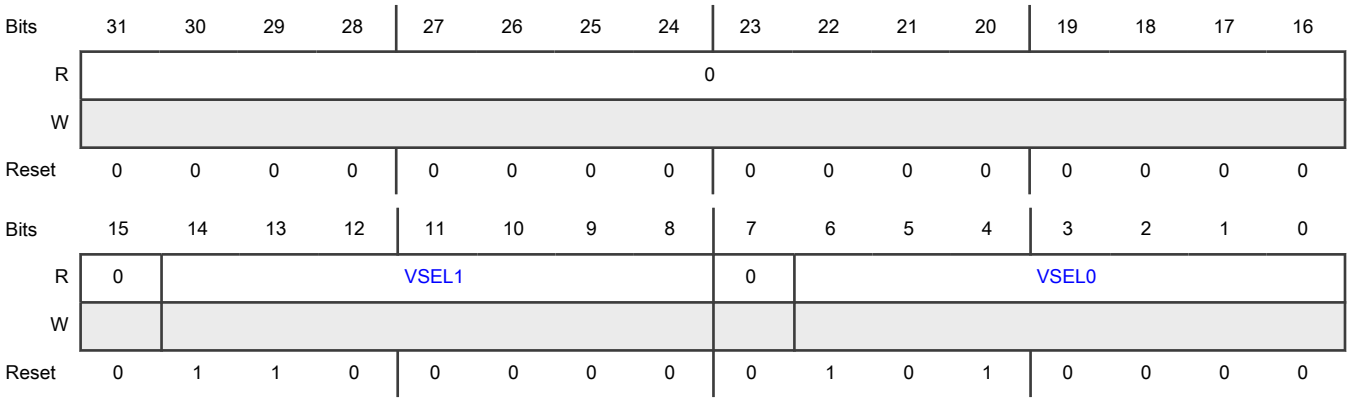
Offset

Register	Offset
DCDCVSEL	20h

Function

Indicates DCDC voltage levels for selection by PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL].

Diagram



Fields

Field	Function
31-15 —	Reserved
14-8 VSEL1	DCDC Voltage Level 1 Indicates the DCDC voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 1. This is the level selected by PD*CFG after reset, where • Voltage = 0.5 V + 6.25 mV * value of [VSEL1]

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• Reset value = 1.1 V • Written value is clipped to 68h • Maximum value = 1.15 V The Compute slot version of this field is read-write. This bit is reset by warm reset. 000_0000b - 0.50 V 010_0000b - 0.70 V 100_0000b - 0.90 V 110_0000b - 1.10 V 110_1000b - 1.15 V (maximum) 111_1111b - 1.15 V (clipped)
7 —	Reserved
6-0 VSEL0	DCDC Voltage Level 0 Indicates the DCDC voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 0. • Voltage = 0.5 V + 6.25 mV * value of [VSEL0] • Reset value = 1.0 V • Written value is clipped to 68h • Maximum value = 1.15 V The Compute slot version of this field is read-write. This bit is reset by warm reset. 000_0000b - 0.50 V 010_0000b - 0.70 V 100_0000b - 0.90 V 110_0000b - 1.10 V 110_1000b - 1.15 V (maximum) 111_1111b - 1.15 V (clipped)

27.7.2.8 LDO VDD2 Voltage Selection (LDOVDD2VSEL)

Offset

Register	Offset
LDOVDD2VSEL	24h

Function

Indicates LDO VDD2 voltage levels for selection by PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		VSEL3						0		VSEL2					
W																
Reset	0	0	1	0	1	1	0	0	0	0	1	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		VSEL1						0		VSEL0					
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0

Fields

Field	Function
31-30 —	Reserved
29-24 VSEL3	LDO VDD2 Voltage Level 3 Indicates the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL3] • Reset value = 1.0 V • Written value is clipped to 38h The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
23-22 —	Reserved
21-16 VSEL2	LDO VDD2 Voltage Level 2

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 2. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL2] • Reset value = 0.9 V • Written value is clipped to 38h The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
15-14 —	Reserved
13-8 VSEL1	LDO VDD2 Voltage Level 1 Indicates the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 1. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL1] • Reset value = 0.8 V • Written value is clipped to 38h The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
7-6	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
5-0 VSEL0	LDO VDD2 Voltage Level 0 Indicates the LDO voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 0. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL0] • Reset value = 0.7 V • Written value is clipped to 38h The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)

27.7.2.9 LDO VDD1 Voltage Selection (LDOVDD1VSEL)

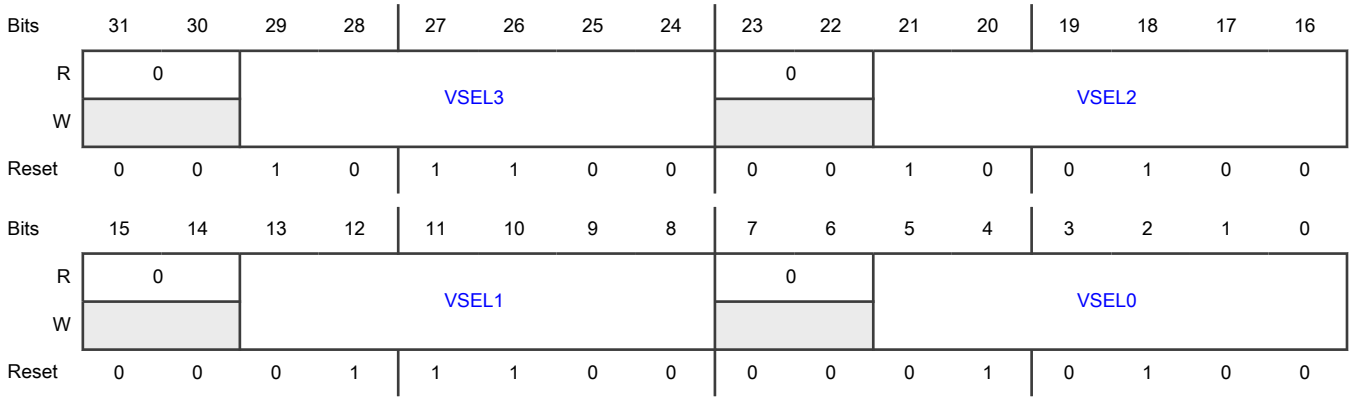
Offset

Register	Offset
LDOVDD1VSEL	28h

Function

Configures LDO VDD1 voltage levels for selection by PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL]. Fields of this register are shared between PMC0 and PMC1. Both Compute and Sense slots can write to this register.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 VSEL3	LDO VDD1 Voltage Level 3 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL3] • Reset value = 1.0 V • Written value is clipped to 38h **NOTE** The value of [VSEL3] must be set larger than that of [VSEL2]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
23-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
21-16 VSEL2	LDO VDD1 Voltage Level 2 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 2. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL2] • Reset value = 0.9 V • Written value is clipped to 38h **NOTE** The value of [VSEL2] must be set larger than that of [VSEL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
15-14 —	Reserved
13-8 VSEL1	LDO VDD1 Voltage Level 1 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 1. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL1] • Reset value = 0.8 V • Written value is clipped to 38h **NOTE** The value of [VSEL1] must be set larger than that of [VSEL0]. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)
7-6 —	Reserved
5-0 VSEL0	LDO VDD1 Voltage Level 0 Sets the LDO voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 0. • Value of LDO = 0.45 V + 12.5 mV * value of [VSEL0] • Reset value = 0.7 V • Written value is clipped to 38h **NOTE** The value of [VSEL0] must be set smaller than that of [VSEL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.45 V 01_0000b - 0.65 V 10_0000b - 0.85 V 11_0000b - 1.05 V 11_1000b - 1.15 V (maximum) 11_1111b - 1.15 V (clipped)

27.7.2.10 VDDN LVD Control (LVDVDDNCTRL)

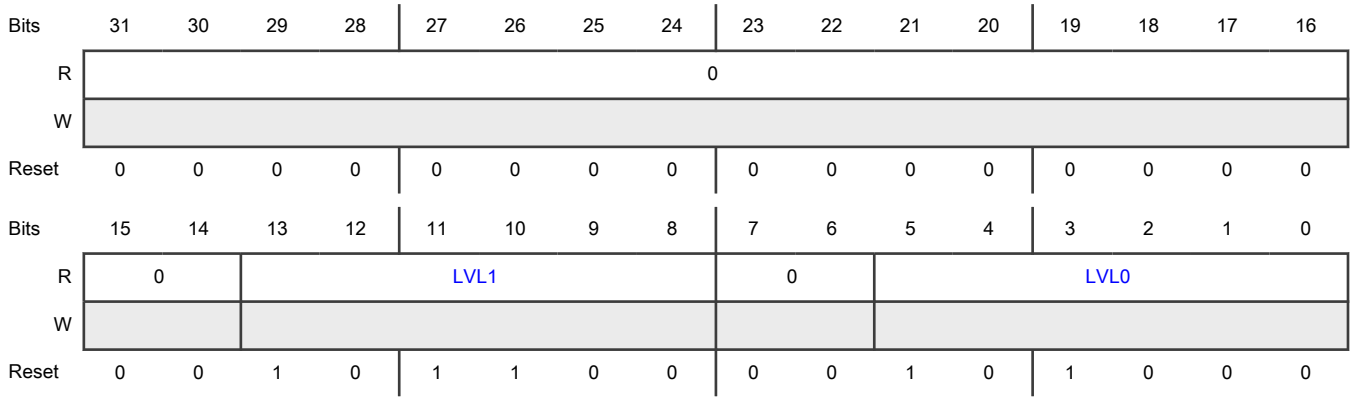
Offset

Register	Offset
LVDVDDNCTRL	2Ch

Function

Indicates the VDDN LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL].

Diagram



Fields

Field	Function
31-14 —	Reserved
13-8 LVL1	LVD Level 1 Voltage Indicates the VDDN LVD falling trip voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 1. This is the level selected by PD*CFG after reset, where: • Falling trip = 0.5 + 10 mV * value of [LVL1] • Reset value = 0.94 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
7-6 —	Reserved
5-0 LVL0	LVD Level 0 Voltage Indicates the VDDN LVD falling trip voltage when the aggregated value of PDRUNCFG0[DCDC_VSEL] and PDSLEEPCFG0[DCDC_VSEL] is 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• Falling trip = 0.5 + 10 mV * value of [LVL0] • Reset value = 0.9 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.2.11 VDD2 LVD Control (LVDVDD2CTRL)

Offset

Register	Offset
LVDVDD2CTRL	30h

Function

Indicates the VDD2 LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		LVL3						0		LVL2					
W																
Reset	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		LVL1						0		LVL0					
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0

Fields

Field	Function
31-30 —	Reserved
29-24 LVL3	LVD Level 3 Voltage Indicates the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Falling trip = 0.5 + 10 mV * value of [LVL3] • Reset value = 0.9 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
23-22 —	Reserved
21-16 LVL2	LVD Level 2 Voltage Indicates the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 2. • Falling trip = 0.5 + 10 mV * value of [LVL2] • Reset value = 0.8 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-14 —	Reserved
13-8 LVL1	LVD Level 1 Voltage Indicates the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 1. • Falling trip = 0.5 + 10 mV * value of [LVL1] • Reset value = 0.7 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
7-6 —	Reserved
5-0 LVL0	LVD Level 0 Voltage Indicates the VDD2 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO2_VSEL] and PDSLEEPCFG0[LDO2_VSEL] is 0. • Falling trip = 0.5 + 10 mV * value of [LVL0] • Reset value = 0.6 V The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.2.12 VDD1 LVD Control (LVDVDD1CTRL)

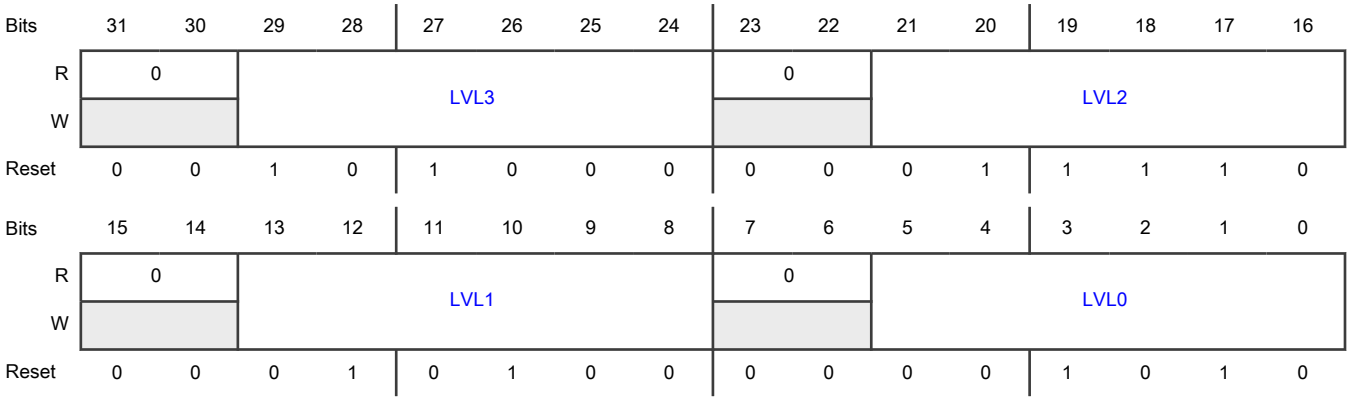
Offset

Register	Offset
LVDVDD1CTRL	34h

Function

Sets the VDD1 LVD falling trip voltages selectable by the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL].

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 LVL3	Level 3 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 3. This is the level selected by PD*CFG after reset, where • Falling trip = 0.5 + 10 mV * value of [LVL3] **NOTE** The value of [LVL3] must be set larger than that of [LVL2]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
23-22 —	Reserved
21-16 LVL2	Level 2 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 2. • Falling trip = 0.5 + 10 mV * value of [LVL2] **NOTE** The value of [LVL2] must be set larger than that of [LVL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V
15-14 —	Reserved
13-8 LVL1	Level 1 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 1. • Falling trip = 0.5 + 10 mV * value of [LVL1] **NOTE** The value of [LVL1] must be set larger than that of [LVL0]. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.
7-6 —	Reserved
5-0 LVL0	Level 0 Voltage Sets the VDD1 LVD falling trip voltage when the aggregated value of PDRUNCFG0[LDO1_VSEL] and PDSLEEPCFG0[LDO1_VSEL] is 0. • Falling trip = 0.5 + 10 mV * value of [LVL0] **NOTE** The value of [LVL0] must be set smaller than that of [LVL1]. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00_0000b - 0.50 V 01_0000b - 0.66 V 10_0000b - 0.82 V 11_0000b - 0.98 V 11_1111b - 1.13 V

27.7.2.13 Power-on Reset Control (PORCTRL)

Offset

Register	Offset
PORCTRL	38h

Function

Indicates the POR falling trip voltages in VDD1, VDD2, and VDDN domains. When the supply voltage falls below the selected trip voltage, it triggers a cold reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												VDDNLVL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			VDD2LVL					0			VDD1LVL				
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Function
31-21 —	Reserved
20-16 VDDNLVL	POR Level in VDDN Domain Indicates the POR falling trip voltage in VDDN domain. The Compute slot version of this field is read-write. • Falling trip = 0.4 + 10 mV * value of [VDDNLVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V 0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V
15-13 —	Reserved
12-8 VDD2LVL	POR Level in VDD2 Domain Indicates the POR falling trip voltage in VDD2 domain. The Compute slot version of this field is read-write. • Falling trip = 0.4 + 10 mV * value of [VDD2LVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V
7-5 —	Reserved
4-0 VDD1LVL	POR Level in VDD1 Domain Indicates the POR falling trip voltage in VDD1 domain. The Compute slot version of this field is read-write. • Falling trip = 0.4 + 10 mV * value of [VDD1LVL] • Reset value = 0.48 V This field is reset by cold reset. 0_0000b - 0.40 V 0_1000b - 0.48 V 1_0000b - 0.56 V 1_1000b - 0.64 V 1_1111b - 0.71 V

27.7.2.14 Body Bias Control (BBCTRL)

Offset

Register	Offset
BBCTRL	40h

Function

Indicates the configuration of the body bias blocks when they are enabled by PD*CFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						0	0	0							RBBNLVL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							RBB2LVL	0							RBB1LVL
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Fields

Field	Function
31-26 —	Reserved
25 —	Reserved
24 —	Reserved
23-17 —	Reserved
16 RBBNLVL	Reverse Body Bias Voltage for VDDN Domain Indicates RBB Voltage for VDDN domain. The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V
15-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 RBB2LVL	Reverse Body Bias Voltage for VDD2 Domain Indicates RBB Voltage for VDD2 domain. The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V
7-1 —	Reserved
0 RBB1LVL	Reverse Body Bias Voltage for VDD1 Domain Indicates RBB Voltage for VDD1 domain. The Compute slot version of this field is read-write. This bit is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Value change only takes affect after setting CTRL[APPLYCFG] or a mode change occurs. 0b - Voltage is set to 1.0 V 1b - Voltage is set to 1.3 V

27.7.2.15 Temperature Sensor Control (TSENSOR)

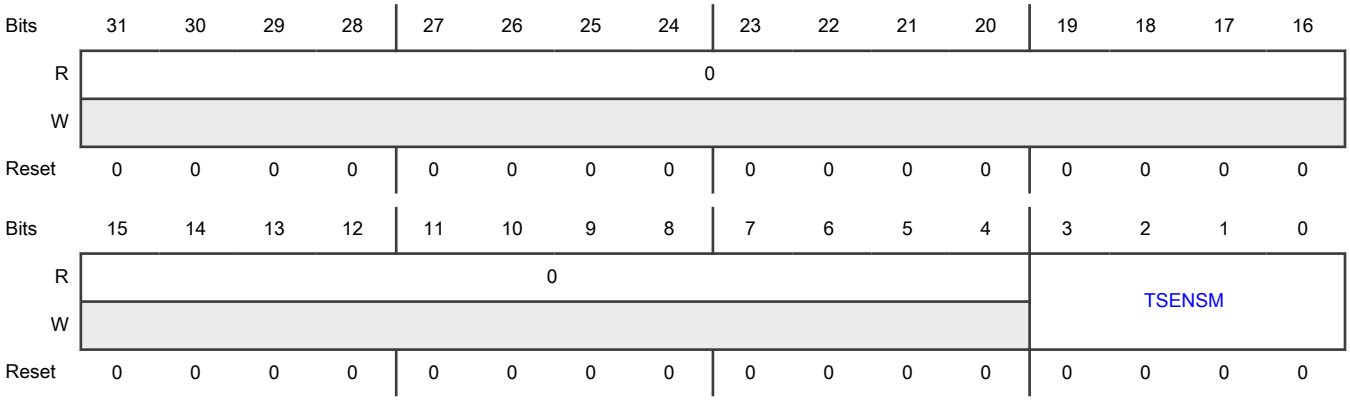
Offset

Register	Offset
TSENSOR	44h

Function

Controls the PMC temperature sensor measurement mode. See PMC temperature sensor for details.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TSENSM	Temperature Sensor Mode Select Selects the temperature sensor mode according to the procedure described in PMC temperature sensor. All modes must be selected in sequence during the temperature measurement procedure to dynamically cancel various internal error sources. **NOTE** This field is shared between PMC0 and PMC1. Software should only do temperature measurement from one slot at a time. 0000b - Perform AD conversion and record C*_000 0001b - Perform AD conversion and record C*_001 0010b - Perform AD conversion and record C*_010 0011b - Perform AD conversion and record C*_011 0100b - Perform AD conversion and record CM_100 0101b - Perform AD conversion and record C*_101 0110b - Perform AD conversion and record C*_110 0111b - Perform AD conversion and record C*_111

27.7.2.16 Wake-up Configuration for Sense Domain (WAKEUP)

Offset

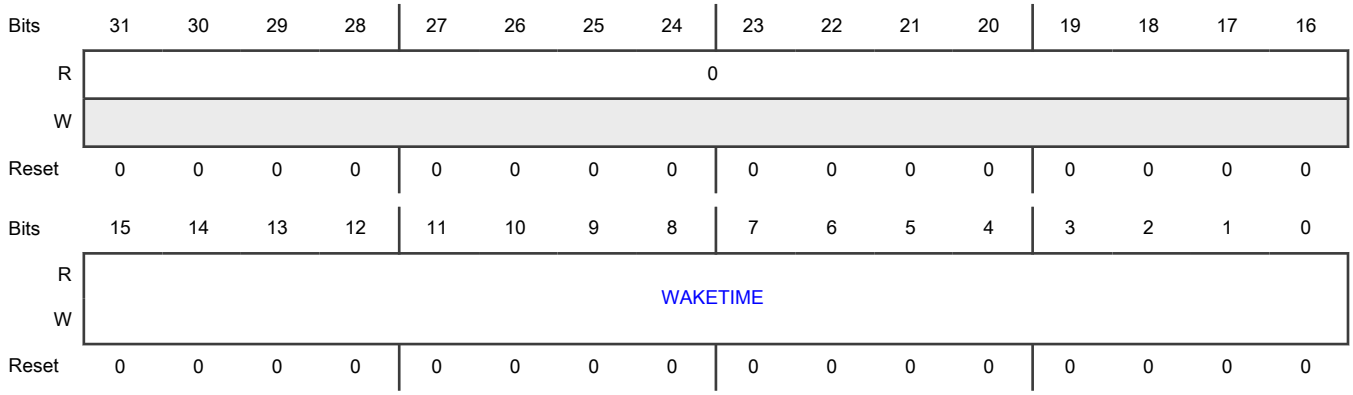
Register	Offset
WAKEUP	50h

Function

Sets the configuration of the general purpose wake-up counter.

The wake-up timer is enabled by INTRCTRL[AUTOWKIE], and will wake up the associated PMC slot when the associated CPU enters Deep Sleep mode. When the counter times out, PMC wakes up and generates an interrupt for that slot.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 WAKETIME	Wake-up Timer Sets timer to wake up from Deep Sleep mode after mode sequencing is completed. Time = value of [WAKETIME] / 16 MHz = value of [WAKETIME] / 16 in µs This bit is reset by warm reset. 0000_0000_0000_0000b - 0 0000_0000_0000_0001b - 62.5 ns = 1 / (16e + 6) 0000_0000_0000_0010b - 125 ns = 2 / (16e + 6) ...

27.7.2.17 Power Configuration (POWERCFG)

Offset

Register	Offset
POWERCFG	64h

Function

Indicates internal regulators and internal references during FDSR mode, FDPD modes, or when external supplies are used. Also configures the wait time for PMIC or external supplies to respond to changes on the PMIC_MODE pins.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MODEDLY				0			FDPD BGLP	0							
W																
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			DCDC PD	0		LDO2 PD	LDO1 PD	SCPC PD	FDDC PD	FDLD O2PD	FDLD O1PD	0		FDSC P2PD	FDSC P1PD
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Fields

Field	Function
31-28 MODEDLY	Mode Delay Indicates delay to wait for external PMIC mode changes to complete after any mode value change occurs on the PMIC_MODE[1:0] pins. The PMIC_MODE pins change state whenever the aggregated PD*CFG0[PMICMODE] changes. • Time = Period of 4 clocks * 2 ^ value of [MODEDLY] • Reset value = 16 µs The Compute slot version of this field is read-write. This field in the Sense slot is read-only and shows the values written to the Compute slot. This bit is reset by cold reset. Values shown below are examples based on calculations with the period of 4 clocks (250 ns) at a frequency of 16 MHz. 0000b - 250 ns 0001b - 500 ns 0010b - 1000 ns 0011b - 2000 ns 0100b - 4000 ns 0101b - 8000 ns 0110b - 16 µs 0111b - 32 µs 1000b - 64 µs 1001b - 128 µs 1010b - 256 µs 1011b - 512 µs

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1100b - 1024 µs 1101b - 2048 µs 1110b - 4096 µs 1111b - 8192 µs
27-25 —	Reserved
24 FDPDBGLP	Force Band Gap during FDPD Mode to LP Mode In FDPD mode, force band gap to low-power mode. Rising VDD1V8 will not wake up the chip from FDPD mode when this bit is set. If clear, band gap will remain in high-power mode and consume 85 µA. This bit is reset by cold reset. **NOTE** This bit is cleared on wake-up from FDPD mode and any other cold reset! If the rising VDD1V8 is not used as the wake-up source then this bit should be set before entering FDPD every time. 0b - Enables band gap to high-power mode during FDPD mode and wakes up if VDD1V8 POR is asserted then negated 1b - Enables band gap to low-power mode during FDPD mode and will not wake up on VDD1V8 POR negation. RTC, INTR, or reset is used as wake-up
23-13 —	Reserved
12 DCDCPD	DCDC Power Down Indicates power down configuration for DCDC when using a PMIC or other external supply to drive VDDN. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDDN never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE If this bit is set, then software should also write 11b to PMC0.PDRUNCFG0[DCDC_MODE], PMC0.PDSLEEPCFG0[DCDC_MODE], PMC1.PDRUNCFG0[DCDC_MODE], and PMC1.PDSLEEPCFG0[DCDC_MODE] to minimize power when the other blocks of the PMC are all in low power mode and/or powered down. 0b - Power on 1b - Power down
11-10 —	Reserved
9 LDO2PD	LDO VDD2 Power Down Indicates power down configuration for DCDC when using a PMIC or other external supply to drive VDD2. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDD2 never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes. 0b - Power on 1b - Power down
8 LDO1PD	LDO VDD1 Power Down Indicates power down configuration for DCDC when using a PMIC or other external supply to drive VDD1. This bit is reset by cold reset. **NOTE** When set, the control logic waits for the external supply to rise above LVD trip whenever mode changes occur (APPLY_CFG/DeepSleep, etc.). If VDD1 never rises above the LVD trip then the control state machines will get stuck. Software and the external supply controls must make sure that the LVD trip level is less than the final value of the external supply during mode changes. 0b - Power on 1b - Power down
7 SCPCPD	SCPC Power Down Indicates power down configuration for both SCPCs when a PMIC is driving. This bit is reset by cold reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
6 FDDCPD	DCDC Power Down in FDSR Mode Indicates power down configuration for DCDC in FDSR mode. This bit is overridden by [DCDCPD] when [DCDCPD] is set to 1. This bit is reset by cold reset. 0b - Power on 1b - Power down
5 FDLDO2PD	LDO VDD2 Power Down in FDSR Mode Indicates power down configuration for LDO VDD2 in FDSR mode. This bit is overridden by [LDO2PD] when [LDO2PD] is set to 1. This bit is reset by cold reset. 0b - Power on 1b - Power down
4 FDLDO1PD	LDO VDD1 Power Down in FDSR Mode Indicates power down configuration for LDO VDD1 in FDSR mode. This bit is overridden by [LDO1PD] when [LDO1PD] is set to 1. This bit is reset by cold reset. 0b - Power on 1b - Power down
3-2 —	Reserved
1 FDSCP2PD	SCPC VDD2 Power Down in FDSR Mode Indicates power down configuration for SCPC VDD2 in FDSR mode. This bit is overridden by [SCPCPD] when [SCPCPD] is set to 1. This bit is reset by cold reset. 0b - Power on 1b - Power down
0 FDSCP1PD	SCPC VDD1 Power Down in FDSR Mode Indicates power down configuration for SCPC VDD1 in FDSR mode. This bit is overridden by [SCPCPD] when [SCPCPD] is set to 1. This bit is reset by cold reset. 0b - Power on

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Power down

27.7.2.18 3 V Capable VDDIO Range Select (PADVRANGE)

Offset

Register	Offset
PADVRANGE	68h

Function

Indicates the allowed voltage range for the 3 V capable IOs which have a detector to Sense the value of supply, but this consumes current. To reduce power the IOs can be configured for 1.8 V or 3.3 V operation, powering down the detector, but requiring the application to restrict the IO supply voltage to the selected range. The detector is enabled after any cold reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														VRANGE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 VRANGE	3 V Capable IO Bank VDDIO Range Select Indicates the VDDIO range configuration. The Compute slot version of this field is read-write. This bit is reset by cold reset. 00b - Continuous mode. Allows 1.71 - 3.6 V range but consumes static current. 01b - 1.71 - 1.98 V, supply detector off 10b - 3.00 - 3.6 V, supply detector off 11b - Reserved

27.7.2.19 IO Bank State Configuration (PADCFG)

Offset

Register	Offset
PADCFG	6Ch

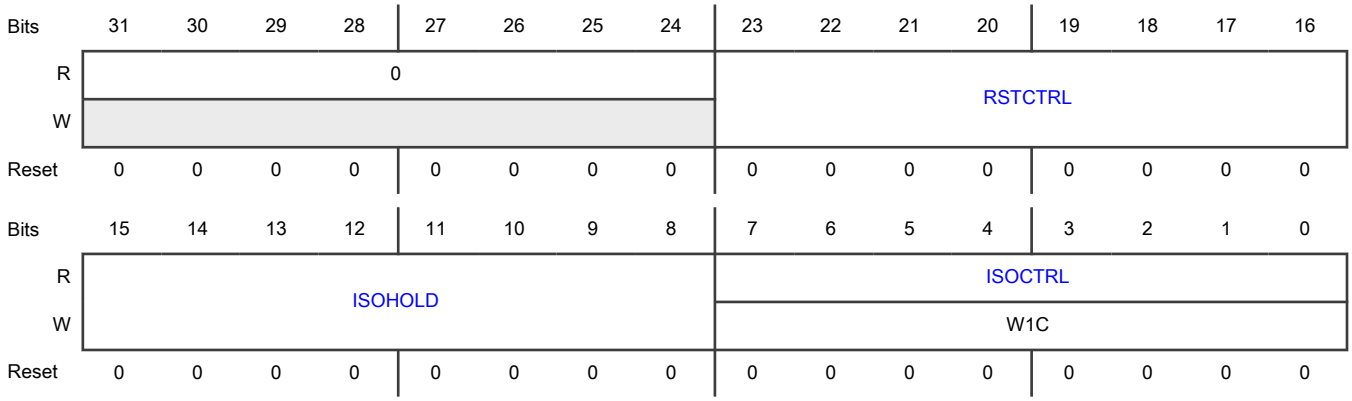
Function

Controls IO bank reset and state retain. Each bit field has the same IO bank distribution map.

Table 233. IO bank mapping

Bit	IO bank(s)	Domain
Bit 7	PMIC_I2C	VDD1_SENSE
Bit 6	Port 8/9/10	VDD1_SENSE
Bit 5	Port 7	VDDN_COM
Bit 4	Port 6	VDDN_COM
Bit 3	Port 5	VDDN_COM
Bit 2	Port 4	VDDN_COM
Bit 1	Port 2	VDD2_COM
Bit 0	Port 0/1/3	VDD2_COM

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16	Reset Control

Table continues on the next page...

Table continued from the previous page...

Field	Function
RSTCTRL	Sets the IO bank reset which tristates the ports. Intended to be used prior to powering off the VDDIO supplies. This bit is reset by cold reset. 0000_0000b - IO banks are not reset 1xxx_xxxxb - PMIC_I2C reset = tristate x1xx_xxxxb - Port 8/9/10 reset = tristate xx1x_xxxxb - Port 7 reset = tristate xxx1_xxxxb - Port 6 reset = tristate xxxx_1xxxb - Port 5 reset = tristate xxxx_x1xxb - Port 4 reset = tristate xxxx_xx1xb - Port 2 reset = tristate xxxx_xxx1b - Port 0/1/3 reset = tristate
15-8 ISOHOLD	IO Bank Isolation Hold Maintains IO bank isolation state after wake-up from FDSR/DPD modes. If clear, the IO bank's state will be retained during FDSR & DPD modes, but will be controllable after wake-up. Clear to 0 so that after PMC powers the controlling logic domain, the isolation is automatically removed and the port is once again controlled by the controlling domain. If the controlling domain is state-retained, then the port state will not change when the isolation is auto-negated. If set, the IO bank will remain in retain mode until ISOCTRL cleared by software. Set to 1 for banks that you want to remain isolated even after power is restored to the controlling logic. If the controlling logic's state is not retained, this enables the software to configure the controlling logic back to its previous state before clearing the isolation/retention to restore control back to the logic. This bit is reset by cold reset. 0000_0000b - IO bank is controllable on wake-up from FDSR/DPD mode 1xxx_xxxxb - PMIC_I2C is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 7 to regain control x1xx_xxxxb - Port 8/9/10 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 6 to regain control xx1x_xxxxb - Port 7 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 5 to regain control xxx1_xxxxb - Port 6 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 4 to regain control xxxx_1xxxb - Port 5 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 3 to regain control xxxx_x1xxb - Port 4 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 2 to regain control xxxx_xx1xb - Port 2 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 1 to regain control

Table continues on the next page...

Table continued from the previous page...

Field	Function
	xxxx_xxx1b - Port 0/1/3 is retained/isolated on wake-up from FDSR/DPD mode. Software must clear [ISOCTRL] bit 0 to regain control
7-0 ISOCTRL	IO bank State Retain Control Controls IO bank state retain, one bit for each VDDIO/VDD domain pair that consists of a controlling VDD logic domain (for example, VDD2_COM or VDD1_SENSE) and a VDDIO supply (VDDIO, VDDIO1, etc.). Hardware sets this bit when driving domain powers down (FDSR/DPD). This bit cannot be cleared if associated domain is not powered. This bit is reset by cold reset. 0000_0000b - IO bank is controllable 1xxx_xxxxb - PMIC_I2C is in retain mode x1xx_xxxxb - Port 8/9/10 is in retain mode xx1x_xxxxb - Port 7 is in retain mode xxx1_xxxxb - Port 6 is in retain mode xxxx_1xxxb - Port 5 is in retain mode xxxx_x1xxb - Port 4 is in retain mode xxxx_xx1xb - Port 2 is in retain mode xxxx_xxx1b - Port 0/1/3 is in retain mode

27.7.2.20 PD Run Configuration 0 in Sense Domain (PDRUNCFG0)

Offset

Register	Offset
PDRUNCFG0	A0h

Function

Configures power down for CPU1 in active mode. This drives the aggregation input for the power down and other controls of PMC functions when the Compute domain is in active mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Sense domain CTRL[APPLYCFG] is set, triggering the state machines or CPU1 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W	1	1														
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	V2DSP_PD	VNCOM_D...	V2COM_D...	V2NMED_...	V2COMP_...	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 RBBSR2_PD	Power Down SRAM RBB in VDD2 Domain SRAM RBB should be activated only when no SRAMs are accessed in the VDD2 domain. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain SRAM RBB should be activated only when no SRAMs are accessed in the VDD1 domain. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
27 AFBBN_PD	Power Down AFBB in VDDN Domain Controls AFBB mode for VDDN domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
26 RBBN_PD	Power Down RBB in VDDN Domain Controls RBB mode for VDDN domain. RBB mode should only be enabled when VDDN domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBBN_PD] = 1, power on. Else power down 1b - Power down
25 AFBB2_PD	Power Down AFBB in VDD2 Domain Controls AFBB mode for VDD2 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
24 RBB2_PD	Power Down RBB in VDD2 Domain Controls RBB mode for VDD2 domain. RBB mode should only be enabled when VDD2 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBB2_PD] = 1, power on. Else power down 1b - Power down
23 AFBB1_PD	Power Down AFBB in VDD1 Domain Controls AFBB mode for VDD1 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 Controls RBB mode for VDD1 domain. RBB mode should only be enabled when VDD1 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDRUNCFG0[AFBB1_PD] = 1, power on. Else power down 1b - Power down
21-20 LDO2_VSEL	Select LDO VDD2 Voltage Selects one of four voltages in LDOVDD2VSEL register. Also selects from one of four LVD trip points in LVDVDD2CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Selects LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level 01b - Selects LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level 10b - Selects LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level 11b - Selects LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level
19-18 LDO2_MODE	LDO VDD2 Mode Selects regulation mode of LDO VDD2. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Select Selects one of four voltages in LDOVDD1VSEL register. Also selects from one of four LVD trip points in LVDVDD1CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level 01b - Selects LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level 10b - Selects LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level 11b - Selects LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level
15-14 LDO1_MODE	LDO VDD1 Mode Selects regulation mode of LDO VDD1. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
13 DCDC_VSEL	Select DCDC Voltage Selects PMC DCDC voltage and LVD level VDDN supply rail from DCDCVSEL registers. DCDCVSEL[VSEL1] is the higher voltage for aggregation. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has 2 copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Selects DCDCVSEL[VSEL0] level 1b - Selects DCDCVSEL[VSEL1] level
12-11 DCDC_MODE	DCDC Power Mode Selects DCDC power mode. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has 2 copies separately accessed by the Compute and Sense slots. This field is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** When using an external supply to power VDDN and POWERCFG[DCDCPD] = 1b, software must also set field DCDC_MODE = 11b in PMC0.PDRUNCFG0, PMC0.PDSLEEPCFG0, PMC1.PDRUNCFG0, and PMC1.PDSLEEPCFG to minimze power when all other PMC subblocks go to low power or power down mode. 00b - Enables DCDC in high-power mode 01b - Enables DCDC in low-power mode 10b - Enables DCDC in ultra-low-power mode 11b - Enables DCDC in ultra-low-power mode
10	Power Down MIPI PHY in VDD2 Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
V2MIPI_PD	Controls power down of the MIPI PHY. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down
9 V2DSP_PD	Power Down VDD2_DSP **NOTE** Read only. Value used for aggregation. Disables VDD2_DSP (HiFi4). Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until after the HiFi4 is powered on. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. **NOTE** VDD2_DSP must be powered down if VDD2_COMP is in DSR mode. 0b - Power on 1b - Power down
8 VNCOM_DSR	Power Switch and DSR for VDDN_COM Enables DSR mode for VDDN_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. You must write 0 to this bit and apply it before entering DPD mode. Write 0 to the bit in either Compute or Sense slot since it is aggregated low. You can write 1 to this bit again after DPD wakeup. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDDN_COM domain 1b - Enables VDDN_COM domain in DSR mode
7 V2COM_DSR	Power Switch and DSR Enable for VDD2_COM

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Enables DSR mode for VDD2_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-only. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2_COM 1b - Enables DSR mode in VDD2_COM
6 V2NMED_DSR	Power Switch and DSR for VDD2 and VDDN Media Enables DSR mode for VDD2 and VDDN media domains. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. You must write 0 to this bit and apply it before entering DPD mode. Write 0 to the bit in either Compute or Sense slot since it is aggregated low. You can write 1 to this bit again after DPD wakeup. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2 and VDDN media domains 1b - Enables VDD2 and VDDN media domains in DSR mode
5 V2COMP_DSR	Power Switch and DSR Enable in VDD2_COMP Domain **NOTE** Read only. Value used for aggregation. Enables DSR mode for VDD2_COMP domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Sense slot version of this field is read-only. 0b - Powers on VDD2_COMP 1b - Enables DSR mode in VDD2_COMP
4 FDPD	Full Deep Power Down (FDPD) Mode **NOTE** Read only. Value used for aggregation. Activates isolation and corrects sequencing for FDPD mode. The chip will only go to FDPD mode if both domains enter Deep Sleep with this bit set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables FDPD mode
3 DPD	Deep Power Down (DPD) Mode **NOTE** Read only. Value used for aggregation. Activates isolation and corrects sequencing for DPD mode. The chip will only go to DPD mode if both domains enter Deep Sleep with this bit set, and at least one domain has [FDPD] cleared. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. 0b - Disables 1b - Enables DPD mode
2 FDSR	Full Deep Sleep Retention (FDSR) Mode **NOTE** Read only. Value used for aggregation. Puts all power switches into DSR or power down. All core domains go to DSR mode or power down mode depending on domain. Muxes VDDN_SR to VDD2 before ramping down supply voltages and turning off VDDN. This is the only way to put VDD1_SENSE domain into DSR mode and control the associated power switch. The chip will only go to FDSR mode if both domains enter Deep Sleep with this bit set, and [DPD] and [FDPD] are not set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. 0b - All power switch states controlled by their respective PD*CFG bits 1b - All core domains in DSR or Power Down mode
1-0 PMICMODE	PMIC Power Mode Select Sets value of PMIC_MODE output pins. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Drives PMIC_MODE pins to 0h 01b - Drives PMIC_MODE pins to 1h 10b - Drives PMIC_MODE pins to 2h 11b - Drives PMIC_MODE pins to 3h

27.7.2.21 PD Run Configuration 1 in Sense Domain (PDRUNCFG1)

Offset

Register	Offset
PDRUNCFG1	A4h

Function

Configures power down for CPU1 in active mode. This drives the aggregation input for the power down and other controls of PMC functions when Compute domain is in active mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Sense domain CTRL[APPLYCFG] is set, triggering the state machines or CPU1 enters and then exits the Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		1														1
W	0															
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1	1	HVDN_PD	LVDN_LP	PORN_LP	AGDET2_...	HVD2_PD	LVD2_LP	POR2_LP	AGDET1_...	HVD1_PD	LVD1_LP	POR1_LP	HVD1V8_...	PMCREF_...	TEMP_PD
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Function
31 —	Reserved
30-17 —	Reserved
16 ROM_PD	Power Down ROM The power switch configuration for the ROM used by aggregation. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 OTP_PD	Power Down OTP The power switch configuration for OTP0 VDD2 Core used by aggregation. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-write. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	Power Down HVD in VDDN Domain Controls the power mode of the VDDN high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
12 LVDN_LP	LVD Low-Power Mode in VDDN Domain Sets LVD in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 PORN_LP	POR Low-Power Mode in VDDN Domain Sets POR in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets POR in low-power mode
10 AGDET2_PD	Power Down aGDET in VDD2 Domain Powers down the VDD2 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 HVD2_PD	Power Down HVD in VDD2 Domain Controls the power mode of the VDD2 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
8 LVD2_LP	LVD Low-Power Mode in VDD2 Domain Sets LVD in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode
7 POR2_LP	POR Low-Power Mode in VDD2 Domain Sets POR in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets POR in low-power mode
6 AGDET1_PD	Power Down aGDET in VDD1 Domain Powers down the VDD1 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 HVD1_PD	Power Down HVD in VDD1 Domain Controls the power mode of the VDD1 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
4 LVD1_LP	LVD Low-Power Mode in VDD1 Domain Sets LVD in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets LVD in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets LVD in low-power mode
3 POR1_LP	POR Low-Power Mode in VDD1 Domain Sets POR in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets POR in high-power mode. In this case, PMC references will be in high-power mode 1b - Sets POR in low-power mode
2 HVD1V8_PD	HVD Power Down Controls the power switch for the VDD1V8 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down
1 PMCREF_LP	PMC References in Low-Power Mode Sets PMC references in low-power mode if high power is not needed by any regulators, HVDs, POR, LVD, AFBB, or the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Sets PMC references in high-power mode 1b - Sets PMC references in low-power mode if not overridden by any *_PD bits
0	Power Down PMC Temperature Sensor

Table continues on the next page...

Table continued from the previous page...

Field	Function
TEMP_PD	Controls the power switch for the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down

27.7.2.22 PD Run Configuration 2 in Sense Domain (PDRUNCFG2)

Offset

Register	Offset
PDRUNCFG2	A8h

Function

Configures main memory array and periphery power down for CPU1 in active mode. This drives aggregation input for the power down control of the arrays and peripheries of the main SRAM partitions when CPU1 is in active mode. If a partition's aggregated PD*CFG2[SRAMn] is 0, then the array is powered and the periphery power is controlled by PD*CFG3. If PD*CFG2[SRAMn] is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until Sense domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU1 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	Power Down RAM Partition n Array Powers down the array and periphery of the associated memory, regardless of PD*CFG3 state. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Array power on, periphery power controlled by PD*CFG3 1b - Power down

27.7.2.23 PD Run Configuration 3 in Sense Domain (PDRUNCFG3)

Offset

Register	Offset
PDRUNCFG3	ACh

Function

Powers down the selected main memory partitions' periphery logic to save power, while maintaining the contents if the corresponding PD*CFG2[SRAMn] is configured to keep the array on.

This register is reset by cold reset.

NOTE

SRAM0 to SRAM17 are the VDD2_COM main memories. SRAM18 to SRAM29 are the VDD1_SENSE main memories. The memory map in some other part of the RM should have this info, and should also include (for internal use) how many instances are in each partition, and size of these memory instances.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down***
- Changes to this register do not affect the memories until Sense domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU1 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	Power Down RAM Partition n Periphery Powers down SRAM partition n peripheral. SRAM partition n peripheral is powered on only when the corresponding array SRAM n is powered on. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on when the corresponding array SRAM 0 is 0 1b - Power down

27.7.2.24 PD Run Configuration 4 in Sense Domain (PDRUNCFG4)

Offset

Register	Offset
PDRUNCFG4	B0h

Function

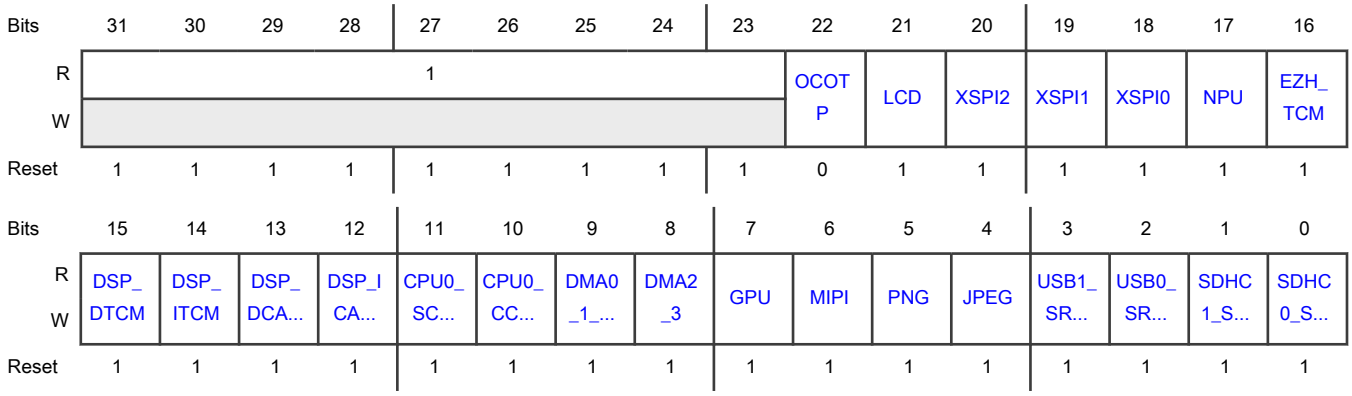
Configures peripheral memory array and peripheral power down for CPU1 in active mode. This drives aggregation input for the power down control of the arrays and peripheries of the Peripherals" SRAM partitions when CPU1 is in active mode. If a partition's aggregated PD*CFG4[PD] bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4[PD] bit is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until Sense domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU1 enters and then exits Deep Sleep mode.

Diagram



Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[LCD] or PDRUNCFG5[LCD] or both are set the LCDIF controller RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2 and MMU2 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16	EZH-V TCM

Table continues on the next page...

Table continued from the previous page...

Field	Function
EZH_TCM	Powers down EZH-V TCM RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_DTCM] or PDRUNCFG5[DSP_DTCM] or both are set the HiFi4 data TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_ITCM] or PDRUNCFG5[DSP_ITCM] or both are set the HiFi4 instruction TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_DCACHE] or PDRUNCFG5[DSP_DCACHE] or both are set the HiFi4 data cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DSP_ICACHE] or PDRUNCFG5[DSP_ICACHE] or both are set the HiFi4 instruction cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Powers down DMA0-1 and PKC periphery, and ETF RAMs array. This RAM does not support a power down of DMA and PKC array, so PDRUNCFG4 and PDRUNCFG5 both control power down of DMA and PKC periphery. If either PDRUNCFG4[DMA0_1_P_E] or PDRUNCFG5[DMA0_1_P_E] or both are set the DMA0-1 and PKC RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This RAM does not support a power down of the array, so PDRUNCFG4 and PDRUNCFG5 both control power down of the periphery. If either PDRUNCFG4[DMA2_3] or PDRUNCFG5[DMA2_3] or both are set the DMA2 and DMA3 RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.2.25 PD Run Configuration 5 in Sense Domain (PDRUNCFG5)

Offset

Register	Offset
PDRUNCFG5	B4h

Function

Configures peripheral memory periphery power down for CPU1 in active mode. This drives aggregation input for the power down control of the peripheries of the Peripherals" SRAM partitions when CPU1 is in active mode. If a partition's aggregated PD*CFG4 bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4 bit is 1, then the periphery is powered off and the array power is determined by PD*CFG4. When the power domain of the peripheral in DSR mode, the peripheries for all SRAMs in that domain are automatically powered down.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down***
- Changes to this register do not affect the memories until Sense domain CTRL[APPLYCFG] is set, triggering the state machines, or CPU1 enters and then exits Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2, MMU2, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16	EZH-V TCM

Table continues on the next page...

Table continued from the previous page...

Field	Function
EZH_TCM	Powers down EZH-V TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1, PKC, and ETF RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.2.26 PD Sleep Configuration 0 in Sense Domain (PDSLEEPCFG0)

Offset

Register	Offset
PDSLEEPCFG0	B8h

Function

Configures power down for CPU1 in Deep Sleep mode. This drives the aggregation input for the power down and other controls of PMC functions when Sense domain is in Deep Sleep mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Sense domain enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W	1	1														
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	1	VNCOM_D...	V2COM_D...	V2NMED_...	1	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0

Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 —	Reserved
29 RBBSR2_PD	Power Down SRAM RBB in VDD2 Domain SRAM RBB should be activated only when no SRAMs in the domain are accessed. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain SRAM RBB should be activated only when no SRAMs in the domain are accessed. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
27 AFBBN_PD	Power Down AFBB in VDDN Domain Controls AFBB mode for VDDN domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
26 RBBN_PD	Power Down RBB in VDDN Domain Controls RBB mode for VDDN domain. RBB mode should only be enabled when VDDN domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBBN_PD] = 1, power on. Else power down 1b - Power down
25 AFBB2_PD	Power Down AFBB in VDD2 Domain Controls AFBB mode for VDD2 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
24 RBB2_PD	Power Down RBB in VDD2 Domain Controls RBB mode for VDD2 domain. RBB mode should only be enabled when VDD2 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBB2_PD] = 1, power on. Else power down 1b - Power down
23 AFBB1_PD	Power Down AFBB in VDD1 Domain Controls AFBB mode for VDD1 domain. AFBB mode must be enabled for high-frequency operation. See Datasheet for details. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 Domain Controls RBB mode for VDD1 domain. RBB mode should only be enabled when VDD1 domain is not clocked. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - If PDSLEEPCFG0[AFBB1_PD] = 1, power on. Else power down 1b - Power down
21-20 LDO2_VSEL	LDO VDD2 Voltage Select Selects one of four voltages in LDOVDD2VSEL register. Also selects from one of four LVD trip points in LVDVDD2CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level 01b - Selects LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level 10b - Selects LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level 11b - Selects LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level
19-18 LDO2_MODE	LDO VDD2 Regulator Mode Selects regulation mode of LDO VDD2. When high-power mode is selected, the PMC references automatically enter high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - High-power mode 10b,11b - Low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Select Selects one of four voltages in LDOVDD1VSEL register. Also selects from one of four LVD trip points in LVDVDD1CTRL. 00b means the lowest voltage and 11b means the highest voltage. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Selects LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level 01b - Selects LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level 10b - Selects LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level 11b - Selects LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level
15-14 LDO1_MODE	LDO VDD1 Regulator Mode Selects regulation mode of LDO VDD1. When high-power mode is selected, the PMC references automatically enter high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Bypass mode 01b - High-power mode 10b,11b - Low-power mode
13 DCDC_VSEL	DCDC Voltage Select Selects the PMC DCDC voltage and LVD level VDDN supply rail from the DCDCVSEL register. DCDCVSEL[VSEL1] is the higher voltage for aggregation. This field is aggregated high. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Selects DCDCVSEL[VSEL0] level 1b - Selects DCDCVSEL[VSEL1] level
12-11 DCDC_MODE	DCDC Power Mode Selects DCDC power mode. When high-power mode is selected, the PMC references automatically enter the high-power mode. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has 2 copies separately accessed by the Compute and Sense slots. This field is reset by warm reset. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** When using an external supply to power VDDN and POWERCFG[DCDCPD] = 1b, software must also set field DCDC_MODE = 11b in PMC0.PDRUNCFG0, PMC0.PDSLEEPCFG0, PMC1.PDRUNCFG0, and PMC1.PDSLEEPCFG to minimze power when all other PMC subblocks go to low power or power down mode. 00b - Enables DCDC in high-power mode 01b - Enables DCDC in low-power mode 10b - Enables DCDC in ultra-low-power mode 11b - Enables DCDC in ultra-low-power mode
10 V2MIPI_PD	Power Down MIPI PHY in VDD2 Domain Controls power down of the MIPI PHY. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down
9 V2DSP_PD	Power Down VDD2_DSP Configuration used for V2DSP_PD for aggregation. Disables VDD2_DSP (HiFi4). This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-write.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** VDD2_DSP must be powered down if VDD2_COMP is in DSR mode. **NOTE** Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until the HiFi4 is powered. 0b - Power on 1b - Power down
8 VNCOM_DSR	Power Switch and DSR Enable in VDDN_COM Domain Enables DSR mode for VDDN_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDDN_COM 1b - Enables DSR mode in VDDN_COM
7 V2COM_DSR	Power Switch and DSR Enable in VDD2_COM Domain Enables DSR mode for VDD2_COM domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2_COM 1b - Enables DSR mode in VDD2_COM
6 V2NMED_DSR	Power Switch and DSR Enable in VDD2_MEDIA and VDDN_MEDIA Domains Enables DSR mode for VDD2_MEDIA and VDDN_MEDIA domains. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. **NOTE** Merged VDD2 and VDDN media switch controls. 0b - Powers on VDD2_MEDIA and VDDN_MEDIA 1b - Enables DSR mode in VDD2_MEDIA and VDDN_MEDIA
5 V2COMP_DSR	Power Switch and DSR Enable in VDD2_COMP Domain Configuration for V2COMP_DSR used for aggregation. Enables DSR mode for VDD2_COMP domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-write. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on VDD2_COMP 1b - Enables DSR mode in VDD2_COMP
4 FDPD	Full Deep Power Down (FDPD) Mode Activates isolation and corrects sequencing for FDPD mode. The chip will only go to FDPD mode if both domains enter Deep Sleep with this bit set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. 0b - Disables 1b - Enables FDPD mode
3 DPD	Deep Power Down (DPD) Mode Activates isolation and corrects sequencing for DPD mode. The chip will only go to DPD mode if both domains enter Deep Sleep with this bit set, and at least one domain has [FDPD] cleared. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. You must clear this bit whenever PDSLEEPCFG0[FDSR] is set. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Disables 1b - Enables DPD mode
2	Full Deep Sleep Retention (FDSR) Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
FDSR	Puts all power switches into DSR or power down. All core domains go to DSR mode or power down mode depending on domain. Muxes VDDN_SR to VDD2 before ramping down supply voltages and turning off VDDN. This is the only way to put VDD1_SENSE domain into DSR mode and control the associated power switch. The chip will only go to FDSR mode if both domains enter Deep Sleep with this bit set, and [DPD] and [FDPD] are not selected instead. You must clear this bit whenever PDSLEEPCFG0[DPD] is set. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - All power switch states controlled by their respective PD*CFG bits 1b - All core domains in DSR or Power Down mode
1-0 PMICMODE	PMIC Mode Sets value of PMIC_MODE output pins. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 00b - Drives PMIC_MODE pins to 0h 01b - Drives PMIC_MODE pins to 1h 10b - Drives PMIC_MODE pins to 2h 11b - Drives PMIC_MODE pins to 3h

27.7.2.27 PD Sleep Configuration 1 in Sense Domain (PDSLEEPCFG1)

Offset

Register	Offset
PDSLEEPCFG1	BCh

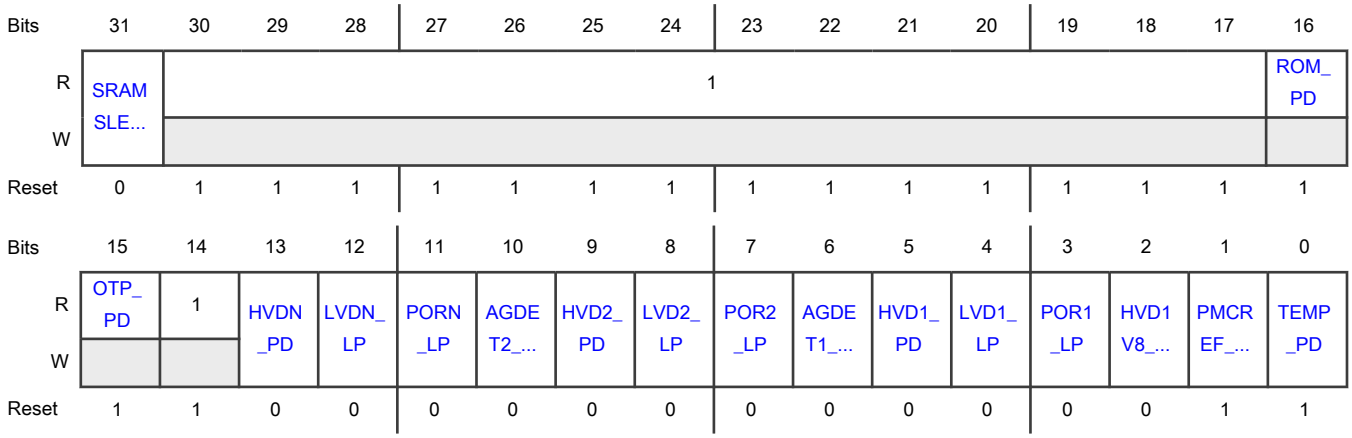
Function

Configures power down for CPU1 in Deep Sleep mode. This drives the aggregation input for the power down and other controls of PMC functions when Sense domain is in Deep Sleep mode. See PD*CFG and aggregation. Reserved bits must not be changed by user software.

NOTE

Changes to this register do not affect PMC until Sense domain enters Deep Sleep mode.

Diagram



Fields

Field	Function
31 SRAMSLEEP	SRAM Sleep Mode Sets SRAM sleep mode. This is needed when VDD Core can be < 0.85 V to ensure contents retained. No power savings. Memories are not accessible in RAM sleep mode. It should only be set in PDSLEEPCFG. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Normal 1b - Sleep
30-17 —	Reserved
16 ROM_PD	ROM Power Down Configuration used for ROM_PD for aggregation. Controls the power switch configuration for the ROM. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-write. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15	OTP Power Down in VDD2 Domain

Table continues on the next page...

Table continued from the previous page...

Field	Function
OTP_PD	Configuration used for OTP_PD for aggregation. Controls the power switch configuration for OTP0 VDD2 Core. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. The Compute slot version of this field is read-write. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	HVD Power Down in VDDN Domain Controls the power mode of the VDDN high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
12 LVDN_LP	LVD Low power in VDDN Domain Sets LVD in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
11 PORN_LP	POR Low Power in VDDN Domain Sets POR in low-power mode in VDDN domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
10 AGDET2_PD	aGDET Power Down in VDD2 Domain Powers down the VDD2 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 HVD2_PD	HVD Power Down in VDD2 Domain Controls the power mode of the VDD2 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
8 LVD2_LP	LVD Low power in VDD2 Domain Sets LVD in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 POR2_LP	POR Low power in VDD2 Domain Sets POR in low-power mode in VDD2 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
6 AGDET1_PD	aGDET Power Down in VDD1 Domain Powers down the VDD1 analog glitch detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 HVD1_PD	HVD Power Down in VDD1 Domain Controls the power mode of the VDD1 high-voltage detect. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
4 LVD1_LP	LVD Low power in VDD1 Domain Sets LVD in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
3 POR1_LP	POR Low Power in VDD1 Domain Sets POR in low-power mode in VDD1 domain. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Low-power mode
2 HVD1V8_PD	HVD VDD1V8 Power Down This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case the aggregated PDCFGSTATUS1[PMCREF_LP] bit will be in high-power mode 1b - Power down
1 PMCREF_LP	PMC References Low Power Sets PMC reference in low-power mode if high power is not needed by any regulators, HVDs, POR, LVD, AFBB, or the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - High-power mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Low-power mode if not overridden by any *_PD bits
0 TEMP_PD	PMC Temperature Sensor Power Down Controls the power switch for the PMC temperature sensor. This field is aggregated low. For details on aggregation logic, see PD*CFG and aggregation. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on. In this case, PMC references will be in high-power mode 1b - Power down

27.7.2.28 PD Sleep Configuration 2 in Sense Domain (PDSLEEPCFG2)

Offset

Register	Offset
PDSLEEPCFG2	C0h

Function

Configures power down for main memory array and periphery for CPU1 in Deep Sleep mode. This drives aggregation input for the power down control of the arrays and peripheries of the min SRAM partitions when CPU1 is in Deep Sleep mode. If a partition's aggregated PD*CFG2[SRAMn] is 0, then the array is powered and the periphery power is controlled by PD*CFG3. If PD*CFG2[SRAMn] is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until CPU1 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Array Power Down This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Powers on RAM Partition 0 array 1b - Powers down RAM Partition 0 array and periphery

27.7.2.29 PD Sleep Configuration 3 in Sense Domain (PDSLEEPCFG3)

Offset

Register	Offset
PDSLEEPCFG3	C4h

Function

Powers down the selected main memory partitions' periphery logic to save power, while maintaining the contents if the corresponding PD*CFG2[SRAMn] is configured to keep the array on.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down*** .
- Changes to this register do not affect the memories until CPU1 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Periphery Power Down Periphery is powered only when the corresponding array SRAMn is powered. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.2.30 PD Sleep Configuration 4 in Sense Domain (PDSLEEPCFG4)

Offset

Register	Offset
PDSLEEPCFG4	C8h

Function

Configures power down for peripheral memory array and periphery for CPU1 in Deep Sleep mode. This drives aggregation input for the power down control of the arrays and peripheries of the peripherals" SRAM partitions when CPU1 is in Deep Sleep mode. If a partition's aggregated PD*CFG4[PD] bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4[PD] bit is 1, then the partition's array and periphery are both powered off.

This register is reset by cold reset.

NOTE

Changes to this register do not affect the memories until CPU1 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[LCD] or PDSLEEPCFG5[LCD] or both are set the LCDIF controller RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2 and MMU2 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_DTCM] or PDSLEEPCFG5[DSP_DTCM] or both are set the HiFi4 data TCM RAM periphery will be powered down.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_ITCM] or PDSLEEPCFG5[DSP_ITCM] or both are set the HiFi4 instruction TCM RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_DCACHE] or PDSLEEPCFG5[DSP_DCACHE] or both are set the HiFi4 data cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DSP_ICACHE] or PDSLEEPCFG5[DSP_ICACHE] or both are set the HiFi4 instruction cache RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1 and PKC periphery, and ETF RAMs array. This RAM does not support a power down of DMA and PKC array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of DMA and PKC periphery. If either PDSLEEPCFG4[DMA0_1_P_E] or PDSLEEPCFG5[DMA0_1_P_E] or both are set the DMA0-1 and PKC RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Powers down DMA2 and DMA3 RAMs periphery. This RAM does not support a power down of the array, so PDSLEEPCFG4 and PDSLEEPCFG5 both control power down of the periphery. If either PDSLEEPCFG4[DMA2_3] or PDSLEEPCFG5[DMA2_3] or both are set the DMA2 and DMA3 RAM periphery will be powered down. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY Powers down MIPI PHY RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0	uSDHC0 SRAM

Table continues on the next page...

Table continued from the previous page...

Field	Function
SDHC0_SRAM	Powers down uSDHC0 RAMs array. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.2.31 PD Sleep Configuration 5 in Sense Domain (PDSLEEPCFG5)

Offset

Register	Offset
PDSLEEPCFG5	CCh

Function

Configures power down for peripheral memory periphery for CPU1 in Deep Sleep mode. This drives aggregation input for the power down control of the peripheries of the Peripherals" SRAM partitions when CPU1 is in Deep Sleep. If a partition's aggregated PD*CFG4 bit is 0, then the array is powered and the periphery power is controlled by PD*CFG5. If the PD*CFG4 bit is 1, then the periphery is powered off and the array power is determined by PD*CFG4. When the power domain of the Peripheral in DSR mode the peripheries for all SRAMs in that domain are automatically powered down.

This register is reset by cold reset.

NOTE

- Memory peripheries are powered down whenever **the array is powered down** or **the associated power domain is in *DSR mode* or *powered down*** .
- Changes to this register do not affect the memories until CPU1 enters Deep Sleep mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Powers down OCOTP shadow RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
21 LCD	LCDIF Powers down LCDIF controller RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 Powers down XSPI2, MMU2, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache Powers down XSPI1, MMU1, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache Powers down XSPI0, MMU0, and cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
17 NPU	NPU Powers down NPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM Powers down EZH-V TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM Powers down HiFi4 data TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM Powers down HiFi4 instruction TCM RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache Powers down HiFi4 data cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache Powers down HiFi4 instruction cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache Powers down CPU0 system cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache Powers down CPU0 code cache RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF Powers down DMA0-1, PKC, and ETF RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 Powers down DMA2 and DMA3 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
7 GPU	VGPU Powers down VGPU RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
6 MIPI	MIPI Powers down MIPI RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
5 PNG	PNGDEC Powers down PNGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC Powers down JPEGDEC RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 SRAM Powers down USB1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
2 USB0_SRAM	USB0 SRAM Powers down USB0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 SRAM Powers down uSDHC1 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 SRAM Powers down uSDHC0 RAMs periphery. This field has two copies separately accessed by the Compute and Sense slots. **NOTE** To update PMC's configuration with the updated register value software needs to set CTRL[APPLYCFG], or a mode change must occur. 0b - Power on 1b - Power down

27.7.2.32 PD Configuration Status 0 (PDCFGSTATUS0)

Offset

Register	Offset
PDCFGSTATUS0	D0h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG0 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep mode), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1	1	RBBSR2_...	RBBSR1_...	AFBBN_PD	RBBN_PD	AFBB2_PD	RBB2_PD	AFBB1_PD	RBB1_PD	LDO2_VSEL		LDO2_MODE		LDO1_VSEL	
W																
Reset	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDO1_MODE		DCDC_VS...	DCDC_MODE		V2MIPI_...	V2DSP_PD	VNCOM_D...	V2COM_D...	V2NMED_...	V2COMP_...	FDPD	DPD	FDSR	PMICMODE	
W																
Reset	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 RBBSR2_PD	SRAM RBB Power Down in VDD2 Domain Indicates SRAM RBB power down status in VDD2 domain. SRAM RBB should be activated only when no SRAMs in the domain are accessed. This bit is reset by warm reset. 0b - Power on 1b - Power down
28 RBBSR1_PD	Power Down SRAM RBB in VDD1 Domain Indicates SRAM RBB power down status in VDD1 domain. SRAM RBB should be activated only when no SRAMs in the domain are accessed. This bit is reset by warm reset. 0b - Power on 1b - Power down
27	AFBB Power Down in VDDN domain Indicates AFBB power down status in VDDN domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
AFBBN_PD	This bit is reset by warm reset. 0b - Power on 1b - Power down
26 RBBN_PD	RBB Power Down in VDDN domain Indicates RBB power down status in VDDN domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBBN_PD] = 1, otherwise RBB is disabled 1b - Power down
25 AFBB2_PD	AFBB Power Down in VDD2 domain Indicates AFBB power down status in VDD2 domain. This bit is reset by warm reset. 0b - Power 1b - Power down
24 RBB2_PD	RBB Power Down in VDD2 domain Indicates RBB power down status in VDD2 domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBB2_PD] = 1, otherwise RBB is disabled 1b - Power down
23 AFBB1_PD	AFBB Power Down in VDD1 domain Indicates AFBB power down status in VDD1 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
22 RBB1_PD	Power Down RBB in VDD1 domain Indicates RBB power down status in VDD1 domain. This bit is reset by warm reset. 0b - Enables RBB if the aggregated value of [AFBB1_PD] = 1, otherwise RBB is disabled 1b - Power down
21-20 LDO2_VSEL	LDO VDD2 Voltage Selection Indicates the selected level for LDO VDD2 and LVD from the LDOVDD2VSEL and LVDVDD2CTRL registers. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - LDOVDD2VSEL[VSEL0] and LVDVDD2CTRL[LVL0] level selected (lowest voltage) 01b - LDOVDD2VSEL[VSEL1] and LVDVDD2CTRL[LVL1] level selected 10b - LDOVDD2VSEL[VSEL2] and LVDVDD2CTRL[LVL2] level selected 11b - LDOVDD2VSEL[VSEL3] and LVDVDD2CTRL[LVL3] level selected (highest voltage)
19-18 LDO2_MODE	LDO VDD2 Regulator Mode Indicates the effective regulation mode for LDO VDD2. This bit is reset by warm reset. 00b - LDO VDD1 is in bypass mode 01b - LDO VDD2 is in high-power mode 10b,11b - LDO VDD2 is in low-power mode
17-16 LDO1_VSEL	LDO VDD1 Voltage Selection Indicates the selected level for LDO VDD1 and LVD from the LDOVDD1VSEL and LVDVDD1CTRL registers. This bit is reset by warm reset. 00b - LDOVDD1VSEL[VSEL0] and LVDVDD1CTRL[LVL0] level selected (lowest voltage) 01b - LDOVDD1VSEL[VSEL1] and LVDVDD1CTRL[LVL1] level selected 10b - LDOVDD1VSEL[VSEL2] and LVDVDD1CTRL[LVL2] level selected 11b - LDOVDD1VSEL[VSEL3] and LVDVDD1CTRL[LVL3] level selected (highest voltage)
15-14 LDO1_MODE	LDO VDD1 Regulator Mode Indicates the effective regulation mode for LDO VDD1. This bit is reset by warm reset. 00b - LDO VDD1 is in bypass mode 01b - LDO VDD1 is in high-power mode 10b,11b - LDO VDD1 is in low-power mode
13 DCDC_VSEL	DCDC Voltage Selection Indicates the selection of DCDC voltage and VDDN LVD levels from the DCDCVSEL and LVDVDDNCTRL registers. This bit is reset by warm reset. This bit shows the effective aggregated value based on the CPU0 and CPU1 status and the PDRUNCFG and PDSLEEPCFG registers. 0b - [VSEL0] level selected 1b - [VSEL1] level selected
12-11	DCDC Power Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
DCDC_MODE	Indicates DCDC power mode. This field is reset by warm reset. 00b - DCDC in high-power mode 01b - DCDC in low-power mode 10b - DCDC in ultra-low-power mode 11b - DCDC in ultra-low-power mode
10 V2MIPI_PD	MIPI PHY Power Down Indicates the power down status of MIPI PHY. This bit is reset by warm reset. **NOTE** MIPI PHY must be powered down if VDD2_MEDIA is in DSR mode. 0b - Power on 1b - Power down
9 V2DSP_PD	HiFi4 Power Down Indicates the power down status of HiFi4. This bit is reset by warm reset. **NOTE** Must be powered down if VDD2_COMP is in DSR mode. **NOTE** Software must power down the HiFi4 cache and TCM periphery power switches prior to powering down the HiFi4, and the memories must not be powered on until the HiFi4 is powered. 0b - Power on 1b - Power down
8 VNCOM_DSR	DSR of VDDN_COM Domain Indicates the DSR status of VDDN_COM domain. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
7 V2COM_DSR	DSR of VDD2_COM Domain Indicates the DSR status of VDD2_COM domain. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - DSR mode
6 V2NMED_DSR	DSR of VDD2_MEDIA and VDDN_MEDIA Domains Indicates the DSR status of VDD2_MEDIA and VDDN_MEDIA domains. **NOTE** Merged VDD2 and VDDN media switch controls. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
5 V2COMP_DSR	DSR of VDD2_COMP Domain Indicates the DSR status of VDD2_COMP domain. This bit is reset by warm reset. 0b - Power on 1b - DSR mode
4 FDPD	FDPD Mode Indicates FDPD status. This bit is reset by warm reset. 0b - Enables 1b - Activates isolation and correct sequencing for FDPD mode
3 DPD	DPD Mode Indicates DPD status. This bit is reset by warm reset. 0b - Enables 1b - Activates isolation and correct sequencing for DPD mode
2 FDSR	FDSR Mode Indicates the FDSR mode of all core domains. This bit is reset by warm reset. 0b - Power switches and DSR status are determined by the single power domain controls 1b - All core domains are in FDSR mode or power down
1-0 PMICMODE	PMIC_MODE Output Value This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates the aggregated PMIC_MODE output value driven to PMIC_MODE[1:0] pins, which is lowest value (highest power) mode selected by the active PD*CFG registers in Compute and Sense domains. PMIC_MODE output value is used to control external power supplies. 00b - PMIC_MODE[1:0] pins = 00 01b - PMIC_MODE[1:0] pins = 01 10b - PMIC_MODE[1:0] pins = 10 11b - PMIC_MODE[1:0] pins = 11

27.7.2.33 PD Configuration Status 1 (PDCFGSTATUS1)

Offset

Register	Offset
PDCFGSTATUS1	D4h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG1 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SRAM SLE...	1														ROM_ PD
W																
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OTP_ PD	1	HVDN _PD	LVDN_ LP	PORN _LP	AGDE T2_...	HVD2_ PD	LVD2_ LP	POR2 _LP	AGDE T1_...	HVD1_ PD	LVD1_ LP	POR1 _LP	HVD1 V8_...	PMCR EF_...	TEMP _PD
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Function
31	SRAM Sleep Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
SRAMSLEEP	Indicates the status of SRAM Sleep mode. This is needed when VDD Core can be < 0.85 V to ensure contents retained. No power savings. Memories are not accessible in RAM sleep mode. It should only be set in PDSLEEPCFG. This bit is reset by warm reset. 0b - Normal 1b - Sleep
30-17 —	Reserved
16 ROM_PD	ROM Power Down Indicates the power down status of ROM. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 OTP_PD	OTP Power Down Indicates the power down status of OTP VDD2. This bit is reset by warm reset. 0b - Power on 1b - Power down
14 —	Reserved
13 HVDN_PD	HVD Power Down in VDDN Domain Indicates the power down status of the high-voltage detect in VDDN domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 LVDN_LP	LVD Low Power in VDDN Domain Indicates the status of LVD low-power mode in VDDN domain. This bit is reset by warm reset. 0b - High power 1b - Low power
11 PORN_LP	POR Low Power in VDDN Domain Indicates the status of POR low-power mode in VDDN domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This bit is reset by warm reset. 0b - High power 1b - Low power
10 AGDET2_PD	aGDET Power Down in VDD2 Domain Indicates the power down status of the analog glitch detect in VDD2 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 HVD2_PD	HVD Power Down in VDD2 Domain Indicates the power down status of the high-voltage detect in VDD2 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
8 LVD2_LP	LVD Low Power in VDD2 Domain Indicates the status of LVD low-power mode in VDD2 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
7 POR2_LP	POR Low Power in VDD2 Domain Indicates the status of POR low-power mode in VDD2 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
6 AGDET1_PD	aGDET Power Down in VDD1 Domain Indicates the power down status of the analog glitch detect in VDD1 domain. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 HVD1_PD	HVD Power Down in VDD1 Domain Indicates the power down status of the high-voltage detect in VDD1 domain. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
4 LVD1_LP	LVD Low Power in VDD1 Domain Indicates the status of LVD low-power mode in VDD1 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
3 POR1_LP	POR Low Power in VDD1 Domain Indicates the status of POR low-power mode in VDD1 domain. This bit is reset by warm reset. 0b - High power 1b - Low power
2 HVD1V8_PD	HVD Power Down Indicates the power down status of HVD. This bit is reset by warm reset. 0b - Power on 1b - Power down
1 PMCREF_LP	PMC References Low Power Indicates the status of PMC reference low-power mode. This bit is reset by warm reset. 0b - High-power mode 1b - Low-power mode
0 TEMP_PD	PMC Temperature Sensor Power Down Indicates the power down status of PMC temperature sensor. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.2.34 PD Configuration Status 2 (PDCFGSTATUS2)

Offset

Register	Offset
PDCFGSTATUS2	D8h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG2 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Array Power Down Indicates the power down status of RAM Partition n Array. This bit is reset by warm reset. 0b - Array was powered on 1b - Array and periphery were powered down

27.7.2.35 PD Configuration Status 3 (PDCFGSTATUS3)

Offset

Register	Offset
PDCFGSTATUS3	DCh

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG3 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1		SRAM 29	SRAM 28	SRAM 27	SRAM 26	SRAM 25	SRAM 24	SRAM 23	SRAM 22	SRAM 21	SRAM 20	SRAM 19	SRAM 18	SRAM 17	SRAM 16
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRAM 15	SRAM 14	SRAM 13	SRAM 12	SRAM 11	SRAM 10	SRAM 9	SRAM 8	SRAM 7	SRAM 6	SRAM 5	SRAM 4	SRAM 3	SRAM 2	SRAM 1	SRAM 0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fields

Field	Function
31-30 —	Reserved
29-0 SRAMn	RAM Partition n Periphery Power Down This bit is reset by warm reset. Indicates if the SRAM partition n periphery is powered or not. Periphery is powered only when the corresponding array SRAMn is powered. 0b - Power on 1b - Power down

27.7.2.36 PD Configuration Status 4 (PDCFGSTATUS4)

Offset

Register	Offset
PDCFGSTATUS4	E0h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG4 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Indicates the power down status of OCOTP shadow RAMs array. This bit is reset by warm reset. 0b - Array Powered 1b - Array Power Down
21 LCD	LCDIF Controller RAMs Array Indicates the power down status of LCDIF controller RAMs array. This bit is reset by warm reset. 0b - Power on

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Power down
20 XSPI2	XSPI2 and MMU2 RAMs Array Indicates the power down status of XSPI2 and MMU2 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache RAMs Array Indicates the power down status of XSPI1, MMU1, and cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache RAMs Array Indicates the power down status of XSPI0, MMU0, and cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
17 NPU	NPU RAMs Array Indicates the power down status of NPU RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
16 EZH_TCM	EZH-V TCM RAMs Array Indicates the power down status of EZH-V TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM RAMs Array Indicates the power down status of HiFi4 data TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
14 DSP_ITCM	HiFi4 Instruction TCM RAMs Array Indicates the power down status of HiFi4 instruction TCM RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache RAMs Array Indicates the power down status of HiFi4 data cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache RAMs Array Indicates the power down status of HiFi4 instruction cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache RAMs Array Indicates the power down status of CPU0 system cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
10 CPU0_CCACHE	CPU0 Code Cache RAMs Array Indicates the power down status of CPU0 code cache RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF RAMs Array Indicates the power down status of DMA0-1, PKC, and ETF RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
8	DMA2 and DMA3 RAMs Array Indicates the power down status of DMA2 and DMA3 RAMs array.

Table continues on the next page...

Table continued from the previous page...

Field	Function
DMA2_3	This bit is reset by warm reset. 0b - Power on 1b - Power down
7 GPU	VGPU RAMs Array Indicates the power down status of VGPU RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY RAMs Array Indicates the power down status of MIPI PHY RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 PNG	PNGDEC RAMs Array Indicates the power down status of PNGDEC RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
4 JPEG	JPEGDEC RAMs Array Indicates the power down status of JPEGDEC RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 RAMs Array Indicates the power down status of USB1 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 RAMs Array Indicates the power down status of USB0 RAMs array. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 RAMs Array Indicates the power down status of uSDHC1 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 RAMs Array Indicates the power down status of uSDHC0 RAMs array. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.2.37 PD Configuration Status 5 (PDCFGSTATUS5)

Offset

Register	Offset
PDCFGSTATUS5	E4h

Function

Shows the currently aggregated value of Sense and Compute domains in PD*CFG5 registers. Updates immediately after the mode change event (e.g. CTRL[APPLYCFG] = 1, CPUx enters Deep Sleep, etc), while PMC is transitioning to the new configuration (as indicated by reading STATUS[BUSY] and CTRL[APPLYCFG]) and held until the next mode transition or APPLYCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1									OCOTP	LCD	XSPI2	XSPI1	XSPI0	NPU	EZH_TCM
W																
Reset	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSP_DTCM	DSP_ITCM	DSP_DCA...	DSP_ICA...	CPU0_SC...	CPU0_CC...	DMA0_1_...	DMA2_3	GPU	MIPI	PNG	JPEG	USB1_SR...	USB0_SR...	SDHC1_S...	SDHC0_S...
W																
Reset	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-23 —	Reserved
22 OCOTP	OCOTP Shadow SRAM Indicates the power down status of OCOTP shadow RAMs periphery. This bit is reset by warm reset. 0b - Array Powered 1b - Array Power Down
21 LCD	LCDIF Controller RAMs Periphery Indicates the power down status of LCDIF controller RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
20 XSPI2	XSPI2 and MMU2 RAMs Periphery Indicates the power down status of XSPI2, MMU2, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
19 XSPI1	XSPI1, MMU1, and Cache RAMs Periphery Indicates the power down status of XSPI1, MMU1, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
18 XSPI0	XSPI0, MMU0, and Cache RAMs Periphery Indicates the power down status of XSPI0, MMU0, and cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
17 NPU	NPU RAMs Periphery Indicates the power down status of NPU RAMs periphery. This bit is reset by warm reset. 0b - Power on

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Power down
16 EZH_TCM	EZH-V TCM RAMs Periphery Indicates the power down status of EZH-V TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
15 DSP_DTCM	HiFi4 Data TCM RAMs Periphery Indicates the power down status of HiFi4 data TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
14 DSP_ITCM	HiFi4 Instruction TCM RAMs Periphery Indicates the power down status of HiFi4 instruction TCM RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
13 DSP_DCACHE	HiFi4 Data Cache RAMs Periphery Indicates the power down status of HiFi4 data cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
12 DSP_ICACHE	HiFi4 Instruction Cache RAMs Periphery Indicates the power down status of HiFi4 instruction cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
11 CPU0_SCACHE	CPU0 System Cache RAMs Periphery Indicates the power down status of CPU0 system cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 CPU0_CCACHE	CPU0 Code Cache RAMs Periphery Indicates the power down status of CPU0 code cache RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
9 DMA0_1_P_E	DMA0-1, PKC, and ETF RAMs Periphery Indicates the power down status of DMA0-1, PKC, and ETF RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
8 DMA2_3	DMA2 and DMA3 RAMs Periphery Indicates the power down status of DMA2 and DMA3 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
7 GPU	VGPU RAMs Periphery Indicates the power down status of VGPU RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
6 MIPI	MIPI PHY RAMs Periphery Indicates the power down status of MIPI PHY RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
5 PNG	PNGDEC RAMs Periphery Indicates the power down status of PNGDEC RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
4	JPEGDEC RAMs Periphery Indicates the power down status of JPEGDEC RAMs periphery.

Table continues on the next page...

Table continued from the previous page...

Field	Function
JPEG	This bit is reset by warm reset. 0b - Power on 1b - Power down
3 USB1_SRAM	USB1 RAMs Periphery Indicates the power down status of USB1 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
2 USB0_SRAM	USB0 RAMs Periphery Indicates the power down status of USB0 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
1 SDHC1_SRAM	uSDHC1 RAMs Periphery Indicates the power down status of uSDHC1 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down
0 SDHC0_SRAM	uSDHC0 RAMs Periphery Indicates the power down status of uSDHC0 RAMs periphery. This bit is reset by warm reset. 0b - Power on 1b - Power down

27.7.2.38 PD Wake-up Configuration for Sense Domain (PDWAKECFG)

Offset

Register	Offset
PDWAKECFG	E8h

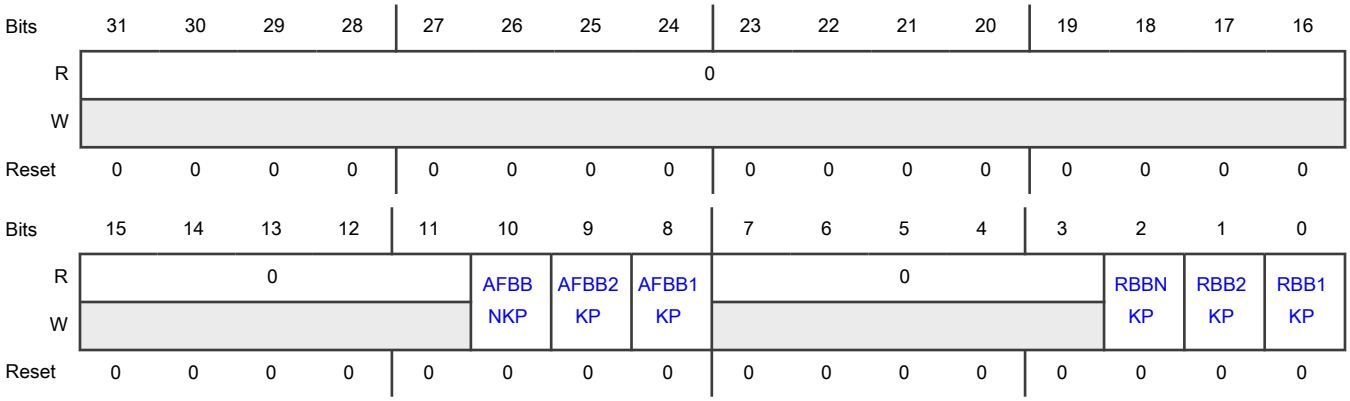
Function

Copies the selected values from the Sense slot PDSLEEPCFG0 to PDRUNCFG0 when CPU1 wakes up from Deep Sleep mode.

Power down wake-up is used when the PMC module switches from one body bias mode to another body bias mode in active mode, but it allows the change to occur while the affected domains are in Deep Sleep mode and the clocks are stopped to avoid timing issues during the transition. The clock controls need to be configured to stop in Deep Sleep mode in the domain's clock controls.

When PMC enters in Deep Sleep mode, the PDSLEEPCFG0 changes the body bias modes and the PDRUNCFG0 body bias power down controls are replaced with the PDSLEEPCFG0 controls selected in PDWAKECFG. This feature is used with INTRCTRL[AUTOWKIE] which enables the AUTOWAKE feature that wakes up PMC when the transition to Deep Sleep is completed and the updated body bias modes are active. The AUTOWAKE triggers a PMC interrupt which triggers a wake-up. Since PDRUNCFG0 was updated with the body bias state from PDSLEEPCFG0, the body bias for the selected domains is maintained.

Diagram



Fields

Field	Function
31-11 —	Reserved
10 AFBBNKP	AFBB Wake-up in VDDN Domain Copies PDSLEEPCFG0[AFBBN_PD] to PDRUNCFG0[AFBBN_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
9 AFBB2KP	AFBB Wake-up in VDD2Domain Copies PDSLEEPCFG0[AFBB2_PD] to PDRUNCFG0[AFBB2_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
8 AFBB1KP	AFBB Wake-up in VDD1 Domain Copies PDSLEEPCFG0[AFBB1_PD] to PDRUNCFG0[AFBB1_PD] on wake-up to maintain AFBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
7-3 —	Reserved
2 RBBNKP	RBB Wake-up in VDDN Domain Copies PDSLEEPCFG0[RBBN_PD] to PDRUNCFG0[RBBN_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
1 RBB2KP	RBB Wake-up in VDD2 Domain Copies PDSLEEPCFG0[RBB2_PD] to PDRUNCFG0[RBB2_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep
0 RBB1KP	RBB Wake-up in VDD1 Domain Copies PDSLEEPCFG0[RBB1_PD] to PDRUNCFG0[RBB1_PD] on wake-up to maintain RBB state. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by warm reset. 0b - Do not copy during Deep Sleep 1b - Copy during Deep Sleep

27.7.2.39 Power Domain Flags for Sense Domain (PWRFLAGS)

Offset

Register	Offset
PWRFLAGS	ECh

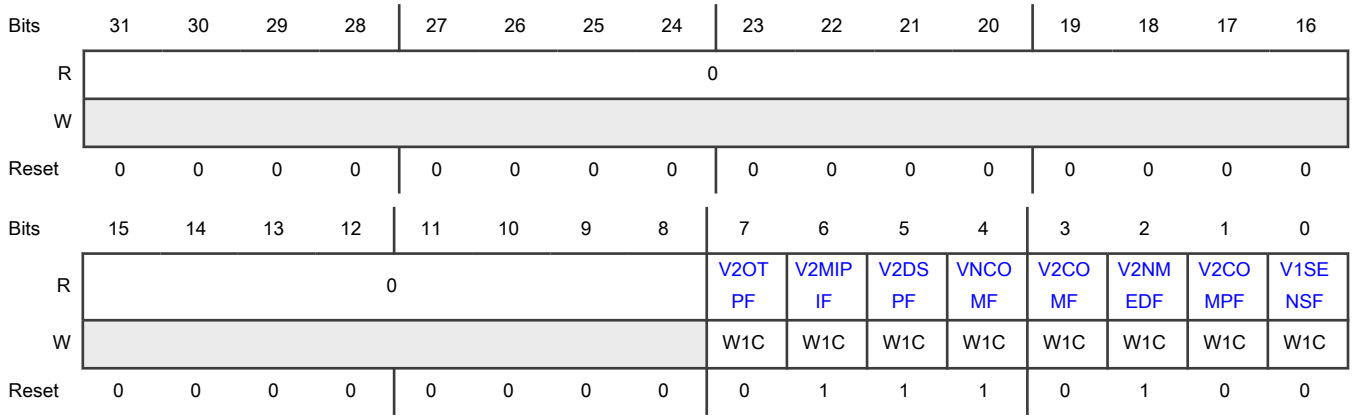
Function

Flags indicate which domains were put into Deep Sleep Retain or Power Down mode since the last time the flag was cleared by software. Flags are clearable only when the domain is powered.

NOTE

The Compute and Sense slots each have their own set of flags. Clearing a flag in the Compute slot will not clear the corresponding flag in the Sense slot.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 V2OTPF	VDD2_OTP Domain Flag Indicates if VDD2_OTP domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
6 V2MIPIF	VDD2_MIPI Domain Flag Indicates if VDD2_MIPI domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
5 V2DSPF	VDD2_DSP Domain Flag Indicates if VDD2_DSP domain enters power down mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No power-down event occurred 1b - Power-down event occurred
4 VNCOMF	VDDN_COM Domain Flag Indicates if VDDN_COM domain enters DSR mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
3 V2COMF	VDD2_COM Domain Flag Indicates if VDD2_COM domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
2 V2NMEDF	VDD2_MEDIA and VDDN_MEDIA Domain Flag Indicates if VDD2_MEDIA and VDDN_MEDIA domains enter DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
1 V2COMPF	VDD2_COMP Domain Flag Indicates if VDD2_COMP domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred
0 V1SENSF	VDD1_SENSE Domain Flag Indicates if VDD1_SENSE domain enters DSR mode. This field has 2 copies separately accessed by the Compute and Sense slots. This bit is reset by VDD1V8 POR reset. 0b - No DSR event occurred 1b - DSR event occurred

Chapter 28
Clock Controller (CLKCTL)

28.1 Chip-specific CLKCTL information

Table 234. Reference links to related information

Topic	Related module	Reference
Full description	CLKCTL	CLKCTL
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview
Signal multiplexing	IOPCTL	Signal multiplexing IO PAD controller

NOTE

The CLKCTL module includes some security functionality that is not described here. Refer to the i.MX RT700 Security Reference Manual for more details.

28.1.1 Module instances

This device has five instances of the CLKCTL modules, CLKCTL0, CLKCTL1, CLKCTL2, CLKCTL3 and CLKCTL4.

Table 235. CLKCTL instances

	CLKCTL0	CLKCTL1	CLKCTL2	CLKCTL3	CLKCTL4
Located in	VDD2_COM	VDD1_SENSE	VDDN_COM	VDD1_SENSE	VDD2_MEDIA
Access	CPU0 HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

28.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

28.2 Overview

The CLKCTL modules configure the clocks on the chip, including clock generation and clock gating for system and peripheral clocks. CLKCTL uses a distributed structure to manage domain clocks and are co-located with other system control modules like RSTCTL and SYSCON. The CLKCTL instances are illustrated in the System Diagram below.

28.2.1 System diagram

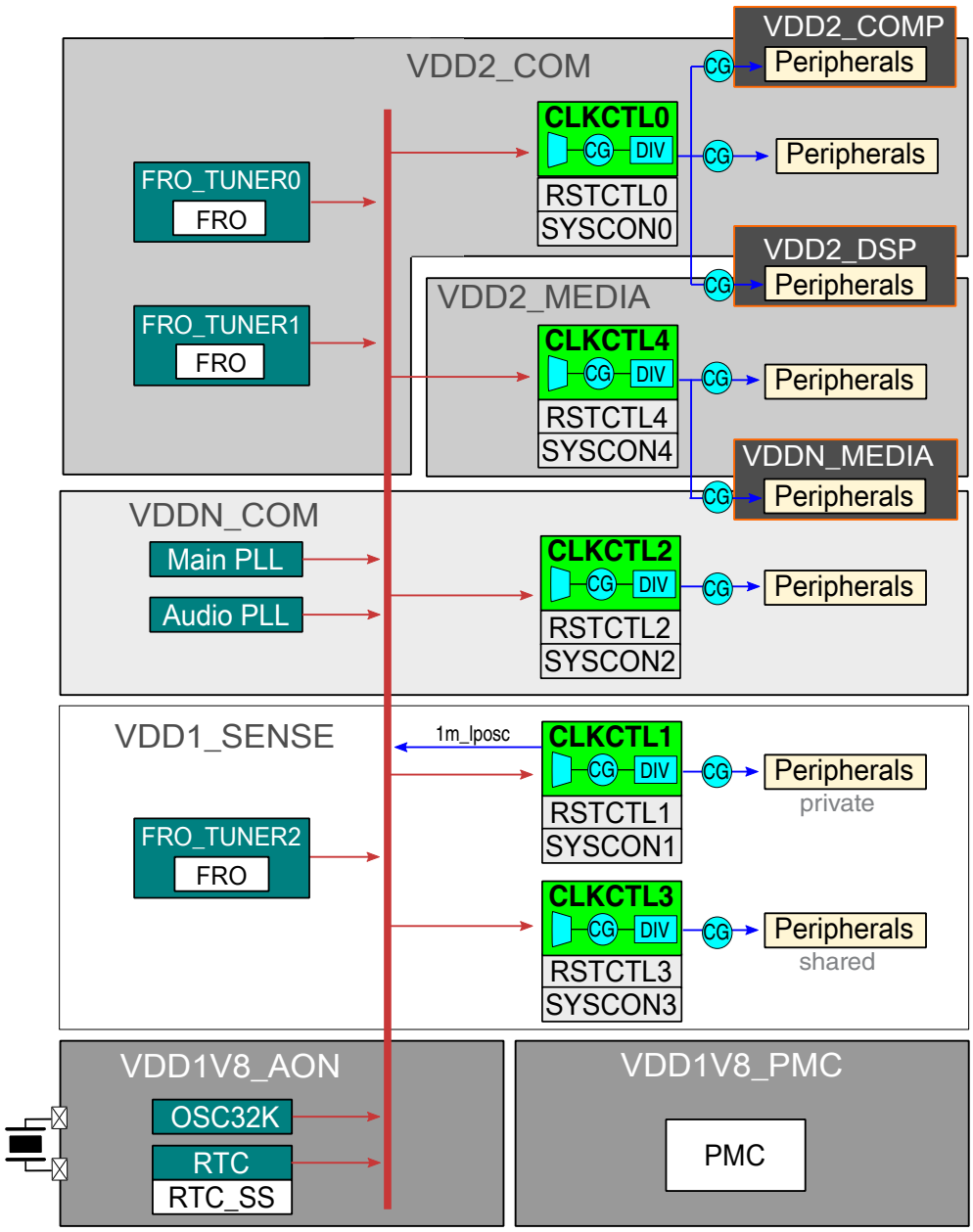


Figure 92. CLKCTL system diagram

28.2.2 Features

CLKCTL has the following features:

- Selects functional clocks from various clock sources:
  - One fixed frequency FRO (FRO1)
  - Two configurable FROs (FRO0 and FRO2)
  - 1 MHz Low-Power Internal Oscillator (LPOSC)
  - XTAL oscillator with frequency range of 4-32 MHz (osc_clk)

  - 32 kHz XTAL oscillator (osc_32k)
  - Separated PLLs (MAIN_PLL and AUDIO_PLL)
- Supports clock gating.
- Supports various clock dividers
  - Fractional PLL dividers
  - Post divider (/2, /3, /6, /8) outputs for FROs
  - Clock slice dividers

28.3 System clocks

This section covers the module clock connections on the chip, and associated clock gating options. For more information about the Clock Root configuration or clock gating enables, refer to the Clock generation section and associated clock gating registers.

NOTE

- SENSE_MAIN_CLK and SENSE_MAIN_CLK_1 are from the same clock mux (CLKCTL3 MAINCLKSEL)
- SENSE_MAIN_CLK is used for the private devices of CPU1 (which is configured by SLEEPCON_SENSE's RUNCFG[SENSEP_MAINCLK_SHUTOFF] and SLEEPCFG[SENSEP_MAINCLK_SHUTOFF])
- SENSE_MAIN_CLK_1 is used for the shared devices between CPU0 and CPU1 (which is configured by SLEEPCON_SENSE's RUNCFG[SENSES_MAINCLK_SHUTOFF] and SLEEPCFG[SENSES_MAINCLK_SHUTOFF])

Table 236. System clocks

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
ACMP0	ipg_clk	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[ACMP0]
	ipg_clk_s	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[ACMP0]
ADC0	adc_clk	SARADC_FCLK	CLKCTL3	PSCCTL0[SARADC0]
	ipg_clk	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[SARADC0]
	ipg_clk_test	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[SARADC0]
	ipg_clk_s	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[SARADC0]
CACHE64_CTRL0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[CACHE64_0]
CACHE64_CTRL1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[CACHE64_1]
CLKCTL0	pclk	COMPUTE_MAIN_CLK	-	-
CLKCTL1	pclk	SENSE_MAIN_CLK	-	-
CLKCTL2	pclk	COMMON_VDDN_CLK	-	-
CLKCTL3	pclk	SENSE_MAIN_CLK_1	-	-
CLKCTL4	pclk	MEDIA_MAIN_CLK	-	-
CMX_PERFMON0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[CMX_PERFMON0]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[CMX_PERFMON0]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
CMX_PERFMON1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[CMX_PERFMON1]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[CMX_PERFMON1]
CPU0	CLKIN	COMPUTE_MAIN_CLK	-	-
CPU1	CLKIN	SENSE_MAIN_CLK	CLKCTL3	PSCCTL0[CPU1]
CRC	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CRC0]
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CRC0]
CTIMER0	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CTIMER0]
	clk_and_reset_ctclk	CTIMER0_FCLK	CLKCTL0	PSCCTL2[CTIMER0]
CTIMER1	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CTIMER1]
	clk_and_reset_ctclk	CTIMER1_FCLK	CLKCTL0	PSCCTL2[CTIMER1]
CTIMER2	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CTIMER2]
	clk_and_reset_ctclk	CTIMER2_FCLK	CLKCTL0	PSCCTL2[CTIMER2]
CTIMER3	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CTIMER3]
	clk_and_reset_ctclk	CTIMER3_FCLK	CLKCTL0	PSCCTL2[CTIMER3]
CTIMER4	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[CTIMER4]
	clk_and_reset_ctclk	CTIMER4_FCLK	CLKCTL0	PSCCTL2[CTIMER4]
CTIMER5	clk_and_reset_pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[CTIMER5]
	clk_and_reset_ctclk	CTIMER5_FCLK	CLKCTL1	PSCCTL1[CTIMER5]
CTIMER6	clk_and_reset_pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[CTIMER6]
	clk_and_reset_ctclk	CTIMER6_FCLK	CLKCTL1	PSCCTL1[CTIMER6]
CTIMER7	clk_and_reset_pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[CTIMER7]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	clk_and_reset_ctclk	CTIMER7_FCLK	CLKCTL1	PSCCTL1[CTIMER7]
eDMA0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA0]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA0]
	dma_ram_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA0]
eDMA1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA1]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA1]
	dma_ram_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[eDMA1]
eDMA2	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA2]
	ipg_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA2]
	dma_ram_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA2]
eDMA3	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA3]
	ipg_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA3]
	dma_ram_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[eDMA3]
EZH-V	m_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[EZHV]
	nex_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[EZHV]
FLEXIO	flexio_clk	FLEXIO_FCLK	CLKCTL4	PSCCTL0_MEDIA[FLEXIO0]
	ipt_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[FLEXIO0]
	pclk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[FLEXIO0]
FREQME	pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[FREQME0]
GPIO0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO0]
GPIO1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO1]
GPIO2	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO2]
GPIO3	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO3]
GPIO4	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO4]
GPIO5	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO5]
GPIO6	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO6]
GPIO7	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[GPIO7]
GPIO8	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[GPIO8]
GPIO9	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[GPIO9]
GPIO10	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[GPIO10]
HiFi1	CLK	SENSE_DSP_CLK	CLKCTL1	PSCCTL1[HIFI1]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	PBCLK	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[HIFI1]
HiFi4	CLK	DSP_CLK	CLKCTL0	PSCCTL4[HIFI4]
	PBCLK	DSP_CLK	CLKCTL0	PSCCTL4[HIFI4]
	ATCLK	DSP_CLK	CLKCTL0	PSCCTL4[HIFI4]
I3C0	PCLK	I3C01_PCLK	CLKCTL0	PSCCTL2[I3C0]
	FCLK	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C0]
	CLK_SLOW	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C0]
	CLK_SLOW_TC	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C0]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[I3C0]
I3C1	PCLK	I3C01_PCLK	CLKCTL0	PSCCTL2[I3C1]
	FCLK	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C1]
	CLK_SLOW	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C1]
	CLK_SLOW_TC	I3C01_FCLK	CLKCTL0	PSCCTL2[I3C1]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[I3C1]
I3C2	PCLK	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[I3C2]
	FCLK	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C2]
	CLK_SLOW	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C2]
	CLK_SLOW_TC	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C2]
I3C3	PCLK	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[I3C3]
	FCLK	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C3]
	CLK_SLOW	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C3]
	CLK_SLOW_TC	I3C23_FCLK	CLKCTL1	PSCCTL1[I3C3]
INPUTMUX0	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[INPUTMUX0]
INPUTMUX1	clk_and_reset_pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[INPUTMUX1]
IOPCTL0	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL5[IOPCTL0]
IOPCTL1	ipg_clk_s	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[IOPCTL1]
IOPCTL2	ipg_clk_s	COMMON_VDDN_CLK	CLKCTL2	PSCCTL0[IOPCTL2]
JPEGDEC	jpeg_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	ips_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	rf_bitstreamfifo_ram__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	rf_bitstreamfifo_ram__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	rf_pixel_outbuf__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	rf_pixel_outbuf__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_jpeg_d_dqt_mem__ipg_ram_rw1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_huffman_d_tables_mem__ipg_ram_rw1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_jpeg_d_sfifo_mem__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_jpeg_d_sfifo_mem__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_huffman_d_sfifo_mem__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_huffman_d_sfifo_mem__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_idct_transpose_mem__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_idct_transpose_mem__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_idct_buffer_mem__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]
	u_idct_buffer_mem__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[JPEGDEC]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
LCDIF	ACLK	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[DCN]
	hclk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[DCN]
	pixelClk	DCPIXEL_FCLK	CLKCTL4	PSCCTL0_MEDIA[DCN]
	sdaClk	DCPIXEL_FCLK	CLKCTL4	PSCCTL0_MEDIA[DCN]
LP_FLEXCOMM[x] ([x] - 0 to 13)	hclk_gated[0]	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	hclk_gated[1]	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	hclk_gated[2]	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	hclk_gated[3]	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_clk	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[0]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[1]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[2]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[3]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[4]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk[5]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	fc_gated_clk_n[4]	FLEXCOMM[x]_FCLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[LP_FLEXCOMM[x]]
LP_FLEXCOMM[y] ([y] - 17 to 20)	hclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	hclk_gated[0]	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	hclk_gated[1]	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	hclk_gated[2]	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	hclk_gated[3]	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_clk	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[0]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[1]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[2]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[3]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[4]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk[5]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]
	fc_gated_clk_n[4]	FLEXCOMM[y]_FCLK	CLKCTL1	PSCCTL1[LP_FLEXCOMM[y]]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
LPI2C15 (LP_FLEXCOMM15)	ipg_clk	SENSE_MAIN_CLK_1	CLKCTL0	PSCCTL2[LPI2C]
	ipg_clk_s	SENSE_MAIN_CLK_1	CLKCTL0	PSCCTL2[LPI2C]
	lpi2c_clk	LPI2C_FCLK	CLKCTL0	PSCCTL2[LPI2C]
	lpi2c_div_clk	LPI2C_FCLK	CLKCTL0	PSCCTL2[LPI2C]
LPSPI14 (LP_FLEXCOMM14)	ipt_clk	MEDIA_MAIN_CLK	-	-
	ipg_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI14] (LP_FLEXCOMM14)
	ipg_clk_s	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI14] (LP_FLEXCOMM14)
	lpspi_clk	LPSPI14_FCLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI14] (LP_FLEXCOMM14)
	lpspi_div_clk	LPSPI14_FCLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI14] (LP_FLEXCOMM14)
LPSPI16 (LP_FLEXCOMM16)	ipg_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI16] (LP_FLEXCOMM16)
	ipg_clk_s	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI16] (LP_FLEXCOMM16)
	lpspi_clk	LPSPI16_FCLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI16] (LP_FLEXCOMM16)
	lpspi_div_clk	LPSPI16_FCLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI16] (LP_FLEXCOMM16)
	ipt_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[LPSPI16] (LP_FLEXCOMM16)
MICFIL	ipg_clk	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_s	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_app_nonstop	MICFIL_FCLK	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_app	MICFIL_FCLK	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_app_free	MICFIL_FCLK	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_decfil	MICFIL_FCLK	CLKCTL3	PSCCTL0[MICFIL]
	ipg_clk_decfil_free	MICFIL_FCLK	CLKCTL3	PSCCTL0[MICFIL]
MIPI_DSI_HOST	pclk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]
	BITCLK	DPHY_BIT_CLK	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	TxClkEsc	DPHY_TX_CLK	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]
	RxClkEsc	DPHY_RX_CLK	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]
	dpi_pclk	DISPLAY0CLK_TMUXED	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]
	scan_clk_in	DPHY_FCLK	CLKCTL4	PSCCTL0_MEDIA[MIPI_DSI_HOST]
MMU0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU0]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU0]
	hclk_gated	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU0]
MMU1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU1]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU1]
	hclk_gated	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[MMU1]
MMU2	hclk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[MMU2]
	ipg_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[MMU2]
	hclk_gated	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[MMU2]
MRT0	vpb_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[MRT0]
MRT1	vpb_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[MRT1]
MU0	ipg_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
	ipg_clk_s_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
	ipt_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
	ipg_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
	ipg_clk_s_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
	ipt_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU0]
MU1	ipg_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
	ipg_clk_s_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
	ipt_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
	ipg_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
	ipg_clk_s_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
	ipt_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU1]
MU2	ipg_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]
	ipg_clk_s_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	ipt_clk_mua	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]
	ipg_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]
	ipg_clk_s_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]
	ipt_clk_mub	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[MU2]
MU3	ipt_clk_mua	SENSE_MAIN_CLK	-	-
	ipt_clk_mub	SENSE_MAIN_CLK	-	-
	ipg_clk_mua	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[MU3]
	ipg_clk_s_mua	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[MU3]
	ipg_clk_mub	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[MU3]
	ipg_clk_s_mub	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[MU3]
MU4	ipt_clk_mua	COMPUTE_MAIN_CLK	-	-
	ipg_clk_mua	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[MU4]
	ipg_clk_s_mua	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[MU4]
	ipg_clk_mub	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[MU4]
	ipg_clk_s_mub	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[MU4]
NPU	clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL5[NPU0]
OCOTP	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[VDD2_OTP0]
OSTIMER	hclk	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[OSTIMER]
	ostimer_clk	OSEVENT_FCLK	CLKCTL3	PSCCTL0[OSTIMER]
	capture_clk[0]	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[OSTIMER]
	capture_clk[1]	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[OSTIMER]
	capture_clk[2]	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[OSTIMER]
	capture_clk[3]	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[OSTIMER]
PINT0	clk_and_reset_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[PINT0]
PINT1	clk_and_reset_pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[PINT1]
PMC	ipg_clk	CLK_16MHZ_1P8	-	-
	ipg_clk_s	CLK_16MHZ_1P8	-	-
PNGDEC	clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	input_fifo__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	input_fifo__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	output_fifo__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	output_fifo__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	ancillary_fifo__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	ancillary_fifo__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	line_buffer__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	line_buffer__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	lz77_win_ram__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	lz77_win_ram__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	inflate_buffer__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	inflate_buffer__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	palette_ram__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	palette_ram__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_codes_ram__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_codes_ram__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_lengths_ram__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_lengths_ram__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	huffman_literal_ram0__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_literal_ram0__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_literal_ram1__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_literal_ram1__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_distance_ram0__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_distance_ram0__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_distance_ram1__ipg_ram_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_distance_ram1__ipg_ram_w2_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_literal_rom0__ipg_rom_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
	huffman_literal_rom1__ipg_rom_r1_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[PNGDEC]
PVTS0	pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[PVTS0]
	main_clk0	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL3[PVTS0]
	main_clk1	DSP_CLK	CLKCTL0	PSCCTL3[PVTS0]
PVTS1	pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[PVTS1]
	main_clk0	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[PVTS1]
	main_clk1	SENSE_DSP_CLK	CLKCTL1	PSCCTL1[PVTS1]
RSTCTL0	pclk	COMPUTE_MAIN_CLK	-	-
RSTCTL1	pclk	SENSE_MAIN_CLK	-	-
RSTCTL2	pclk	COMMON_VDDN_CLK	-	-

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
RSTCTL3	pclk	SENSE_MAIN_CLK_1	-	-
RSTCTL4	pclk	MEDIA_MAIN_CLK	-	-
RTC0	ipg_clk_hp	RTC_FCLK	-	-
	rtc_irc_clk	RTC_FCLK	-	-
	ipg_clk_hp_s	1m_lposc	-	-
	ipg_clk_lp_s	1m_lposc	-	-
	ipg_clk	1m_lposc	-	-
	rtc_osc_32k_in_clk_hp	osc_32k	-	-
	rtc_osc_32k_in_clk_lp	osc_32k	-	-
RTC1	ipg_clk_hp	RTC_FCLK	-	-
	rtc_irc_clk	RTC_FCLK	-	-
	ipg_clk_hp_s	1m_lposc	-	-
	ipg_clk_lp_s	1m_lposc	-	-
	ipg_clk	1m_lposc	-	-
	rtc_osc_32k_in_clk_hp	osc_32k	-	-
	rtc_osc_32k_in_clk_lp	osc_32k	-	-
SAI0	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI0]
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI0]
	ipg_clk_sai_mclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI0]
	ipt_clk_sai_bclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI0]
	ipt_clk_sai_bclk_b	SAI012_CLK	CLKCTL0	PSCCTL2[SAI0]
SAI1	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI1]
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI1]
	ipg_clk_sai_mclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI1]
	ipt_clk_sai_bclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI1]
	ipt_clk_sai_bclk_b	SAI012_CLK	CLKCTL0	PSCCTL2[SAI1]
SAI2	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI2]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SAI2]
	ipg_clk_sai_mclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI2]
	ipt_clk_sai_bclk	SAI012_CLK	CLKCTL0	PSCCTL2[SAI2]
	ipt_clk_sai_bclk_b	SAI012_CLK	CLKCTL0	PSCCTL2[SAI2]
SAI3	ipg_clk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[SAI3]
	ipg_clk_s	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[SAI3]
	ipg_clk_sai_mclk	SAI3_CLK	CLKCTL1	PSCCTL1[SAI3]
	ipt_clk_sai_bclk	SAI3_CLK	CLKCTL1	PSCCTL1[SAI3]
	ipt_clk_sai_bclk_b	SAI3_CLK	CLKCTL1	PSCCTL1[SAI3]
SCT0	clk_and_reset_hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[SCT0]
	clk_and_reset_pmux_pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[SCT0]
	clk_and_reset_pllclk	SCT_FCLK	CLKCTL0	PSCCTL1[SCT0]
SDADC	APB_CLK	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[SDADC0]
	MCLK	SDADC_FCLK	CLKCTL3	PSCCTL0[SDADC0]
SEMA42_0	clk	SENSE_MAIN_CLK	CLKCTL3	PSCCTL0_COMP[SEMA42_0]
SEMA42_3	clk	SENSE_MAIN_CLK_1	CLKCTL1	PSCCTL1[SEMA42_3]
SEMA42_4	clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[SEMA42_4]
SLEEPCON0	pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[SLEEPCON0]
	pclk_regbank	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[SLEEPCON0]
	clkin	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[SLEEPCON0]
SLEEPCON1	pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL0[SLEEPCON1]
	pclk_regbank	SENSE_MAIN_CLK	CLKCTL1	PSCCTL0[SLEEPCON1]
	clkin	SENSE_MAIN_CLK	CLKCTL1	PSCCTL0[SLEEPCON1]
SYSCON0	pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[SYSCON0]
SYSCON1	pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL0[SYSCON1]
SYSCON2	pclk	COMMON_VDDN_CLK	CLKCTL2	PSCCTL0[SYSCON2]
SYSCON3	pclk	SENSE_MAIN_CLK_1	CLKCTL3	PSCCTL0[SYSCON3]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
SYSCON4	pclk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_COMP[SYSCON4]
TIMESTAMP	CPU1_clk	SENSE_MAIN_CLK	-	-
	HiFi1_clk	SENSE_DSP_CLK	CLKCTL1	PSCCTL1[HIFI1]
	HiFi4_clk	DSP_CLK	CLKCTL0	PSCCTL4[HIFI4]
USB0	ipg_ahb_clk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB0]
	ipg_clk_s	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB0]
	ipg_clk_32khz	USB_WAKE_CLK	-	-
USB1	ipg_ahb_clk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB1]
	ipg_clk_s	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB1]
	ipg_clk_32khz	eUSB_WAKE_CLK	-	-
USBPHY0	ipg_clk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB0]
	ipg_clk_s	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[USB0]
	clk_xtal	OSC_24M	CLKCTL4	PSCCTL1_MEDIA[USB0]
	ckil_32k	USB_WAKE_CLK	-	-
eUSBPHY	CORECLKIN	OSC_24M	-	-
uSDHC0	hclk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC0]
	ipg_clk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC0]
	ipg_clk_s	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC0]
	ipg_clk_perclk	SDIO0_FCLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC0]
	ipg_clk_lp	CKIL_32K	-	-
uSDHC1	hclk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC1]
	ipg_clk	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC1]
	ipg_clk_s	MEDIA_VDDN_CLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC1]
	ipg_clk_perclk	SDIO1_FCLK	CLKCTL4	PSCCTL1_MEDIA[uSDHC1]
	ipg_clk_lp	CKIL_32K	-	-
UTICK0	pclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[UTICK0]
	timer_clk	UTICK0_FCLK	CLKCTL0	PSCCTL2[UTICK0]
UTICK1	pclk	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[UTICK1]
	timer_clk	UTICK1_FCLK	CLKCTL1	PSCCTL1[UTICK1]
VGPU	HCLK	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[VGPU]
	ACLK	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[VGPU]
	clk1x	GPU_FCLK	CLKCTL4	PSCCTL0_MEDIA[VGPU]

Table continues on the next page...

Table 236. System clocks (continued)

Peripheral	Peripheral Clock	Clock Root	Instance	Peripheral Clock Gate Register
	clkIn	GPU_FCLK	CLKCTL4	PSCCTL0_MEDIA[VGPU]
WWDT0	PCLK	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[WWDT0]
	wdclk	WDT0_FCLK	CLKCTL0	PSCCTL2[WWDT0]
WWDT1	PCLK	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL2[WWDT1]
	wdclk	WDT1_FCLK	CLKCTL0	PSCCTL2[WWDT1]
WWDT2	PCLK	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[WWDT2]
	wdclk	WDT2_FCLK	CLKCTL1	PSCCTL1[WWDT2]
WWDT3	PCLK	SENSE_MAIN_CLK	CLKCTL1	PSCCTL1[WWDT3]
	wdclk	WDT3_FCLK	CLKCTL1	PSCCTL1[WWDT3]
XCACHE0	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[SYSTEM_CACHE]
XCACHE1	hclk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL0[CODE_CACHE]
XSPI0	ahb_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI0]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI0]
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI0]
	ipg_clk_sfck	XSPI0_FCLK_DIV2	CLKCTL0	PSCCTL1[XSPI0]
	ipg_clk_sfif	XSPI0_FCLK_DIV2	CLKCTL0	PSCCTL1[XSPI0]
	ipg_clk_2xsfif	XSPI0_FCLK	CLKCTL0	PSCCTL1[XSPI0]
XSPI1	ahb_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI1]
	ipg_clk	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI1]
	ipg_clk_s	COMPUTE_MAIN_CLK	CLKCTL0	PSCCTL1[XSPI1]
	ipg_clk_sfck	XSPI1_FCLK_DIV2	CLKCTL0	PSCCTL1[XSPI1]
	ipg_clk_sfif	XSPI1_FCLK_DIV2	CLKCTL0	PSCCTL1[XSPI1]
	ipg_clk_2xsfif	XSPI1_FCLK	CLKCTL0	PSCCTL1[XSPI1]
XSPI2	ahb_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[XSPI2]
	ipg_clk	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[XSPI2]
	ipg_clk_s	MEDIA_MAIN_CLK	CLKCTL4	PSCCTL0_MEDIA[XSPI2]
	ipg_clk_sfck	XSPI2_FCLK_DIV2	CLKCTL4	PSCCTL0_MEDIA[XSPI2]
	ipg_clk_sfif	XSPI2_FCLK_DIV2	CLKCTL4	PSCCTL0_MEDIA[XSPI2]
	ipg_clk_2xsfif	XSPI2_FCLK	CLKCTL4	PSCCTL0_MEDIA[XSPI2]

28.4 Functional description

The following sections describe the clock generation from source to target (peripheral) clocks, covering the related configuration details and programming of the CLKCTL logic and clock mapping of the CLKCTL module instances.

28.4.1 Clock generation

Clock generation is based on a distributed set of clock sources used to generate clock roots across the chip domains. Clock roots are the source clocks for the distributed network of branch clocks and clock gates that supply all the peripheral clocks on the chip. The sections below cover the associated clock generation logic and configuration information.

28.4.1.1 Clock sources

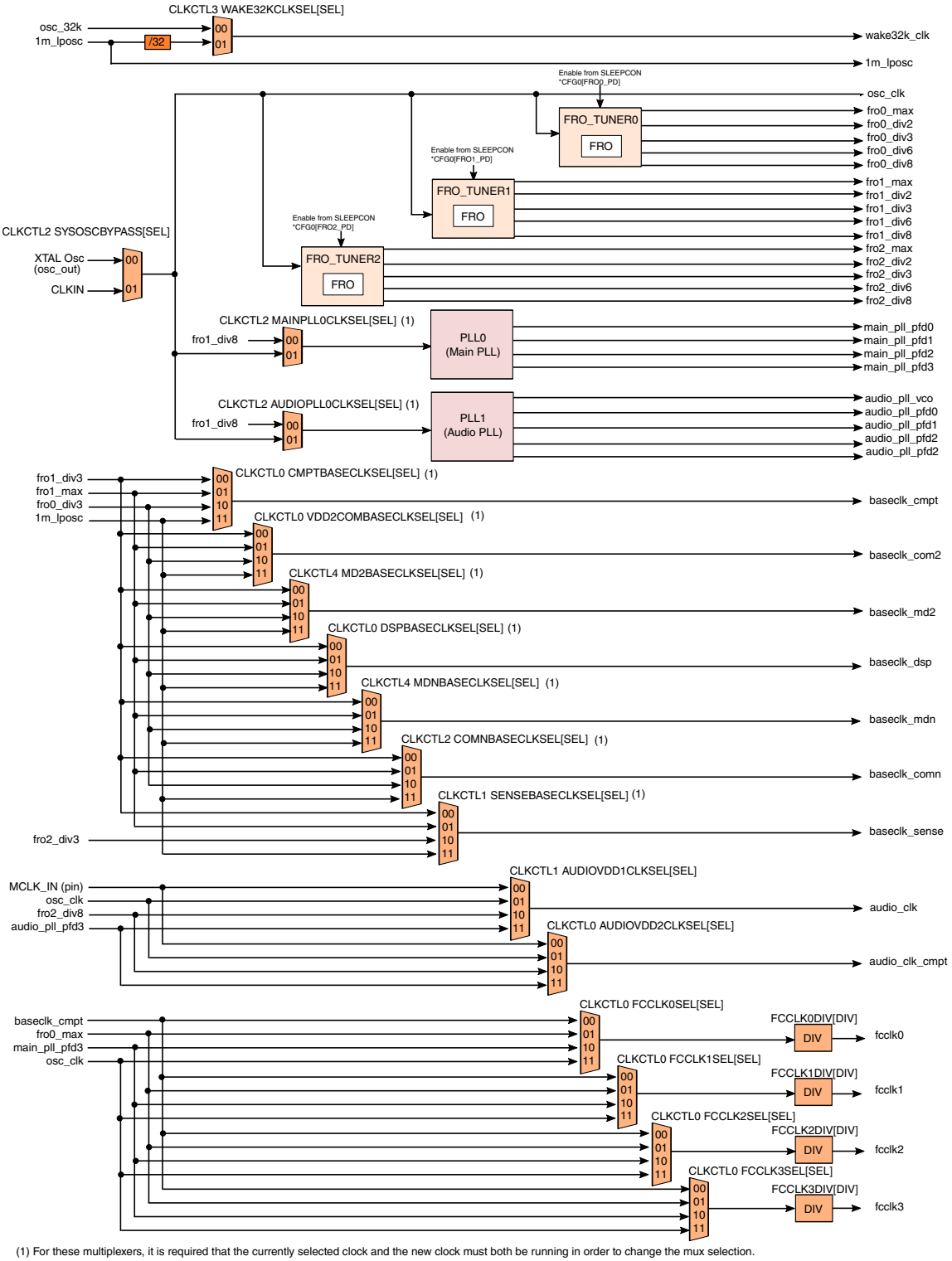
Clock sources include:

- Dynamic FRO
  - Two configurable FROs (150 MHz to 325 MHz) named as FRO0 and FRO2 respectively
  - One fixed frequency FRO (192 MHz) named as FRO1 used for bootup
  - Each FRO has post dividers with divide by 2, 3, 6, and 8 (div2, div3, div6, div8) clock outputs.
  - Digital feedback control block for frequency tuning

NOTE

You can disable fro_clk output and each of the divided outputs individually.

- XTAL Oscillator
- External clock (CLKIN)
- Main PLL
  - Used when high frequencies (> 300MHz) are required, as well as strict clocking specifications, such as jitter and duty cycle are needed.
- Audio PLL
  - Used for audio purposes with strict jitter requirements. The clock inputs and acceptable input frequencies are the same as with the Main PLL.



(1) For these multiplexers, it is required that the currently selected clock and the new clock must both be running in order to change the mux selection.

Figure 93. Clock source diagram

Table 237. Clock sources and generated clock sources

Clock source	Clock source select register	Description
Oscillators		
osc_clk	CLKCTL2.SYSOSCBYPASS	An internal clock that comes from either the CLKIN pin function or the main crystal oscillator. When used, the CLKIN pin function must be connected to a pin by selecting it in the IOPCTL block. osc_clk may be used as a clock source for the main or audio PLLs, FRO tunning clock, common_vddn_clk, and selected peripheral functions including ADC0, OSTIMER, systick and I3C FCLK.
osc_32k	See OSC32KNP chapter for more information	32 kHz output of the RTC oscillator. The 32 kHz clock must be enabled in the OSC32KHZCTL0 register. AON domain clock source's control is not in CLKCTL.
1m_lposc	-	Internal 1 MHz LPOSC that can be used as a low speed or low power system clock to drive selected peripheral functions.
wake32k_clk	CLKCTL3.WAKE32KCLKSEL	Ultra-lower power background clock used by several peripherals. It is selectable from 32k_clk or lp_32k.
ckil_32k	-	This clock only has one source that is wake32k_clk, and it can be gated by CKIL_32K_GATE. It provides the 32K clock required by SDIO0 and SDIO1.
LOW_FREQ_CLKOUT and LOW_FREQ_CLKOUT_N	-	A low-frequency clock-out and its compliment. The low frequency clock is created by routing the 1 kHz output from the 32 kHz clock module through a clock divider. This will provide a clock in the range of 4 Hz to 1 kHz. These two clock pins are multiplexed with two of the 16 FLEXIO pins. The divided 1 kHz clock should go down to 0.5 Hz, which requires 16-bit dividers
Main PLL		
main_pll_pfd0	CLKCTL2.MAINPLL0CLKSEL	PFD0 of the main PLL. It is one clock source of compute_main_clk, dsp_clk, media_main_clk, media_vddn_clk and common_ram_clk, as well as selected peripheral functions including VGPU, dphy_bit_clk, tpiu_clk, sct, i3c01_fclk and lpspi16_fclk
main_pll_pfd1		PFD1 of the main PLL. It is one clock source of dsp_clk and selected peripheral functions including XSPI0, SDIO1, ADC0 and dphy_txrx_clk
main_pll_pfd2		PFD2 of the main PLL. It is one clock source of media_main_clk and media_vddn_clk, as well as selected peripheral functions including XSPI1, SDIO0, VGPU and dcpixel
main_pll_pfd3		PFD3 of the main PLL. It is one clock source of common_vddn_clk, and selected peripheral functions including XSPI2 and FLEXIO

Table continues on the next page...

Table 237. Clock sources and generated clock sources (continued)

Clock source	Clock source select register	Description
Audio PLL		
audio_pll_vco	CLKCTL2.AUDIOPLL0CLKSEL	VCO of the audio PLL. It is one source of SDADC
audio_pll_pfd0		PFD0 of the audio PLL. It is one source of selected peripheral functions including SDIO0, SDIO1 and XSPI2
audio_pll_pfd1		PFD1 of the audio PLL. It is one clock source of compute_main_clk and sense_dsp_clk, as well as selected peripheral functions including XSPI1, XSPI2 and dcpixel
audio_pll_pfd2		PFD2 of the audio PLL. It is one clock source of sense_ram_clk, and selected peripheral functions including dphy_bit_clk and dphy_txrx_clk
audio_pll_pfd3		PFD3 of the audio PLL. It is one clock source of sense_main_clk and selected peripheral functions including MICFIL
FRO_TUNER0		
fro0_max	CLKCTL2.SYSOSCBYPASS See FRO_TUNER chapter for more information. **NOTE:** FRO has a reference clock to auto trim the FRO output frequency, osc_clk is used for this clock. and osc_clk has two clock sources which can be selected by SYSOSCBYPASS.	Output of FRO_TUNER0 clock
fro0_div2		Output of FRO_TUNER0 clock divided by 2
fro0_div3		Output of FRO_TUNER0 clock divided by 3
fro0_div6		Output of FRO_TUNER0 clock divided by 6
fro0_div8		Output of FRO_TUNER0 clock divided by 8
FRO_TUNER1		
fro1_max	CLKCTL2.SYSOSCBYPASS See FRO_TUNER chapter for more information. **NOTE:** FRO has a reference clock to auto trim the FRO output frequency, osc_clk is used for this clock. and osc_clk has two clock sources which can be selected by SYSOSCBYPASS.	Output of FRO_TUNER1 clock
fro1_div2		Output of FRO_TUNER1 clock divided by 2
fro1_div3		Output of FRO_TUNER1 clock divided by 3
fro1_div6		Output of FRO_TUNER1 clock divided by 6
fro1_div8		Output of FRO_TUNER1 clock divided by 8
FRO_TUNER2		
fro2_max	CLKCTL2.SYSOSCBYPASS See FRO_TUNER chapter for more information. **NOTE:** FRO has a reference clock to auto trim the FRO output	Output of FRO_TUNER2 clock
fro2_div2		Output of FRO_TUNER2 clock divided by 2
fro2_div3		Output of FRO_TUNER2 clock divided by 3
fro2_div6		Output of FRO_TUNER2 clock divided by 6

Table continues on the next page...

Table 237. Clock sources and generated clock sources (continued)

Clock source	Clock source select register	Description
fro2_div8	frequency, osc_clk is used for this clock. and osc_clk has two clock sources which can be selected by SYSOSCBYPASS.	Output of FRO_TUNER2 clock divided by 8
Generated Clock Sources		
baseclk_cmpt	CLKCTL0.CMPTBASECLKSEL	Default clock source of VDD2_COMP domain clock slices
baseclk_sense	CLKCTL1.SENSEBASECLKSEL	Default clock source of VDD1_SENSE domain clock slices
baseclk_dsp	CLKCTL0.DSPBASECLKSEL	Default clock source of VDD2_DSP domain clock slices
baseclk_com2	CLKCTL0.COM2BASECLKSEL	Default clock source of VDD2_COM domain clock slices
baseclk_comn	CLKCTL2.COMNBASECLKSEL	Default clock source of VDDN_COM domain clock slices
baseclk_mdn	CLKCTL4.MDNBASECLKSEL	Default clock source of VDDN_MEDIA domain clock slices
baseclk_md2	CLKCTL4.MD2BASECLKSEL	Default clock source of VDD2_MEDIA domain clock slices
audio_clk	CLKCTL1.AUDIOVDD1CLKSEL	Internal clock of CLKCTL that may be used as the function clock of SAI, CTIMER and/or MICFIL peripheral.
audio_clk_cmpt	CLKCTL0.AUDIOVDD2CLKSEL	Internal clock of CLKCTL that may be used as the function clock of SAI, and CTIMER peripherals.
fcclk0	CLKCTL0.FCCLK0SEL	Internal clock of CLKCTL that is used as the function clock of any Flexcomm Interface in compute VDD2 domain
fcclk1	CLKCTL0.FCCLK1SEL	
fcclk2	CLKCTL0.FCCLK2SEL	
fcclk3	CLKCTL0.FCCLK3SEL	

28.4.1.2 Clock roots

Each clock slice shown in Figure 92 creates a clock root.

The following tables describe clock root signals shown in the clocking diagrams. See I.MX RT700 Crossover Microcontrollers Data Sheet for clock root frequency data.

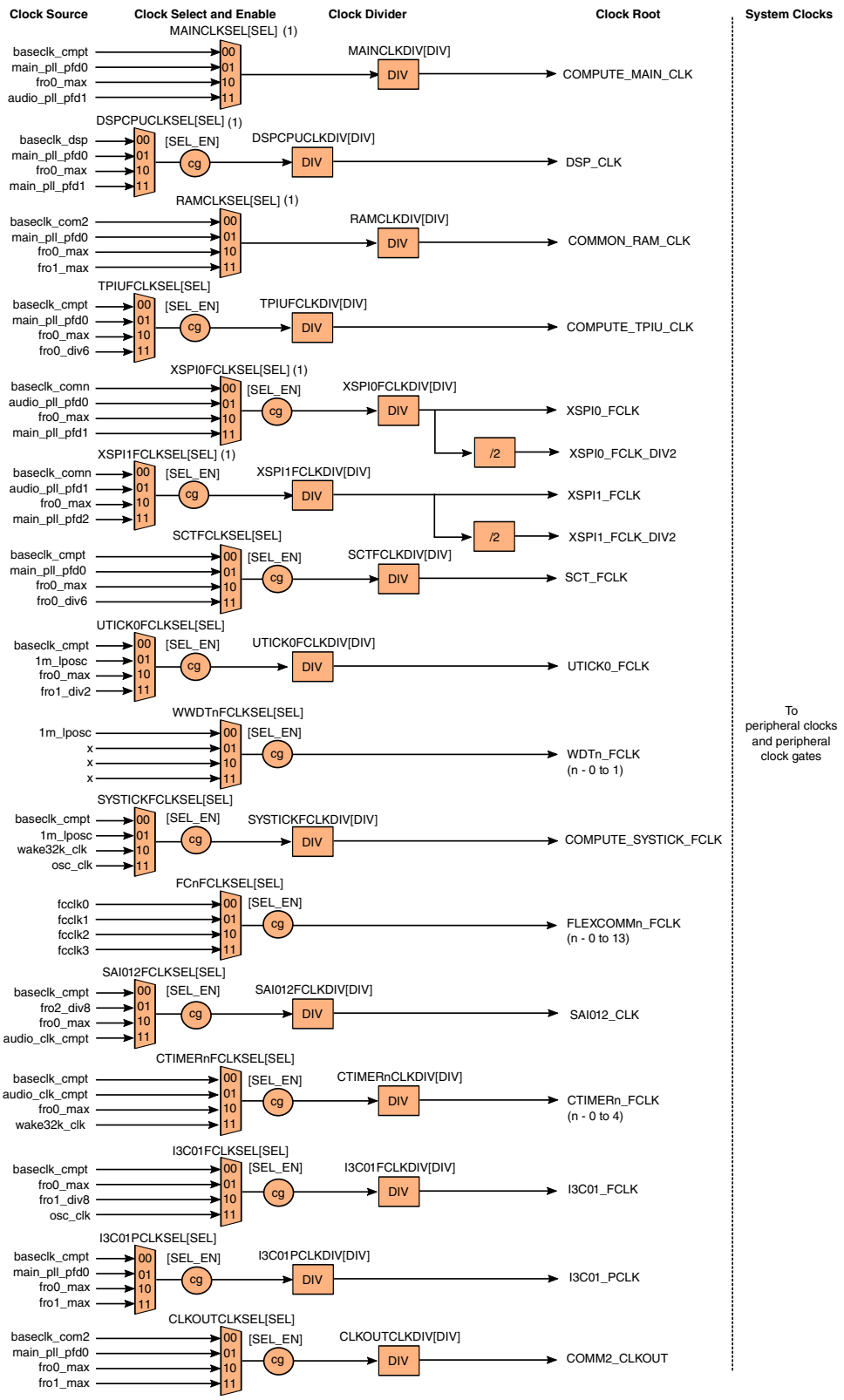
NOTE

The clock source select registers contains MUX output enables (*SEL_EN bitfields) that can disable clock root outputs at the clock source MUX. Disabling the clock root may be useful to reduce power, if the branch clocks sourcing downstream peripherals are unused. If some clocks are used downstream, local clock gating can be used to still save power by gating individual clocks using the associated CLKCTL PSCCTL* register.

28.4.1.2.1 CLKCTL0 clock tree and clock roots

28.4.1.2.1.1 CLKCTL0 clock tree

The following diagram illustrates the clock tree for the CLKCTL0 instance. Detailed configuration information of these clock roots can be found in the CLKCTL0 clock roots section.



(1) For these multiplexers, it is required that the currently selected clock and the new clock must both be running in order to change the mux selection.

Figure 94. CLKCTL0 clock tree

28.4.1.2.1.2 CLKCTL0 clock roots

Table 238. CLKCTL0 clock roots

Clock source select register	Clock root	Description
MAINCLKSEL	COMPUTE_MAIN_CLK	Derives the CPU clock of VDD2_COMP domain
DSPCPUCLKSEL	DSP_CLK	Derives the HiFi4 DSP CPU clock of VDD2_DSP domain
RAMCLKSEL	COMMON_RAM_CLK	Derives the clock of RAM in VDD2_COM domain for async access
XSPI0FCLKSEL	XSPI0_FCLK	Derives the clock of XSPI0
XSPI1FCLKSEL	XSPI1_FCLK	Derives the clock of XSPI1
SYSTICKFCLKSEL	COMPUTE_SYSTICK_FCLK	Derives the systick clock for VDD2_COMP domain
TPIUFCLKSEL	COMPUTE_TPIU_CLK	Derives the functional clock for the TPIU
SCTFCLKSEL	SCT_FCLK	Derives the functional clock for the SCT
SAI012FCLKSEL	SAI012_clk	Derives the clock for the SAI0-SAI2
I3C01PCLKSEL	I3C01_PCLK	Derives the clock for I3C0 and I3C1 P-CLK
I3C01FCLKSEL	I3C01_FCLK	Derives the I3C0 and I3C1 functional clocks
UTICK0FCLKSEL	UTICK0_FCLK	Derives the UTICK0 clock
CTIMER0FCLKSEL	CTIMER0_FCLK	Derives the CTIMER clocks
CTIMER1FCLKSEL	CTIMER1_FCLK	
CTIMER2FCLKSEL	CTIMER2_FCLK	
CTIMER3FCLKSEL	CTIMER3_FCLK	
CTIMER4FCLKSEL	CTIMER4_FCLK	
FC0FCLKSEL	FLEXCOMM0_FCLK	Derives the FLEXCOMM clocks
FC1FCLKSEL	FLEXCOMM1_FCLK	
FC2FCLKSEL	FLEXCOMM2_FCLK	
FC3FCLKSEL	FLEXCOMM3_FCLK	
FC4FCLKSEL	FLEXCOMM4_FCLK	
FC5FCLKSEL	FLEXCOMM5_FCLK	
FC6FCLKSEL	FLEXCOMM6_FCLK	
FC7FCLKSEL	FLEXCOMM7_FCLK	
FC8FCLKSEL	FLEXCOMM8_FCLK	
FC9FCLKSEL	FLEXCOMM9_FCLK	

Table continues on the next page...

Table 238. CLKCTL0 clock roots (continued)

Clock source select register	Clock root	Description
FC10FCLKSEL	FLEXCOMM10_FCLK	
FC11FCLKSEL	FLEXCOMM11_FCLK	
FC12FCLKSEL	FLEXCOMM12_FCLK	
FC13FCLKSEL	FLEXCOMM13_FCLK	
CLKOUTCLKSEL	COMM2_CLKOUT	Derives the clock out of VDD2_COM
WDT0FCLKSEL	WDT0_FCLK	Derives the clock for watchdog timers
WDT1FCLKSEL	WDT1_FCLK	

28.4.1.2.2 CLKCTL1 clock tree and clock roots

28.4.1.2.2.1 CLKCTL1 clock tree

The following diagram illustrates the clock tree for the CLKCTL1 instance. Detailed configuration information of these clock roots can be found in the CLKCTL1 clock roots section.

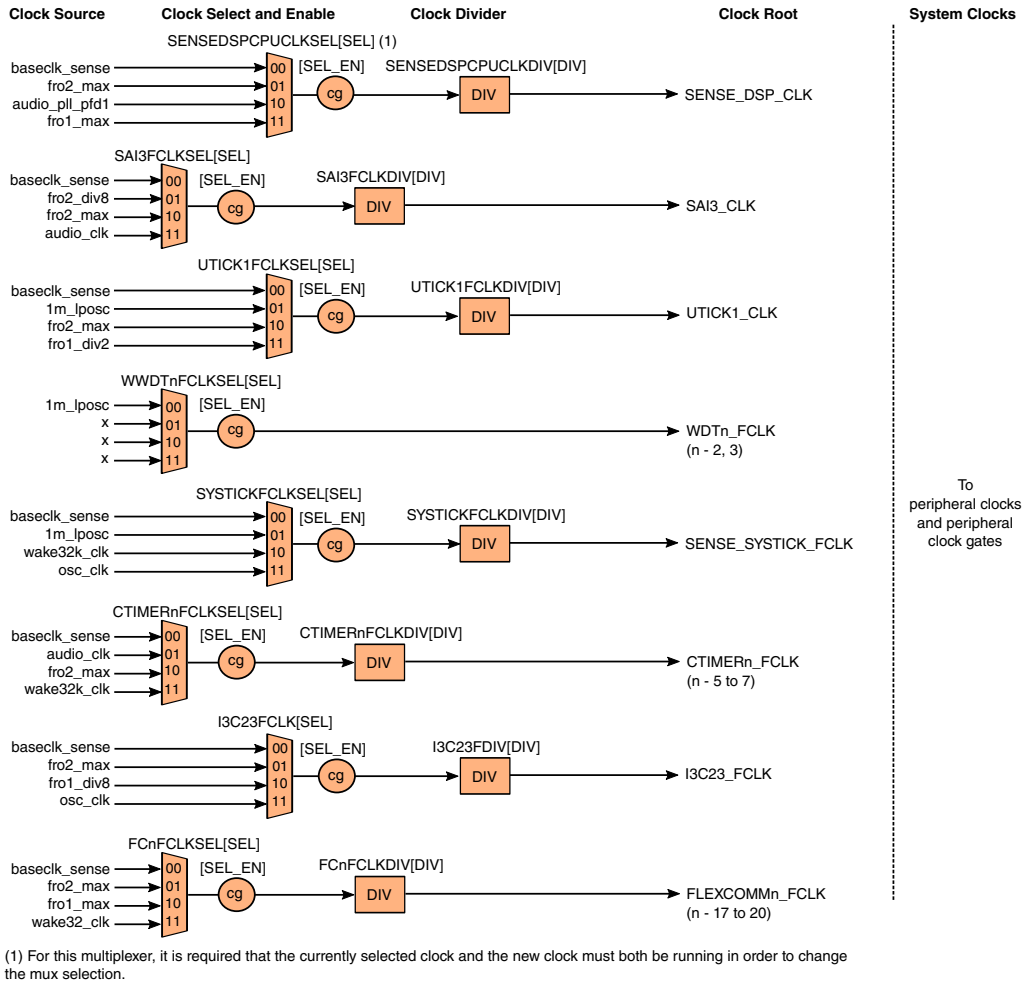


Figure 95. CLKCTL1 clock tree

28.4.1.2.2.2 CLKCTL1 clock roots

Table 239. CLKCTL1 clock roots

Clock source select register	Clock root	Description
SENSEDSPCPUCLKSEL	SENSE_DSP_CLK	Derives the DSP CPU clock of DSP in VDD1_SENSE domain
SYSTICKFCLKSEL	SENSE_SYSTICK_FCLK	Derives the systick clock for VDD1_SENSE domain
SAI3FCLKSEL	SAI3_CLK	Derives the clock for the SAI3
I3C23FCLKSEL	I3C23_FCLK	Derives the I3C2 and I3C3 functional clocks
UTICK1FCLKSEL	UTICK1_FCLK	Derives the UTICK1 clock
CTIMER5FCLKSEL	CTIMER5_FCLK	Derives the CTIMER clocks

Table continues on the next page...

Table 239. CLKCTL1 clock roots (continued)

Clock source select register	Clock root	Description
CTIMER6FCLKSEL	CTIMER6_FCLK	
CTIMER7FCLKSEL	CTIMER7_FCLK	
FC17FCLKSEL	FLEXCOMM17_FCLK	Derives the FLEXCOMM clocks
FC18FCLKSEL	FLEXCOMM18_FCLK	
FC19FCLKSEL	FLEXCOMM19_FCLK	
FC20FCLKSEL	FLEXCOMM20_FCLK	
WDT2FCLKSEL	WDT2_FCLK	Derives the clock for watchdog timers
WDT3FCLKSEL	WDT3_FCLK	

28.4.1.2.3 CLKCTL2 clock tree and clock roots

28.4.1.2.3.1 CLKCTL2 clock tree

The following diagram illustrates the clock tree for the CLKCTL2 instance. Detailed configuration information of these clock roots can be found in the CLKCTL2 clock roots section.

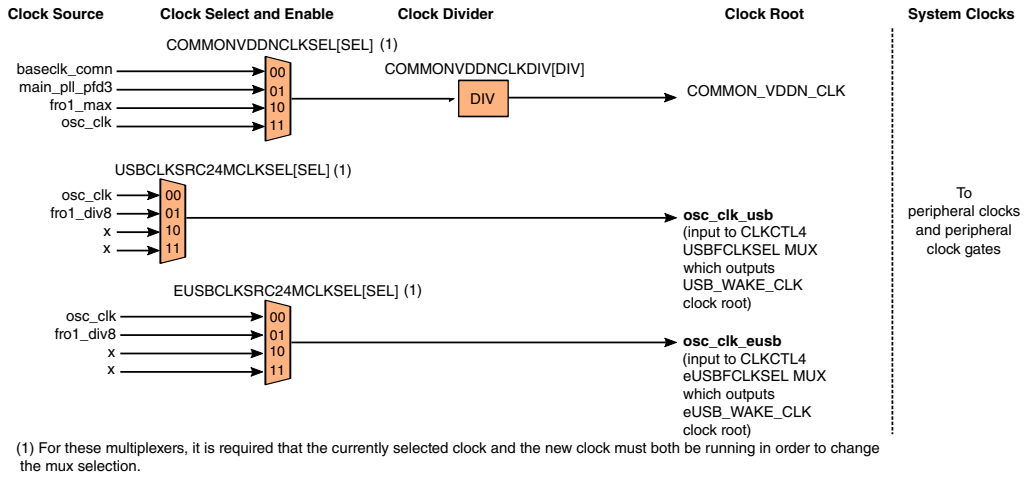


Figure 96. CLKCTL2 clock tree

28.4.1.2.3.2 CLKCTL2 clock roots

Table 240. CLKCTL2 clock roots

Clock source select register	Clock root	Description
COMMONVDDNCLKSEL	COMMON_VDDN_CLK	Derives the clock of VDDN_COM domain
USBCLKSRC24MCLKSEL	osc_clk_usb	Derives the clock for USB0

Table continues on the next page...

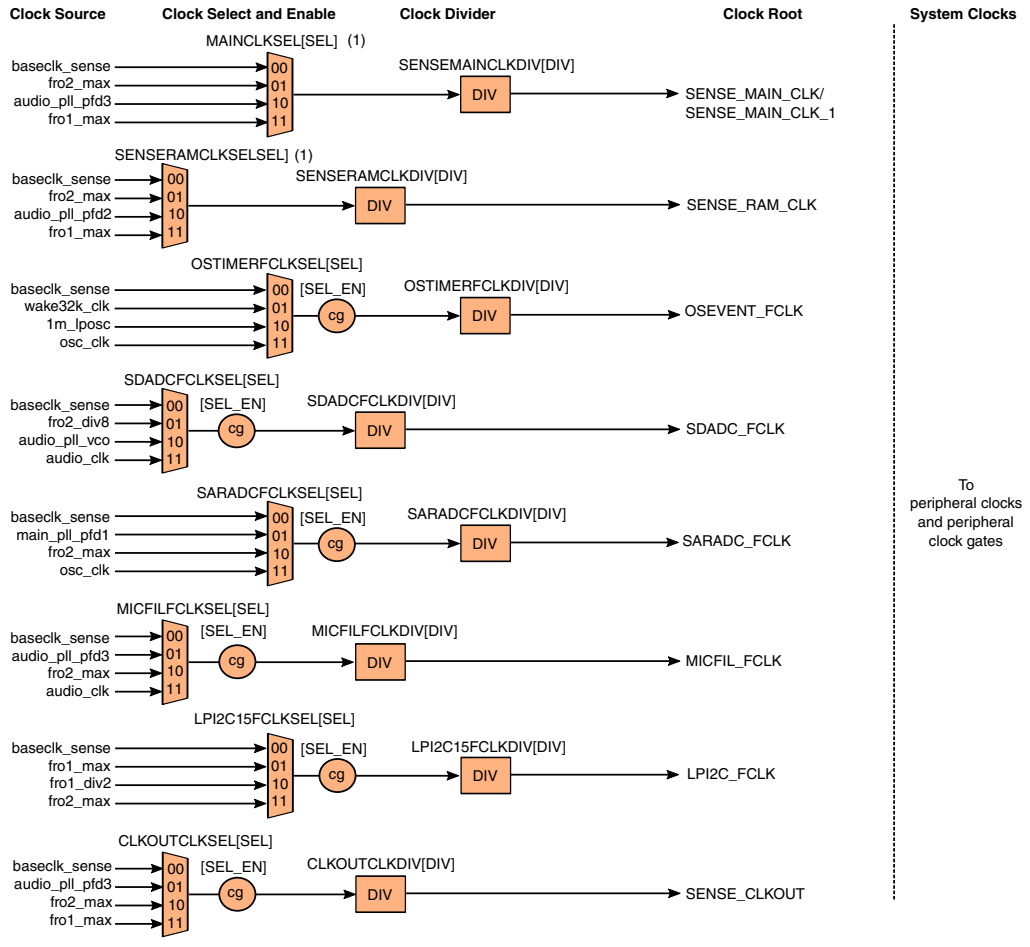
Table 240. CLKCTL2 clock roots (continued)

Clock source select register	Clock root	Description
		(input to USBFCLKSEL MUX, which outputs USB_WAKE_CLK clock root)
EUSBCLKSRC24MCLKSEL	osc_clk_eusb	Derives the USB1 (eUSB) input clock (input to eUSBFCLKSEL MUX, which outputs EUSB_WAKE_CLK clock root)

28.4.1.2.4 CLKCTL3 clock tree and clock roots

28.4.1.2.4.1 CLKCTL3 clock tree

The following diagram illustrates the clock tree for the CLKCTL3 instance. Detailed configuration information of these clock roots can be found in the CLKCTL3 clock roots section.



(1) For these multiplexers, it is required that the currently selected clock and the new clock must both be running in order to change the mux selection.

Figure 97. CLKCTL3 clock tree

28.4.1.2.4.2 CLKCTL3 clock roots

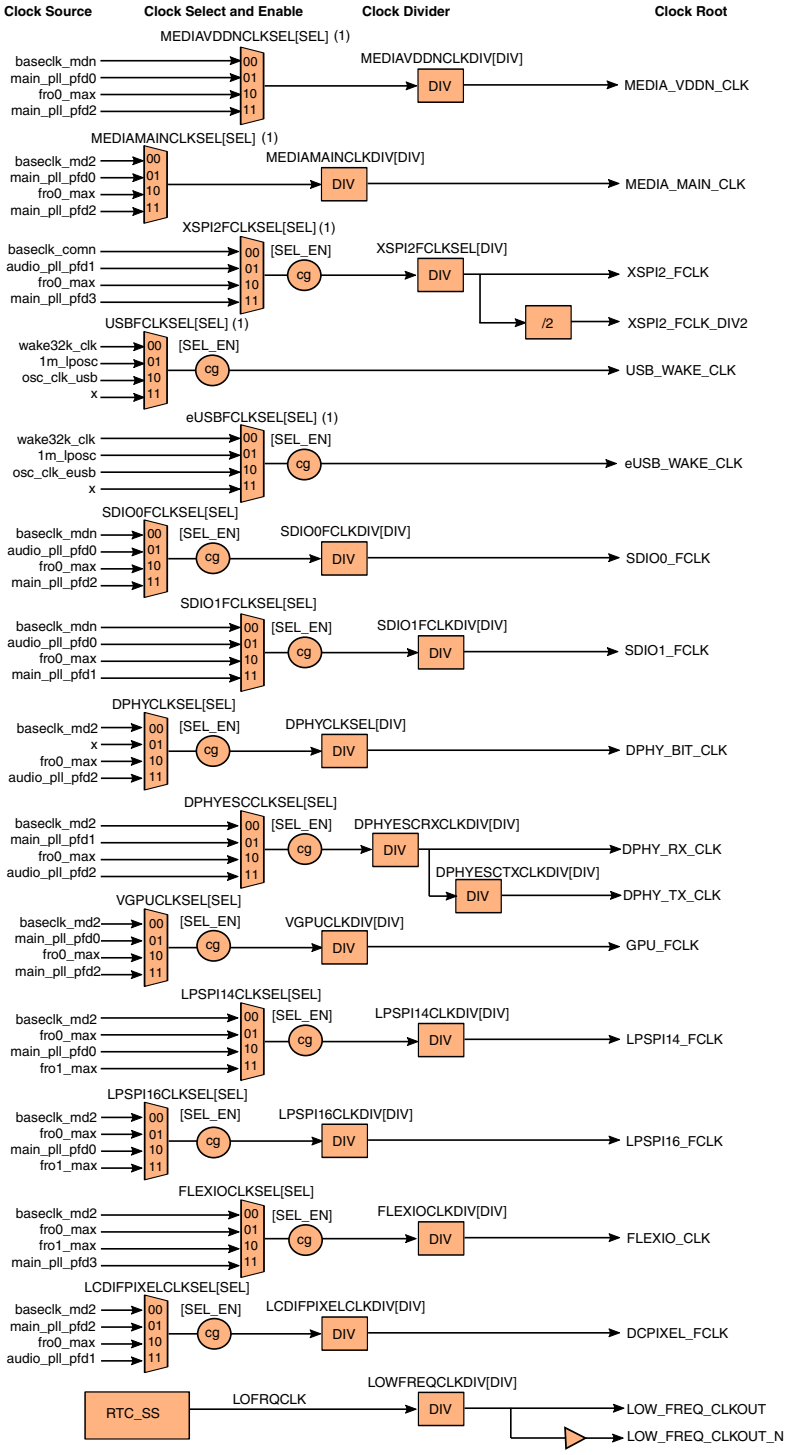
Table 241. CLKCTL3 clock roots

Clock source select register	Clock root	Description
MAINCLKSEL	SENSE_MAIN_CLK/ SENSE_MAIN_CLK_1	Derives the CPU clock of VDD1_SENSE domain **NOTE:** baseclk_sense is in CLKCTL1 instead of CLKCTL3
SENSERAMCLKSEL	SENSE_RAM_CLK	Derives the clock of RAM in VDD1_SENSE domain for async access
SDADCFCLKSEL	SDADC_FCLK	Derives the clock of SDADC
SARADCFCLKSEL	SARADC_FCLK	Derives the clock of ADC0
OSTIMERFCLKSEL	OSEVENT_FCLK	Derives the clock for the OSTIMER
MICFILFCLKSEL	MICFIL_FCLK	Derives the clock for the MICFIL
LPI2C15FCLKSEL	LPI2C_FCLK	Derives the LPI2C15 clock
CLKOUTCLKSEL	SENSE_CLKOUT	Derives the clock out of VDD1_SENSE

28.4.1.2.5 CLKCTL4 clock tree and clock roots

28.4.1.2.5.1 CLKCTL4 clock tree

The following diagram illustrates the clock tree for the CLKCTL4 instance. Detailed configuration information of these clock roots can be found in the CLKCTL4 clock roots section.



(1) For these multiplexers, it is required that the currently selected clock and the new clock must both be running in order to change the mux selection.

Figure 98. CLKCTL4 clock tree

28.4.1.2.5.2 CLKCTL4 clock roots

Table 242. CLKCTL4 clock roots

Clock source select register	Clock root	Description
MEDIAMAINCLKSEL	MEDIA_MAIN_CLK	Derives the clock of VDD2_MEDIA domain
MEDIAVDDNCLKSEL	MEDIA_VDDN_CLK	Derives the clock of VDDN_MEDIA domain
XSPI2FCLKSEL	XSPI2_FCLK	Derives the clock of XSPI2
SDIO0FCLKSEL (for uSDHC clock select)	SDIO0_FCLK	Derives the clock of SDIO0
SDIO1FCLKSEL	SDIO1_FCLK	Derives the clock of SDIO1
VGPUCLKSEL	GPU_FCLK	Derives the clock of the VGPU
LCDIFPIXELCLKSEL	DCPIXEL_FCLK	Derives the clock of the LCDIF
DPHYCLKSEL	DPHY_BIT_CLK	Derives the MIPI_DSI_HOST PHY bit clock
DPHYESCCLKSEL	DPHY_RXTX_CLK	Derives the MIPI_DSI_HOST PHY RX/TX clock
FLEXIOCLKSEL	FLEXIO_CLK	Derives the clock for the FLEXIO
LPSPI14CLKSEL	LPSPI14_FCLK	Derives the LPSPI14 clock
LPSPI16CLKSEL	LPSPI16_FCLK	Derives the LPSPI16 clock
eUSBFCLKSEL	eUSB_WAKE_CLK	Derives the USB1 input clock
USBFCLKSEL	USB_WAKE_CLK	Derives the USB0 clock

28.4.1.2.6 Clock roots - no clock source select

Table 243. Clock roots - no clock source select

Clock source select register	Clock root	0.7V nom Max Freq (MHz)	1.0V nom Max Freq (MHz)	Description/Source Select (SEL)
-	usbphy0 clk_xtal	-	-	USBPHY0 reference clock (osc_24m)

28.4.1.3 Clock gates

Before a clock root goes to on-chip peripherals, the clock root branches into a distributed network of system clocks, which then interface clock gating logic. Without gating the clock root, this distributed network of clocks can be gated individually at the peripheral interface. CLKCTL provides the enables for these clock gates, in the Peripheral Software Clock Control (PSCCTLn) registers of each CLKCTL instance. These clock gates are also mapped to the peripheral clocks in the System Clocks table. Please see the PSCCTLn registers and System Clocks section for more information.

28.4.2 Phase Locked Loops (PLLs)

The PLLs, including main PLL and audio PLL, support a range of input frequencies and a range of VCO frequencies. In addition to a typical integer frequency multiplier, the PLLs also provide for a fractional frequency multiplier. The PLL VCO frequency is given by:

- PLL output frequency = Fref * (MULT + NUM/DENOM)
- Fref = PLL input reference frequency
- MULT = 8-bit PLL multiplier (See Main PLL0 Control 0 and Audio PLL0 Control 0)
- NUM = 30-bit numerator for fractional multiplier (See Main PLL0 Numerator and Audio PLL0 Numerator)
- DENOM = 30-bit denominator for fractional multiplier (See Main PLL0 Denominator and Audio PLL0 Denominator)

28.4.2.1 PLL limitations

Below lists PLL limitations:

- Input frequencies: 16, 19.2, 24, 26, 30 or 32 MHz
- PLL multiplier values: 16 to 22 in integer.
- PLL PFD: limited to 600 MHz (maximum) at 0.9 V or above voltage.

28.4.2.2 Main PLL

The main PLL can be configured to use a number of clock inputs and produce output clocks up to the maximum chip frequency. You can use the main PLL to run most of on-chip functions. The main PLL creates a stable output clock at a higher frequency than the input clock.

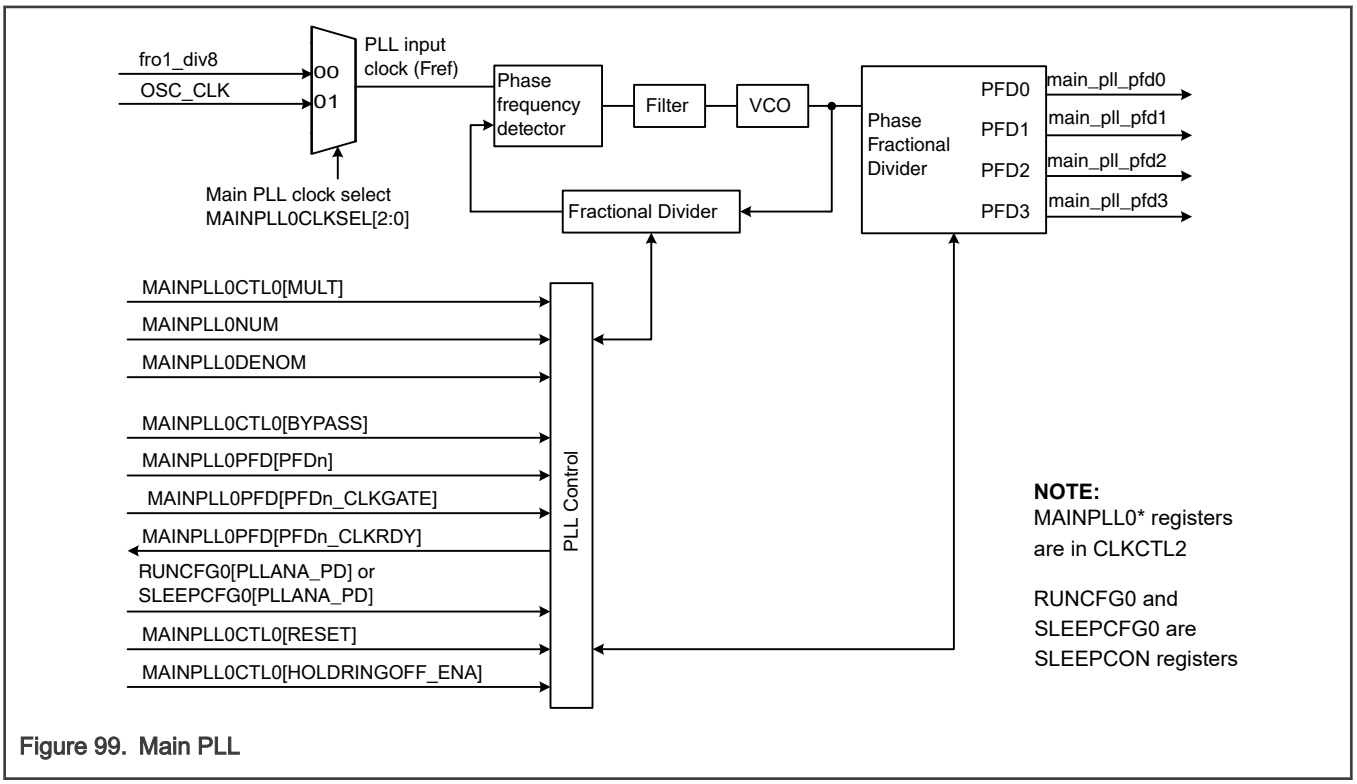


Figure 99. Main PLL

28.4.2.2.1 Setting up the main PLL

When the main PLL is running, you can change the NUM value by a step of 1 at a time and a delay at each change for the main PLL to settle to the new setting. DENUM value cannot be changed while a PLL is running.

The VCO frequency of the main PLL will be outputted on a particular Phase Fractional Divider (PFD) output when the PFD is set to 18. This does not alter the main PLL output (the value 18 causes the PFD to be divided by 1).

You can use PFD settings to alter the VCO frequency of the main PLL before it is outputted. Each PFD output may have a different setting. The PFD output frequency is given by:

- PFD Output = 18/N x FVCO where N = 12 to 35

Table 244. PFD fractions

MAINPLL0PFD[PFDn]	Fraction	Decimal
12	18/12	1.5
13	18/13	1.3846
14	18/14	1.2857
15	18/15	1.2
16	18/16	1.125
17	18/17	1.0588
18	18/18	1
19	18/19	0.9474
20	18/20	0.9
21	18/21	0.8571
22	18/22	0.8182
23	18/23	0.7826
24	18/24	0.75
25	18/25	0.72
26	18/26	0.6923
27	18/27	0.6667
28	18/28	0.6429
29	18/29	0.6207
30	18/30	0.6
31	18/31	0.5806
32	18/32	0.5625
33	18/33	0.5455
34	18/34	0.5294
35	18/35	0.5143

28.4.2.3 Audio PLL

The audio PLL allows using a frequency unrelated to the main PLL for audio processing. It can use a fractional multiplier that is used to generate accurate audio frequencies.

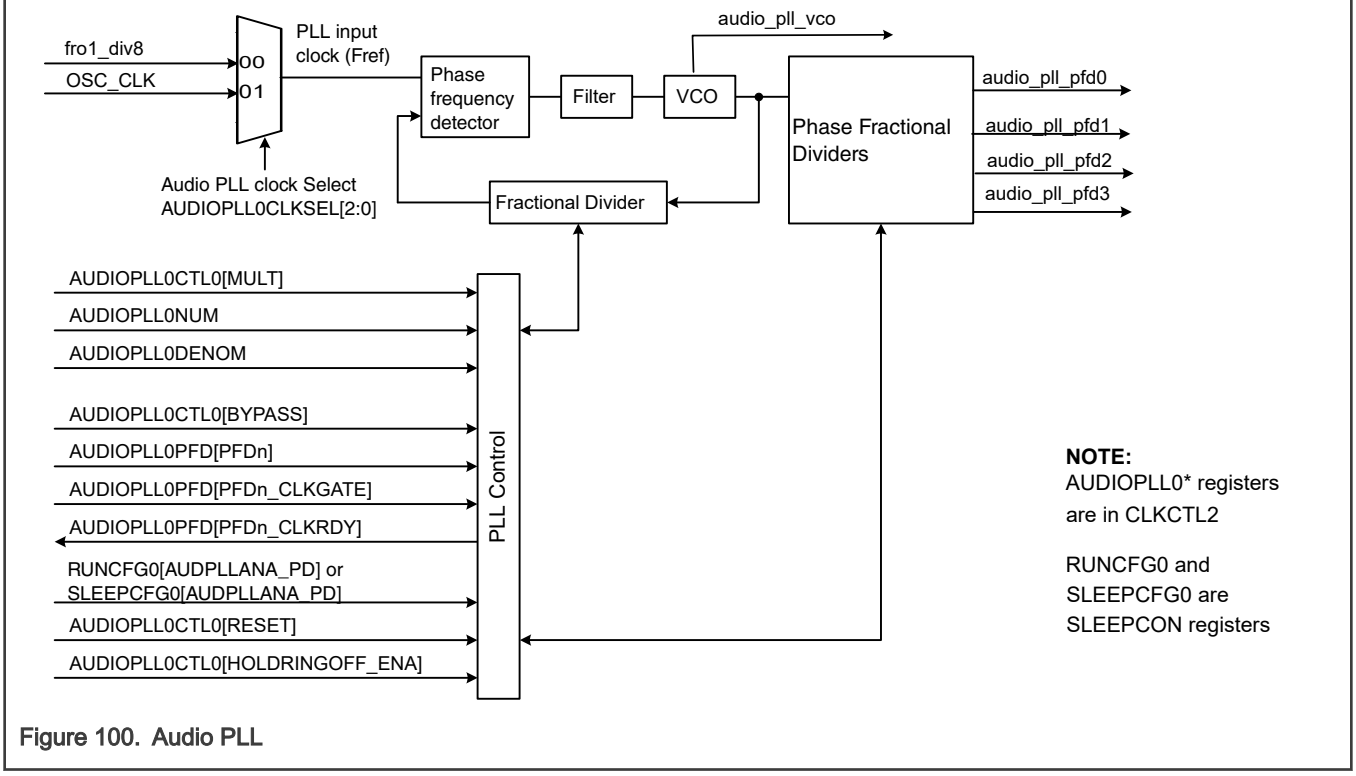


Figure 100. Audio PLL

28.4.2.3.1 Generating audio frequencies for the audio PLL

You can use a fractional multiplier to generate accurate audio frequencies using below settings as an example:

- Input frequency = 24 MHz
- Desired frequency = 44.1 kHz
- Over Sampled frequency = 256 x 44.1 kHz = 11.289 MHz
- Multiplier = 22.578 [AUDIOPLL0CTL0[MULT] = 22, AUDIOPLL0NUM = 578, AUDIOPLL0DENOM = 1000]
- VCO Output = 24 MHz x 22.578 = 541.872 MHz
- The audio PLL output can be divided by 48 to get the 11.289 MHz oversample frequency.

In this case, the related AUDIOPLL0PFD[PFDn] is set to 18, which does not alter the audio PLL output. Other audio frequencies can be similarly generated by manipulating multiplier (AUDIOPLL0CTL0[MULT]), numerator (AUDIOPLL0NUM) and denominator (AUDIOPLL0DENOM). Alternatively, the PFD might be used to create the desired frequency from a different VCO frequency.

28.4.3 Interrupts

This module has no interrupts.

28.5 External signals

Table 245 describes CLKCTL pins.

Table 245. CLKCTL pin description

Name	I/O	Description
CLKIN	I	CLKIN pin. It can optionally be used to supply the internal clock osc_clk. When used, it must be connected to a device pin by selecting it in the IOPCTL block. See IOPCTL chapter for more information.
CLKOUT	O	Many on-chip clocks can potentially be selected to be output on the CLKOUT function. This can be used for debugging purposes or to drive some external logic (see data sheet for limitations on pin output frequency). If used, the CLKOUT pin function must be connected to a pin by selecting it in the IOPCTL block. See IOPCTL chapter for more information.
LOW_FREQ_CLKOUT	O	Output pin for low-frequency clock-out
LOW_FREQ_CLKOUT_N	O	Output pin - complement of LOW_FREQ_CLKOUT
MCLK	I	MCLK input. The MCLK IN pin function is used as base source of the internal clock audio_clk when it is connected to a pin by selecting it in the IOPCTL block. See IOPCTL chapter for more information.
RTCXIN	I	Input pin for the RTC 32 kHz crystal oscillator
RTCXOUT	O	Output pin for the RTC 32 kHz crystal oscillator
XTALIN	I	Input pin for the main crystal oscillator. XTALIN can be input to external square wave when oscillator is bypassed.
XTALOUT	O	Output pin for the main crystal oscillator

28.6 Initialization

This module does not require initialization.

28.7 CLKCTL register descriptions

This section describes the memory map and all registers of CLKCTL. The CLKCTL module is organized into 5 separate clock controllers: CLKCTL0, CLKCTL1, CLKCTL2, CLKCTL3, and CLKCTL4.

28.7.1 CLKCTL0 register descriptions

This section includes the memory map and register descriptions for Clock Controller 0.

28.7.1.1 CLKCTL0 memory map

CLKCTL0 base address: 4000_1000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDD2_COMP Peripheral Clock Control 0 (PSCCTL0)	32	RW	0000_F006h
14h	VDD2_COMP Peripheral Clock Control 1 (PSCCTL1)	32	RW	0000_0200h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
18h	VDD2_COMP Peripheral Clock Control 2 (PSCCTL2)	32	RW	0000_0000h
1Ch	VDD2_COMP Peripheral Clock Control 3 (PSCCTL3)	32	RW	000F_8000h
20h	VDD2_COMP Peripheral Clock Control 4 (PSCCTL4)	32	RW	0000_0001h
24h	VDD2_COMP Peripheral Clock Control 5 (PSCCTL5)	32	RW	0000_0008h
40h	VDD2_COMP Peripheral Clock Control 0 Set (PSCCTL0_SET)	32	RW	0000_0000h
44h	VDD2_COMP Peripheral Clock Control 1 Set (PSCCTL1_SET)	32	RW	0000_0000h
48h	VDD2_COMP Peripheral Clock Control 2 Set (PSCCTL2_SET)	32	RW	0000_0000h
4Ch	VDD2_COMP Peripheral Clock Control 3 Set (PSCCTL3_SET)	32	RW	0000_0000h
50h	VDD2_COMP Peripheral Clock Control 4 Set (PSCCTL4_SET)	32	RW	0000_0000h
54h	VDD2_COMP Peripheral Clock Control 5 Set (PSCCTL5_SET)	32	RW	0000_0000h
70h	VDD2_COMP Peripheral Clock Control 0 Clear (PSCCTL0_CLR)	32	RW	0000_0000h
74h	VDD2_COMP Peripheral Clock Control 1 Clear (PSCCTL1_CLR)	32	RW	0000_0000h
78h	VDD2_COMP Peripheral Clock Control 2 Clear (PSCCTL2_CLR)	32	RW	0000_0000h
7Ch	VDD2_COMP Peripheral Clock Control 3 Clear (PSCCTL3_CLR)	32	RW	0000_0000h
80h	VDD2_COMP Peripheral Clock Control 4 Clear (PSCCTL4_CLR)	32	RW	0000_0000h
84h	VDD2_COMP Peripheral Clock Control 5 Clear (PSCCTL5_CLR)	32	RW	0000_0000h
118h	FRO_TUNER0 and FRO_TUNER1 Clock Status (FRO01CLKSTATUS)	32	R	0000_0000h
128h	FRO0MAX Clock Domain Enable (FRO0MAXDOMAINEN)	32	RW	0000_007Fh
400h	VDD2_COMP Main Clock Divider (MAINCLKDIV)	32	RW	0000_0001h
420h	VDD2_COMP Base Clock Select Source (CMPTBASECLKSEL)	32	RW	0000_0000h
424h	VDD2_DSP Base Clock Select Source (DSPBASECLKSEL)	32	RW	0000_0000h
428h	VDD2_COM Base Clock Select Source (VDD2COMBASECLKSEL)	32	RW	0000_0000h
434h	VDD2_COMP Main Clock Source Select (MAINCLKSEL)	32	RW	0000_0000h
440h	VDD2_DSP Clock Divider (DSPCPUCLKDIV)	32	RW	0000_0001h
444h	VDD2_DSP Clock Select Source (DSPCPUCLKSEL)	32	RW	0000_0004h
450h	RAM Clock Select Source (RAMCLKSEL)	32	RW	0000_0000h
45Ch	RAM Clock Divider (RAMCLKDIV)	32	RW	0000_0001h
560h	TPIU (TRACE_RT700) Functional Clock Select Source (TPIUFCLKSEL)	32	RW	0000_0000h
564h	TPIU (TRACE_RT700) Functional Clock Divider (TPIUCLKDIV)	32	RW	0000_0001h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
600h	XSPI0 Functional Clock Source Select (XSPI0FCLKSEL)	32	RW	0000_0000h
604h	XSPI0 Functional Clock Divider (XSPI0FCLKDIV)	32	RW	4000_0001h
620h	XSPI1 Functional Clock Select Source (XSPI1FCLKSEL)	32	RW	0000_0000h
624h	XSPI1 Functional Clock Divider (XSPI1FCLKDIV)	32	RW	4000_0001h
640h	SCT Functional Clock Source Select (SCTFCLKSEL)	32	RW	0000_0000h
644h	SCT Functional Clock Divider (SCTFCLKDIV)	32	RW	4000_0001h
700h	UTICK0 Functional Clock Select Source (UTICK0FCLKSEL)	32	RW	0000_0000h
704h	UTICK0 Functional Clock Divider (UTICK0FCLKDIV)	32	RW	0000_0001h
720h	WWDT0 Functional Clock Source Select (WWDT0FCLKSEL)	32	RW	0000_0000h
740h	WWDT1 Functional Clock Source Select (WWDT1FCLKSEL)	32	RW	0000_0000h
760h	SYSTICK Functional Clock Source Select (SYSTICKFCLKSEL)	32	RW	0000_0000h
764h	SYSTICK Functional Clock Divider (SYSTICKFCLKDIV)	32	RW	4000_0001h
800h	LP_FLEXCOMM 0 to 13 Clock Source 0 Select (FCCLK0SEL)	32	RW	0000_0000h
804h	LP_FLEXCOMM 0 to 13 Clock Source 0 Divider (FCCLK0DIV)	32	RW	4000_0001h
808h	LP_FLEXCOMM0 Clock Source Select (FC0FCLKSEL)	32	RW	0000_0000h
820h	LP_FLEXCOMM 0 to 13 Clock Source 1 Select (FCCLK1SEL)	32	RW	0000_0000h
824h	LP_FLEXCOMM 0 to 13 Clock Source 1 Divider (FCCLK1DIV)	32	RW	4000_0001h
828h	LP_FLEXCOMM1 Clock Source Select (FC1FCLKSEL)	32	RW	0000_0000h
840h	LP_FLEXCOMM 0 to 13 Clock Source 2 Select (FCCLK2SEL)	32	RW	0000_0000h
844h	LP_FLEXCOMM 0 to 13 Clock Source 2 Divider (FCCLK2DIV)	32	RW	4000_0001h
848h	LP_FLEXCOMM2 Clock Source Select (FC2FCLKSEL)	32	RW	0000_0000h
860h	LP_FLEXCOMM 0 to 13 Clock Source 3 Select (FCCLK3SEL)	32	RW	0000_0000h
864h	LP_FLEXCOMM 0 to 13 Clock Source 3 Divider (FCCLK3DIV)	32	RW	4000_0001h
868h	LP_FLEXCOMM3 Clock Source Select (FC3FCLKSEL)	32	RW	0000_0000h
888h	LP_FLEXCOMM4 Clock Source Select (FC4FCLKSEL)	32	RW	0000_0000h
8A8h	LP_FLEXCOMM5 Clock Source Select (FC5FCLKSEL)	32	RW	0000_0000h
8C8h	LP_FLEXCOMM6 Clock Source Select (FC6FCLKSEL)	32	RW	0000_0000h
8E8h	LP_FLEXCOMM7 Clock Source Select (FC7FCLKSEL)	32	RW	0000_0000h
908h	LP_FLEXCOMM8 Clock Source Select (FC8FCLKSEL)	32	RW	0000_0000h
928h	LP_FLEXCOMM9 Clock Source Select (FC9FCLKSEL)	32	RW	0000_0000h
948h	LP_FLEXCOMM10 Clock Source Select (FC10FCLKSEL)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
968h	LP_FLEXCOMM11 Clock Source Select (FC11FCLKSEL)	32	RW	0000_0000h
988h	LP_FLEXCOMM12 Clock Source Select (FC12FCLKSEL)	32	RW	0000_0000h
9A8h	LP_FLEXCOMM13 Clock Source Select (FC13FCLKSEL)	32	RW	0000_0000h
9C8h	SAI0, SAI1 and SAI2 Clock Source Select (SAI012FCLKSEL)	32	RW	0000_0000h
9CCh	SAI0, SAI1 and SAI2 Functional Clock Divider (SAI012CLKDIV)	32	RW	4000_0001h
A00h	CTIMER0 Functional Clock Divider (CTIMER0CLKDIV)	32	RW	4000_0001h
A04h	CTIMER1 Functional Clock Divider (CTIMER1CLKDIV)	32	RW	4000_0001h
A08h	CTIMER2 Functional Clock Divider (CTIMER2CLKDIV)	32	RW	4000_0001h
A0Ch	CTIMER3 Functional Clock Divider (CTIMER3CLKDIV)	32	RW	4000_0001h
A10h	CTIMER4 Functional Clock Divider (CTIMER4CLKDIV)	32	RW	4000_0001h
AA0h	CTIMER0 Functional Clock Source Select (CTIMER0FCLKSEL)	32	RW	0000_0000h
AA4h	CTIMER1 Functional Clock Source Select (CTIMER1FCLKSEL)	32	RW	0000_0000h
AA8h	CTIMER2 Functional Clock Source Select (CTIMER2FCLKSEL)	32	RW	0000_0000h
AACh	CTIMER3 Functional Clock Source Select (CTIMER3FCLKSEL)	32	RW	0000_0000h
AB0h	CTIMER4 Functional Clock Source Select (CTIMER4FCLKSEL)	32	RW	0000_0000h
B00h	I3C0 and I3C1 Functional Clock Source Select (I3C01FCLKSEL)	32	RW	0000_0000h
B04h	I3C0 and I3C1 P-CLK Source Select (I3C01PCLKSEL)	32	RW	0000_0000h
B08h	I3C0 and I3C1 P-CLK Divider (I3C01PCLKDIV)	32	RW	4000_0001h
B10h	I3C0 and I3C1 Functional Clock Divider (I3C01FCLKDIV)	32	RW	4000_0001h
B20h	CLKOUT_VDD2 Clock Select Source (CLKOUTCLKSEL)	32	RW	0000_0000h
B24h	CLKOUT_VDD2 Clock Divider (CLKOUTCLKDIV)	32	RW	4000_0001h
B30h	VDD2_COMP Audio Clock Source Select (AUDIOVDD2CLKSEL)	32	RW	0000_0000h

28.7.1.2 VDD2_COMP Peripheral Clock Control 0 (PSCCTL0)

Offset

Register	Offset
PSCCTL0	10h

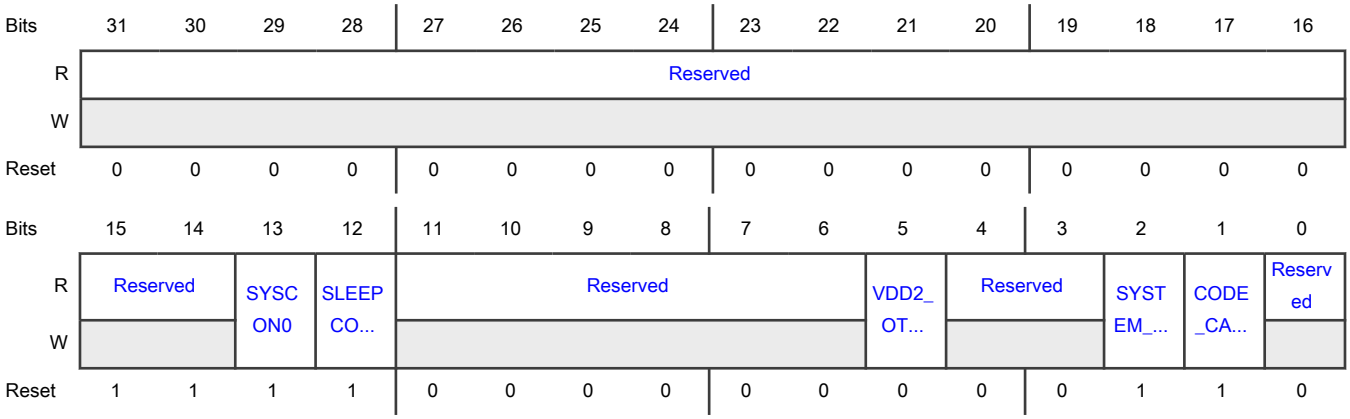
Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-14 —	Reserved
13 SYSCON0	SYSCON0 Clock Enables SYSCON0 clock. 0b - Disable 1b - Enable
12 SLEEPCON0	SLEEPCON0 Clock Enables SLEEPCON0 clock. 0b - Disable 1b - Enable
11-6 —	Reserved
5 VDD2_OTP0	VDD2_OTP_CTRL0 Clock Enables VDD2_OTP_CTRL0 clock. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
4-3 —	Reserved
2 SYSTEM_CACHE	System Cache Clock (XCACHE 0) Enables system cache clock. 0b - Disable 1b - Enable
1 CODE_CACHE	Code Cache Clock (XCACHE 1) Enables code cache clock. 0b - Disable 1b - Enable
0 —	Reserved

28.7.1.3 VDD2_COMP Peripheral Clock Control 1 (PSCCTL1)

Offset

Register	Offset
PSCCTL1	14h

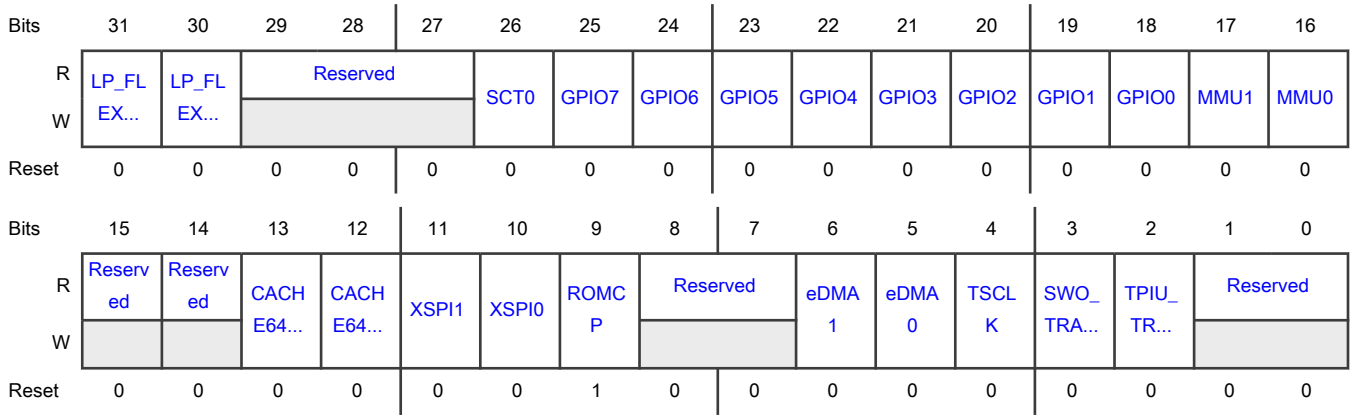
Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Clock Enables LP_FLEXCOMM1 clock. 0b - Disable 1b - Enable
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Clock Enables LP_FLEXCOMM0 clock. 0b - Disable 1b - Enable
29-27 —	Reserved
26 SCT0	SCT0 Clock Enables SCT0 clock. 0b - Disable 1b - Enable
25 GPIO7	GPIO7 Clock Enables GPIO7 clock. 0b - Disable 1b - Enable
24 GPIO6	GPIO6 Clock Enables GPIO6 clock. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
23 GPIO5	GPIO5 Clock Enables GPIO5 clock. 0b - Disable 1b - Enable
22 GPIO4	GPIO4 Clock Enables GPIO4 clock. 0b - Disable 1b - Enable
21 GPIO3	GPIO3 Clock Enables GPIO3 clock. 0b - Disable 1b - Enable
20 GPIO2	GPIO2 Clock Enables GPIO2 clock. 0b - Disable 1b - Enable
19 GPIO1	GPIO1 Clock Enables GPIO1 clock. 0b - Disable 1b - Enable
18 GPIO0	GPIO0 Clock Enables GPIO0 clock. 0b - Disable 1b - Enable
17 MMU1	MMU1 Clock Enables MMU1 clock. 0b - Disable 1b - Enable
16 MMU0	MMU0 Clock Enables MMU0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
15 —	Reserved
14 —	Reserved
13 CACHE64_1	CACHE64_1 Clock Enables CACHE64_1 clock. 0b - Disable 1b - Enable
12 CACHE64_0	CACHE64_0 Clock Enables CACHE64_0 clock. 0b - Disable 1b - Enable
11 XSPI1	XSPI1 Clock Enables XSPI1 clock. 0b - Disable 1b - Enable
10 XSPI0	XSPI0 Clock Enables XSPI0 clock. 0b - Disable 1b - Enable
9 ROMCP	ROMCP Clock Enables ROMCP clock. 0b - Disable 1b - Enable
8-7 —	Reserved
6 eDMA1	eDMA1 Clock Enables eDMA1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
5 eDMA0	eDMA0 Clock Enables eDMA0 clock. 0b - Disable 1b - Enable
4 TSCLK	TS (TRACE_RT700) Clock Enables TS clock. 0b - Disable 1b - Enable
3 SWO_TRACECLKIN	SWO_TRACECLKIN (TRACE_RT700) Clock Enables SWO_TRACECLKIN (TRACE_RT700) clock. 0b - Disable 1b - Enable
2 TPIU_TRACE_CLKIN	TPIU_TRACE_CLKIN (TRACE_RT700) Clock Enables TPIU_TRACE_CLKIN clock. 0b - Disable 1b - Enable
1-0 —	Reserved

28.7.1.4 VDD2_COMP Peripheral Clock Control 2 (PSCCTL2)

Offset

Register	Offset
PSCCTL2	18h

Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MU4	SEMA42_4	Reserved		UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0	WWDT1	WWDT0	CRC0	I3C1	I3C0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAI2	SAI1	SAI0	Reserved	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MU4	MU4 Clock Enables MU4 clock. 0b - Disable 1b - Enable
30 SEMA42_4	SEMA42_4 Clock Enables SEMA42_4 clock. 0b - Disable 1b - Enable
29-28 —	Reserved
27 UTICK0	UTICK0 Clock Enables UTICK0 clock. 0b - Disable 1b - Enable
26 MRT0	MRT0 Clock Enables MRT0 clock. 0b - Disable 1b - Enable
25 CTIMER4	CTIMER4 Clock Enables CTIMER4 clock. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
24 CTIMER3	CTIMER3 Clock Enables CTIMER3 clock. 0b - Disable 1b - Enable
23 CTIMER2	CTIMER2 Clock Enables CTIMER2 clock. 0b - Disable 1b - Enable
22 CTIMER1	CTIMER1 Clock Enables CTIMER1 clock. 0b - Disable 1b - Enable
21 CTIMER0	CTIMER0 Clock Enables CTIMER0 clock. 0b - Disable 1b - Enable
20 WWDT1	WWDT1 Clock Enables WWDT1 clock. 0b - Disable 1b - Enable
19 WWDT0	WWDT0 Clock Enables WWDT0 clock. 0b - Disable 1b - Enable
18 CRC0	CRC0 Clock Enables CRC0 clock. 0b - Disable 1b - Enable
17 I3C1	I3C1 Clock Enables I3C1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
16 I3C0	I3C0 Clock Enables I3C0 clock. 0b - Disable 1b - Enable
15 SAI2	SAI2 Clock Enables SAI2 clock. 0b - Disable 1b - Enable
14 SAI1	SAI1 Clock Enables SAI1 clock. 0b - Disable 1b - Enable
13 SAI0	SAI0 Clock Enables SAI0 clock. 0b - Disable 1b - Enable
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Clock Enables LP_FLEXCOMM13 clock. 0b - Disable 1b - Enable
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Clock Enables LP_FLEXCOMM12 clock. 0b - Disable 1b - Enable
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Clock Enables LP_FLEXCOMM11 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
8 LP_FLEXCOMM10	LP_FLEXCOMM10 Clock Enables LP_FLEXCOMM10 clock. 0b - Disable 1b - Enable
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Clock Enables LP_FLEXCOMM9 clock. 0b - Disable 1b - Enable
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Clock Enables LP_FLEXCOMM8 clock. 0b - Disable 1b - Enable
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Clock Enables LP_FLEXCOMM7 clock. 0b - Disable 1b - Enable
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Clock Enables LP_FLEXCOMM6 clock. 0b - Disable 1b - Enable
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Clock Enables LP_FLEXCOMM5 clock. 0b - Disable 1b - Enable
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Clock Enables LP_FLEXCOMM4 clock. 0b - Disable 1b - Enable
1	LP_FLEXCOMM3 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM3	Enables LP_FLEXCOMM3 clock. 0b - Disable 1b - Enable
0 LP_FLEXCOMM2	LP_FLEXCOMM2 Clock Enables LP_FLEXCOMM2 clock. 0b - Disable 1b - Enable

28.7.1.5 VDD2_COMP Peripheral Clock Control 3 (PSCCTL3)

Offset

Register	Offset
PSCCTL3	1Ch

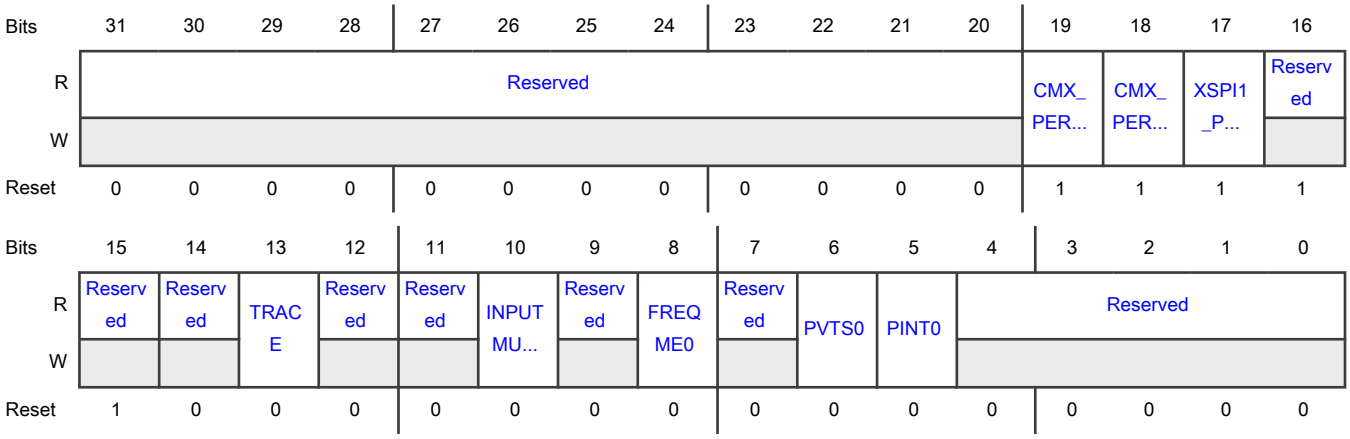
Function

Allows software to control specific peripheral's clock. Writing a one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-20 —	Reserved
19 CMX_PERFMON1	CMX_PERFMON1 Clock Enables CMX_PERFMON1 clock. 0b - Disable 1b - Enable
18 CMX_PERFMON0	CMX_PERFMON0 Clock Enables CMX_PERFMON0 clock. 0b - Disable 1b - Enable
17 XSPI1_PRINCE_XEX	XSPI1 PRINCE_XEX Clock Enables XSPI1 PRINCE_XEX clock. 0b - Disable 1b - Enable
16-15 —	Reserved
14 —	Reserved
13 TRACE	TRACE_RT700 Clock Enables TRACE_RT700 clock. 0b - Disable 1b - Enable
12 —	Reserved
11 —	Reserved
10 INPUTMUX0	INPUTMUX0 Clock Enables INPUTMUX0 clock. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 —	Reserved
8 FREQME0	FREQME0 Clock Enables FREQME0 clock. 0b - Disable 1b - Enable
7 —	Reserved
6 PVTS0	PVTS0 Clock Enables PVTS0 clock. 0b - Disable 1b - Enable
5 PINT0	PINT0 Clock Enables PINT0 clock. 0b - Disable 1b - Enable
4-0 —	Reserved

28.7.1.6 VDD2_COMP Peripheral Clock Control 4 (PSCCTL4)

Offset

Register	Offset
PSCCTL4	20h

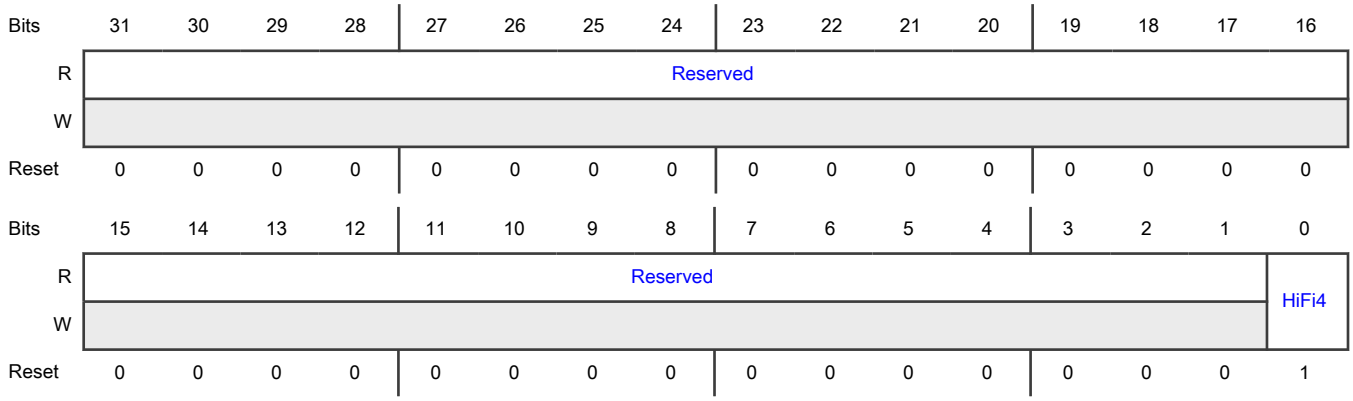
Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after Power On.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 HiFi4	HiFi4 Clock Enables HiFi4 clock. 0b - Disable 1b - Enable

28.7.1.7 VDD2_COMP Peripheral Clock Control 5 (PSCCTL5)

Offset

Register	Offset
PSCCTL5	24h

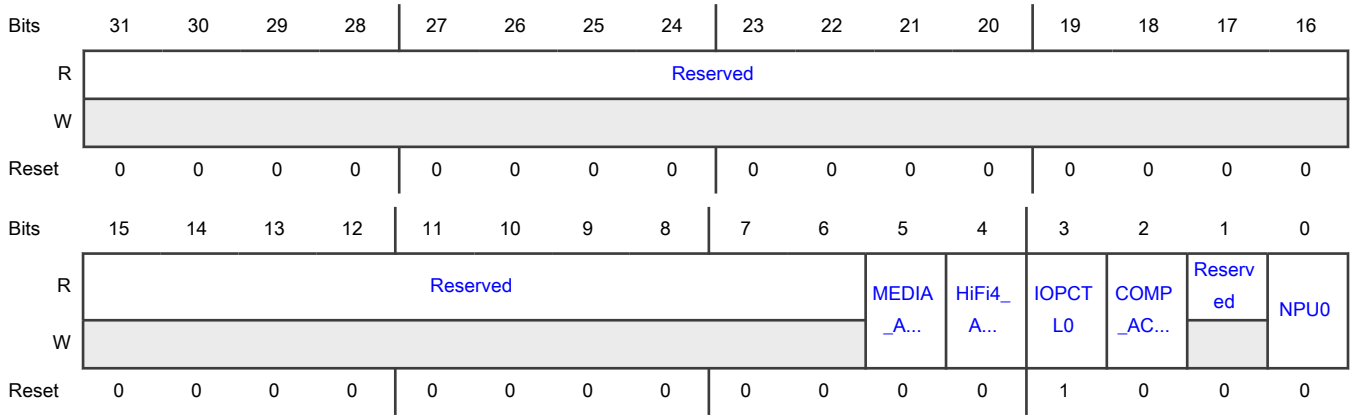
Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing a zero disables the clock. If VDD2_COMP power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 MEDIA_ACCESS_RAM_ARBITER0	VDD2_Media Access RAM Arbiter 0 Clock Enables VDD2_MEDIA access RAM arbiter 0 clock. 0b - Disable 1b - Enable
4 HiFi4_ACCESS_RAM_ARBITER1	HiFi4 Access RAM Arbiter1 Clock Enables HiFi4 access RAM arbiter1 clock. 0b - Disable 1b - Enable
3 IOPCTL0	IOPCTL0 Clock Enables IOPCTL0 clock. 0b - Disable 1b - Enable
2 COMP_ACCESS_RAM_ARBITER1	VDD2_COMP Access RAM Arbiter1 Clock Enables VDD2_COMP access RAM arbiter1 clock. 0b - Disable 1b - Enable
1 —	Reserved
0	NPU0 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
NPU0	Enables NPU0 clock. 0b - Disable 1b - Enable

28.7.1.8 VDD2_COMP Peripheral Clock Control 0 Set (PSCCTL0_SET)

Offset

Register	Offset
PSCCTL0_SET	40h

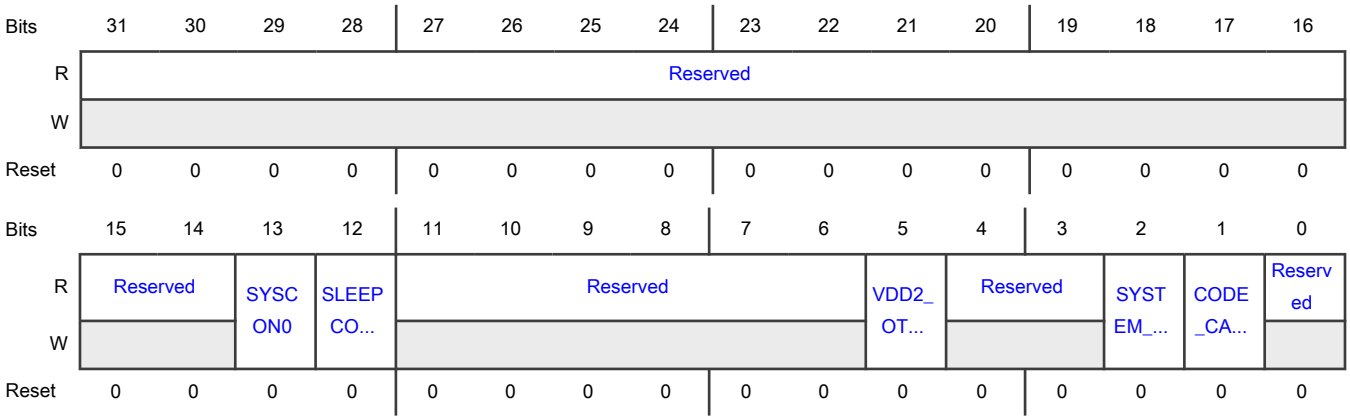
Function

Writing an one to a bit position in PSCCTL0_SET sets the corresponding position in PSCCTL0.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-14 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 SYSCON0	SYSCON0 Clock Enables SYSCON0 clock. 0b - No effect 1b - Enable
12 SLEEPCON0	SLEEPCON0 Clock Enables SLEEPCON0 clock. 0b - No effect 1b - Enable
11-6 —	Reserved
5 VDD2_OTP0	VDD2_OTP_CTRL0 Clock Enables VDD2_OTP_CTRL0 clock. 0b - No effect 1b - Enable
4-3 —	Reserved
2 SYSTEM_CACHE	XCACHE0 (System Cache) Clock Enables XCACHE0 clock. 0b - No effect 1b - Enable
1 CODE_CACHE	XCACHE1 (Code Cache) Clock Enables XCACHE1 clock. 0b - No effect 1b - Enable
0 —	Reserved

28.7.1.9 VDD2_COMP Peripheral Clock Control 1 Set (PSCCTL1_SET)

Offset

Register	Offset
PSCCTL1_SET	44h

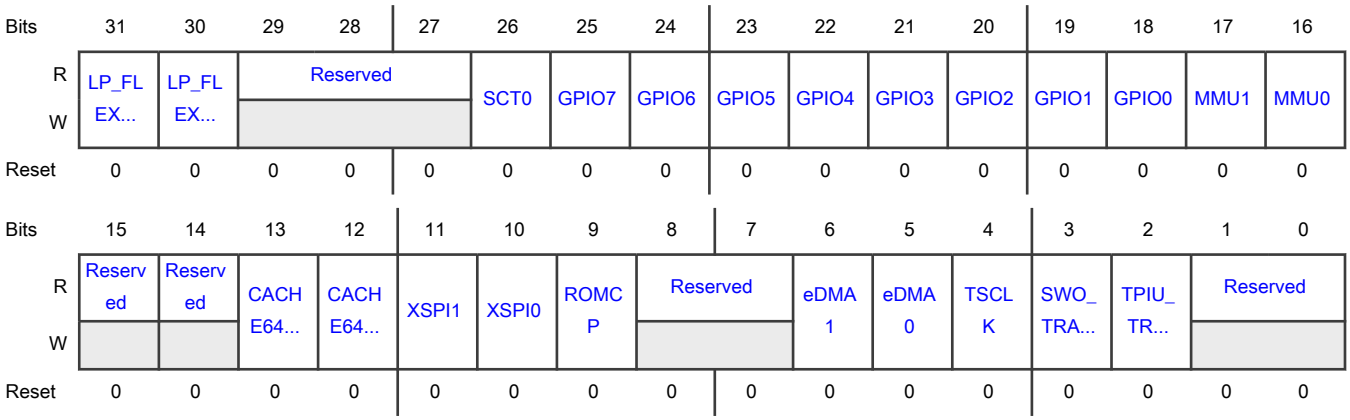
Function

Writing an one to a bit position in PSCCTL1_SET sets the corresponding position in PSCCTL1.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Clock Enables LP_FLEXCOMM1 clock. 0b - No effect 1b - Enable
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Clock Enables LP_FLEXCOMM0 clock. 0b - No effect 1b - Enable
29-27 —	Reserved
26 SCT0	SCT0 Clock Enables SCT0 clock. 0b - No effect 1b - Enable
25 GPIO7	GPIO7 Clock Enables GPIO7 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Enable
24 GPIO6	GPIO6 Clock Enables GPIO6 clock. 0b - No effect 1b - Enable
23 GPIO5	GPIO5 Clock Enables GPIO5 clock. 0b - No effect 1b - Enable
22 GPIO4	GPIO4 Clock Enables GPIO4 clock. 0b - No effect 1b - Enable
21 GPIO3	GPIO3 Clock Enables GPIO3 clock. 0b - No effect 1b - Enable
20 GPIO2	GPIO2 Clock Enables GPIO2 clock. 0b - No effect 1b - Enable
19 GPIO1	GPIO1 Clock Enables GPIO1 clock. 0b - No effect 1b - Enable
18 GPIO0	GPIO0 Clock Enables GPIO0 clock. 0b - No effect 1b - Enable
17	MMU1 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
MMU1	Enables MMU1 clock. 0b - No effect 1b - Enable
16 MMU0	MMU0 Clock Enables MMU0 clock. 0b - No effect 1b - Enable
15 —	Reserved
14 —	Reserved
13 CACHE64_1	CACHE64_1 Clock Enables CACHE64_1 clock. 0b - No effect 1b - Enable
12 CACHE64_0	CACHE64_0 Clock Enables CACHE64_0 clock. 0b - No effect 1b - Enable
11 XSPI1	XSPI1 Clock Enables XSPI1 clock. 0b - No effect 1b - Enable
10 XSPI0	XSPI0 Clock Enables XSPI0 clock. 0b - No effect 1b - Enable
9 ROMCP	ROMCP Clock Enables ROMCP clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
8-7 —	Reserved
6 eDMA1	eDMA1 Clock Enables eDMA1 clock. 0b - No effect 1b - Enable
5 eDMA0	eDMA0 Clock Enables eDMA0 clock. 0b - No effect 1b - Enable
4 TSCLK	TS (TRACE_RT700) Clock Enables TRACE_RT700 clock. 0b - No effect 1b - Enable
3 SWO_TRACECLKIN	SWO TRACECLKIN (TRACE_RT700) Clock Enables SWO TRACE_RT700 clock. 0b - No effect 1b - Enable
2 TPIU_TRACE_CLKIN	TPIU_TRACE_CLKIN (TRACE_RT700) Clock Enables TPIU TRACE_RT700 clock. 0b - No effect 1b - Enable
1-0 —	Reserved

28.7.1.10 VDD2_COMP Peripheral Clock Control 2 Set (PSCCTL2_SET)

Offset

Register	Offset
PSCCTL2_SET	48h

Function

Writing an one to a bit position in PSCCTL2_SET sets the corresponding position in PSCCTL2.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MU4	SEMA42_4	Reserved		UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0	WWDT1	WWDT0	CRC0	I3C1	I3C0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAI2	SAI1	SAI0	Reserved	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MU4	MU4 Clock Enables MU4 clock. 0b - No effect 1b - Enable
30 SEMA42_4	SEMA42_4 Clock Enables SEMA42_4 clock. 0b - No effect 1b - Enable
29-28 —	Reserved
27 UTICK0	UTICK0 Clock Enables UTICK0 clock. 0b - No effect 1b - Enable
26 MRT0	MRT0 Clock Enables MRT0 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 CTIMER4	CTIMER4 Clock Enables CTIMER4 clock. 0b - No effect 1b - Enable
24 CTIMER3	CTIMER3 Clock Enables CTIMER3 clock. 0b - No effect 1b - Enable
23 CTIMER2	CTIMER2 Clock Enables CTIMER2 clock. 0b - No effect 1b - Enable
22 CTIMER1	CTIMER1 Clock Enables CTIMER1 clock. 0b - No effect 1b - Enable
21 CTIMER0	CTIMER0 Clock Enables CTIMER0 clock. 0b - No effect 1b - Enable
20 WWDT1	WWDT1 Clock Enables WWDT1 clock. 0b - No effect 1b - Enable
19 WWDT0	WWDT0 Clock Enables WWDT0 clock. 0b - No effect 1b - Enable
18 CRC0	CRC0 Clock Enables CRC0 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
17 I3C1	I3C1 Clock Enables I3C1 clock. 0b - No effect 1b - Enable
16 I3C0	I3C0 Clock Enables I3C0 clock. 0b - No effect 1b - Enable
15 SAI2	SAI2 Clock Enables SAI2 clock. 0b - No effect 1b - Enable
14 SAI1	SAI1 Clock Enables SAI1 clock. 0b - No effect 1b - Enable
13 SAI0	SAI0 Clock Enables SAI0 clock. 0b - No effect 1b - Enable
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Clock Enables LP_FLEXCOMM13 clock. 0b - No effect 1b - Enable
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Clock Enables LP_FLEXCOMM12 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Clock Enables LP_FLEXCOMM11 clock. 0b - No effect 1b - Enable
8 LP_FLEXCOMM10	LP_FLEXCOMM10 Clock Enables LP_FLEXCOMM10 clock. 0b - No effect 1b - Enable
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Clock Enables LP_FLEXCOMM9 clock. 0b - No effect 1b - Enable
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Clock Enables LP_FLEXCOMM8 clock. 0b - No effect 1b - Enable
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Clock Enables LP_FLEXCOMM7 clock. 0b - No effect 1b - Enable
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Clock Enables LP_FLEXCOMM6 clock. 0b - No effect 1b - Enable
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Clock Enables LP_FLEXCOMM5 clock. 0b - No effect 1b - Enable
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Clock Enables LP_FLEXCOMM4 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
1 LP_FLEXCOMM3	LP_FLEXCOMM3 Clock Enables LP_FLEXCOMM3 clock. 0b - No effect 1b - Enable
0 LP_FLEXCOMM2	LP_FLEXCOMM2 Clock Enables LP_FLEXCOMM2 clock. 0b - No effect 1b - Enable

28.7.1.11 VDD2_COMP Peripheral Clock Control 3 Set (PSCCTL3_SET)

Offset

Register	Offset
PSCCTL3_SET	4Ch

Function

Writing an one to a bit position in PSCCTL3_SET sets the corresponding position in PSCCTL3.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												CMX_PER...	CMX_PER...	XSPI1_P...	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	TRACE	Reserved	Reserved	INPUTMU...	Reserved	FREQME0	Reserved	PVTS0	PINT0	Reserved				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19 CMX_PERFMON1	CMX_PERFMON1 Clock Enables CMX_PERFMON1 clock. 0b - No effect 1b - Enable
18 CMX_PERFMON0	CMX_PERFMON0 Clock Enables CMX_PERFMON0 clock. 0b - No effect 1b - Enable
17 XSPI1_PRINCE_XEX	XSPI1_PRINCE XEX Clock Enables XSPI1_PRINCE XEX clock. 0b - No effect 1b - Enable
16-15 —	Reserved
14 —	Reserved
13 TRACE	TRACE_RT700 Clock Enables TRACE_RT700 clock. 0b - No effect 1b - Enable
12 —	Reserved
11 —	Reserved
10 INPUTMUX0	INPUTMUX0 Clock Enables INPUTMUX0 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 —	Reserved
8 FREQME0	FREQME0 Clock Enables FREQME0 clock. 0b - No effect 1b - Enable
7 —	Reserved
6 PVTS0	PVTS0 Clock Enables PVTS0 clock. 0b - No effect 1b - Enable
5 PINT0	PINT0 Clock Enables PINT0 clock. 0b - No effect 1b - Enable
4-0 —	Reserved

28.7.1.12 VDD2_COMP Peripheral Clock Control 4 Set (PSCCTL4_SET)

Offset

Register	Offset
PSCCTL4_SET	50h

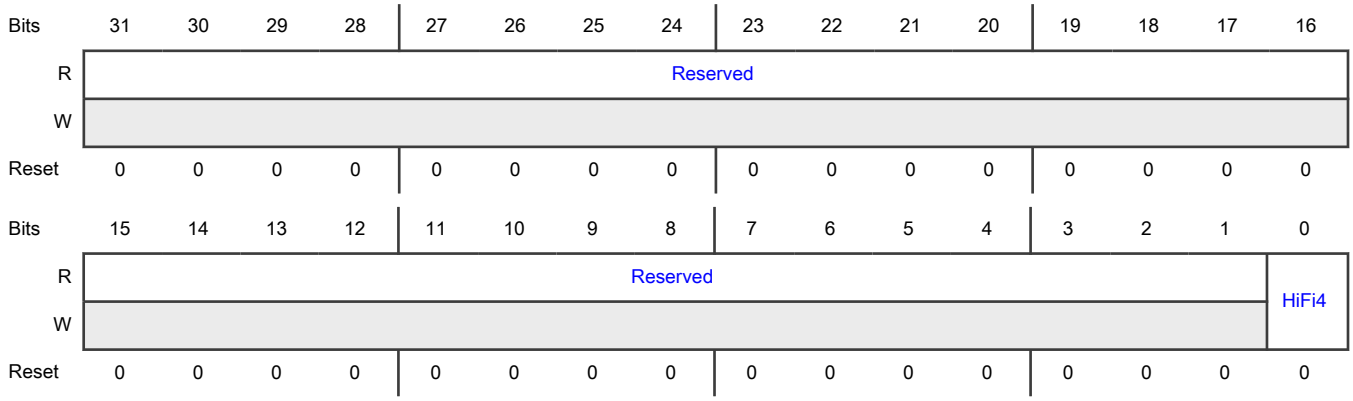
Function

Writing an one to a bit position in PSCCTL4_SET sets the corresponding position in PSCCTL4.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 HiFi4	HiFi4 Clock Enables HiFi4 clock. 0b - No effect 1b - Enable

28.7.1.13 VDD2_COMP Peripheral Clock Control 5 Set (PSCCTL5_SET)

Offset

Register	Offset
PSCCTL5_SET	54h

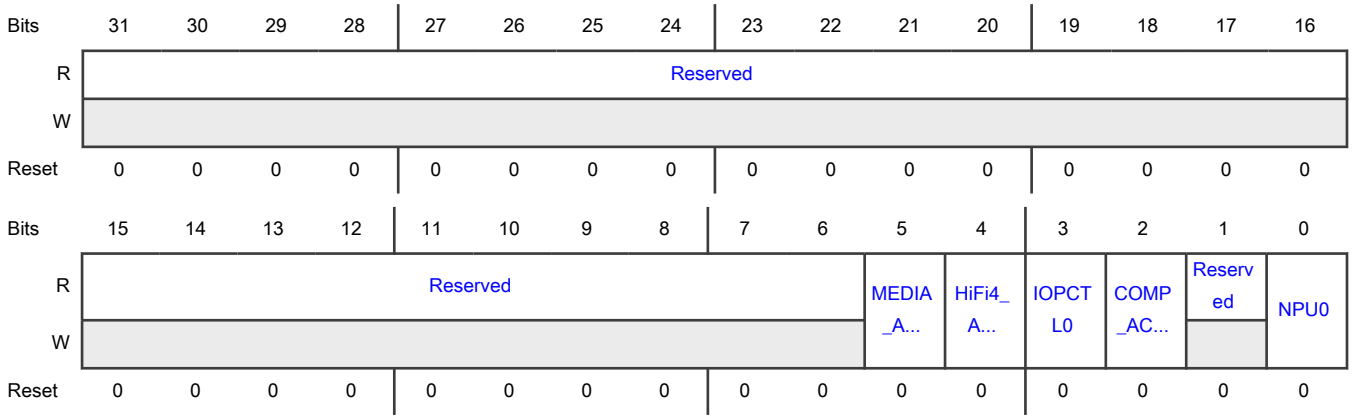
Function

Writing an one to a bit position in PSCCTL5_SET sets the corresponding position in PSCCTL5.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 MEDIA_ACCESS_RAM_ARBITER0	VDD2_Media Access RAM Arbiter 0 Clock Enables VDD2_MEDIA access RAM arbiter 0 clock. 0b - No effect 1b - Enable
4 HiFi4_ACCESS_RAM_ARBITER1	HiFi4 Access RAM Arbiter1 Clock Enables HiFi4 access RAM arbiter1 clock. 0b - No effect 1b - Enable
3 IOPCTL0	IOPCTL0 Clock Enables IOPCTL0 clock. 0b - No effect 1b - Enable
2 COMP_ACCESS_RAM_ARBITER1	VDD2_COMP Access RAM Arbiter1 Clock Enables VDD2_COMP access RAM arbiter1 clock. 0b - No effect 1b - Enable
1 —	Reserved
0	NPU0 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
NPU0	Enables NPU0 clock. 0b - No effect 1b - Enable

28.7.1.14 VDD2_COMP Peripheral Clock Control 0 Clear (PSCCTL0_CLR)

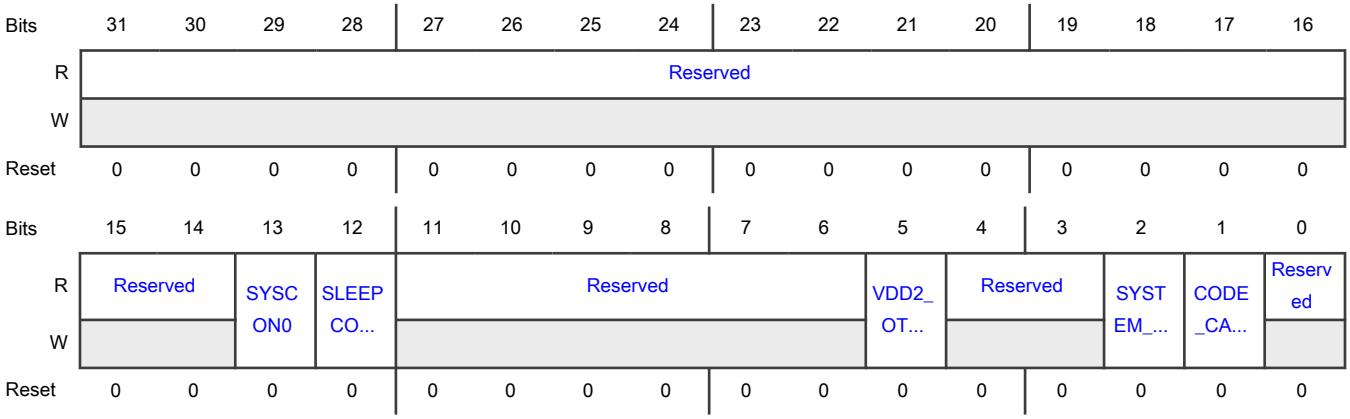
Offset

Register	Offset
PSCCTL0_CLR	70h

Function

Writing an one to a bit position in PSCCTL0_CLR clears the corresponding position in PSCCTL0.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-14 —	Reserved
13 SYSCON0	SYSCON0 Clock Disables SYSCON0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
12 SLEEPCON0	SLEEPCON0 Clock Disables SLEEPCON0 clock. 0b - No effect 1b - Disable
11-6 —	Reserved
5 VDD2_OTP0	VDD2_OTP_CTRL0 Clock Disables VDD2_OTP_CTRL0 clock. 0b - No effect 1b - Disable
4-3 —	Reserved
2 SYSTEM_CACHE	SYSTEM CACHE (XCACHE0)Clock Disables XCACHE0 clock. 0b - No effect 1b - Disable
1 CODE_CACHE	CODE CACHE (XCACHE1) Clock Disables XCACHE1 clock. 0b - No effect 1b - Disable
0 —	Reserved

28.7.1.15 VDD2_COMP Peripheral Clock Control 1 Clear (PSCCTL1_CLR)

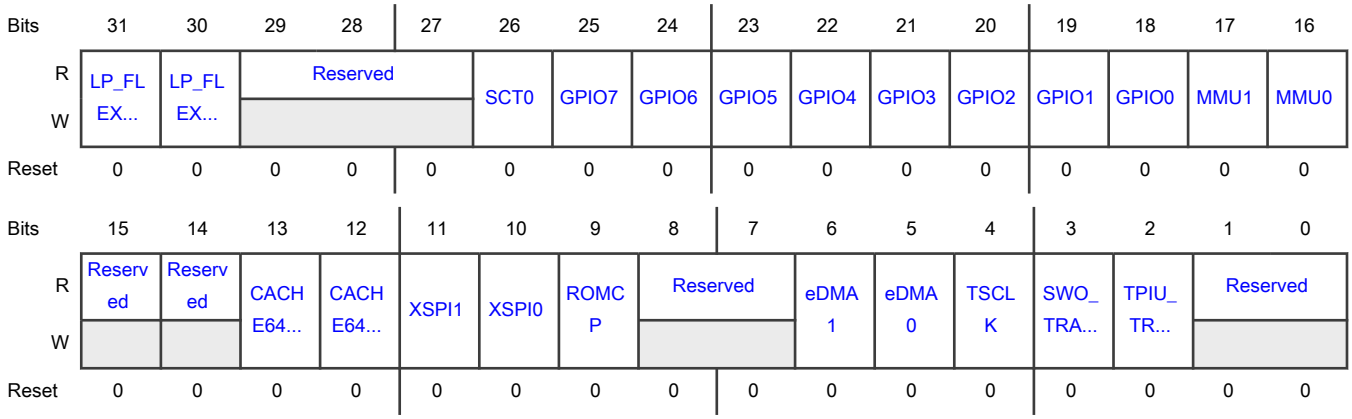
Offset

Register	Offset
PSCCTL1_CLR	74h

Function

Writing an one to a bit position in PSCCTL1_CLR clears the corresponding position in PSCCTL1.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Clock Disables LP_FLEXCOMM1 clock. 0b - No effect 1b - Disable
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Clock Disables LP_FLEXCOMM0 clock. 0b - No effect 1b - Disable
29-27 —	Reserved
26 SCT0	SCT0 Clock Disables SCT0 clock. 0b - No effect 1b - Disable
25 GPIO7	GPIO7 Clock Disables GPIO7 clock. 0b - No effect 1b - Disable
24 GPIO6	GPIO6 Clock Disables GPIO6 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable
23 GPIO5	GPIO5 Clock Disables GPIO5 clock. 0b - No effect 1b - Disable
22 GPIO4	GPIO4 Clock Disables GPIO4 clock. 0b - No effect 1b - Disable
21 GPIO3	GPIO3 Clock Disables GPIO3 clock. 0b - No effect 1b - Disable
20 GPIO2	GPIO2 Clock Disables GPIO2 clock. 0b - No effect 1b - Disable
19 GPIO1	GPIO1 Clock Disables GPIO1 clock. 0b - No effect 1b - Disable
18 GPIO0	GPIO0 Clock Disables GPIO0 clock. 0b - No effect 1b - Disable
17 MMU1	MMU1 Clock Disables MMU1 clock. 0b - No effect 1b - Disable
16 MMU0	MMU0 Clock Disables MMU0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
15 —	Reserved
14 —	Reserved
13 CACHE64_1	CACHE64_1 Clock Disables CACHE64_1 clock. 0b - No effect 1b - Disable
12 CACHE64_0	CACHE64_0 Clock Disables CACHE64_0 clock. 0b - No effect 1b - Disable
11 XSPI1	XSPI1 Clock Disables XSPI1 clock. 0b - No effect 1b - Disable
10 XSPI0	XSPI0 Clock Disables XSPI0 clock. 0b - No effect 1b - Disable
9 ROMCP	ROMCP Clock Disables ROMCP clock. 0b - No effect 1b - Disable
8-7 —	Reserved
6 eDMA1	eDMA1 Clock Disables eDMA1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
5 eDMA0	eDMA0 Clock Disables eDMA0 clock. 0b - No effect 1b - Disable
4 TSCLK	TS (TRACE_RT700) Clock Disables TRACE_RT700 clock. 0b - No effect 1b - Disable
3 SWO_TRACECLKIN	SWO TRACECLKIN (TRACE_RT700) Clock Disables SWO TRACE_RT700 clock. 0b - No effect 1b - Disable
2 TPIU_TRACE_CLKIN	TPIU TRACE_CLKIN (TRACE_RT700)Clock Disables TPIU TRACE_RT700 clock. 0b - No effect 1b - Disable
1-0 —	Reserved

28.7.1.16 VDD2_COMP Peripheral Clock Control 2 Clear (PSCCTL2_CLR)

Offset

Register	Offset
PSCCTL2_CLR	78h

Function

Writing an one to a bit position in PSCCTL2_CLR clears the corresponding position in PSCCTL2. This is a write-only-one register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MU4	SEMA42_4	Reserved		UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0	WWDT1	WWDT0	CRC0	I3C1	I3C0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAI2	SAI1	SAI0	Reserved	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MU4	MU4 Clock Disables MU4 clock. 0b - No effect 1b - Disable
30 SEMA42_4	SEMA42_4 Clock Disables SEMA42_4 clock. 0b - No effect 1b - Disable
29-28 —	Reserved
27 UTICK0	UTICK0 Clock Disables UTICK0 clock. 0b - No effect 1b - Disable
26 MRT0	MRT0 Clock Disables MRT0 clock. 0b - No effect 1b - Disable
25 CTIMER4	CTIMER4 Clock Disables CTIMER4 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable
24 CTIMER3	CTIMER3 Clock Disables CTIMER3 clock. 0b - No effect 1b - Disable
23 CTIMER2	CTIMER2 Clock Disables CTIMER2 clock. 0b - No effect 1b - Disable
22 CTIMER1	CTIMER1 Clock Disables CTIMER1 clock. 0b - No effect 1b - Disable
21 CTIMER0	CTIMER0 Clock Disables CTIMER0 clock. 0b - No effect 1b - Disable
20 WWDT1	WWDT1 Clock Disables WWDT1 clock. 0b - No effect 1b - Disable
19 WWDT0	WWDT0 Clock Disables WWDT0 clock. 0b - No effect 1b - Disable
18 CRC0	CRC0 Clock Disables CRC0 clock. 0b - No effect 1b - Disable
17 I3C1	I3C1 Clock Disables I3C1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
16 I3C0	I3C0 Clock Disables I3C0 clock. 0b - No effect 1b - Disable
15 SAI2	SAI2 Clock Disables SAI2 clock. 0b - No effect 1b - Disable
14 SAI1	SAI1 Clock Disables SAI1 clock. 0b - No effect 1b - Disable
13 SAI0	SAI0 Clock Disables SAI0 clock. 0b - No effect 1b - Disable
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Clock Disables LP_FLEXCOMM13 clock. 0b - No effect 1b - Disable
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Clock Disables LP_FLEXCOMM12 clock. 0b - No effect 1b - Disable
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Clock Disables LP_FLEXCOMM11 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
8 LP_FLEXCOMM10	LP_FLEXCOMM10 Clock Disables LP_FLEXCOMM10 clock. 0b - No effect 1b - Disable
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Clock Disables LP_FLEXCOMM9 clock. 0b - No effect 1b - Disable
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Clock Disables LP_FLEXCOMM8 clock. 0b - No effect 1b - Disable
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Clock Disables LP_FLEXCOMM7 clock. 0b - No effect 1b - Disable
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Clock Disables LP_FLEXCOMM6 clock. 0b - No effect 1b - Disable
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Clock Disables LP_FLEXCOMM5 clock. 0b - No effect 1b - Disable
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Clock Disables LP_FLEXCOMM4 clock. 0b - No effect 1b - Disable
1	LP_FLEXCOMM3 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM3	Disables LP_FLEXCOMM3 clock. 0b - No effect 1b - Disable
0 LP_FLEXCOMM2	LP_FLEXCOMM2 Clock Disables LP_FLEXCOMM2 clock 0b - No effect 1b - Disable

28.7.1.17 VDD2_COMP Peripheral Clock Control 3 Clear (PSCCTL3_CLR)

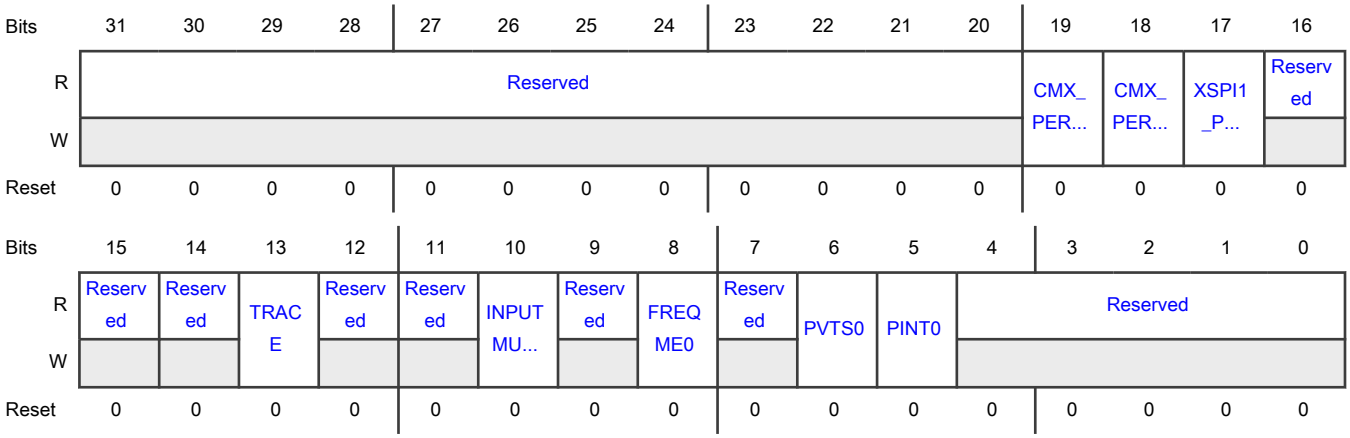
Offset

Register	Offset
PSCCTL3_CLR	7Ch

Function

Writing an one to a bit position in PSCCTL3_CLR clears the corresponding position in PSCCTL3.

Diagram



Fields

Field	Function
31-20 —	Reserved
19	CMX_PERFMON1 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
CMX_PERFMON1	Disables CMX_PERFMON1 clock. 0b - No effect 1b - Disable
18 CMX_PERFMON0	CMX_PERFMON0 Clock Disables CMX_PERFMON0 clock. 0b - No effect 1b - Disable
17 XSPI1_PRINCE_XEX	XSPI1 PRINCE_XEX Clock Disables XSPI1 PRINCE_XEX clock. 0b - No effect 1b - Disable
16-15 —	Reserved
14 —	Reserved
13 TRACE	TRACE Clock Disables TRACE_RT700 clock. 0b - No effect 1b - Disable
12 —	Reserved
11 —	Reserved
10 INPUTMUX0	INPUTMUX0 Clock Disables INPUTMUX0 clock. 0b - No effect 1b - Disable
9 —	Reserved
8	FREQME0 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
FREQME0	Disables FREQME0 clock. 0b - No effect 1b - Disable
7 —	Reserved
6 PVTS0	PVTS0 Clock Disables PVTS0 clock. 0b - No effect 1b - Disable
5 PINT0	PINT0 Clock Disables PINT0 clock. 0b - No effect 1b - Disable
4-0 —	Reserved

28.7.1.18 VDD2_COMP Peripheral Clock Control 4 Clear (PSCCTL4_CLR)

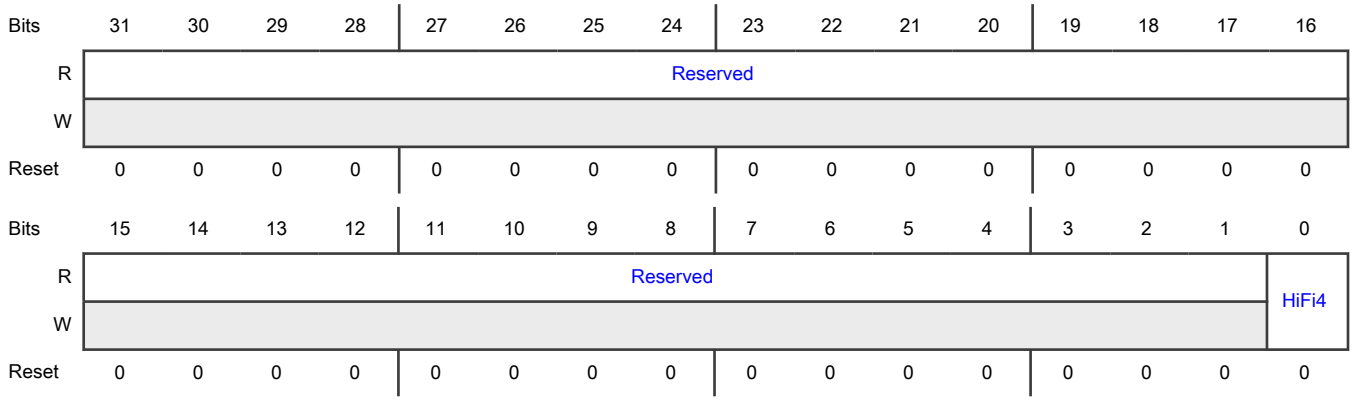
Offset

Register	Offset
PSCCTL4_CLR	80h

Function

Writing an one to a bit position in PSCCTL4_CLR clears the corresponding position in PSCCTL4.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 HiFi4	HiFi4 Clock Disables HiFi4 clock. 0b - No effect 1b - Disable

28.7.1.19 VDD2_COMP Peripheral Clock Control 5 Clear (PSCCTL5_CLR)

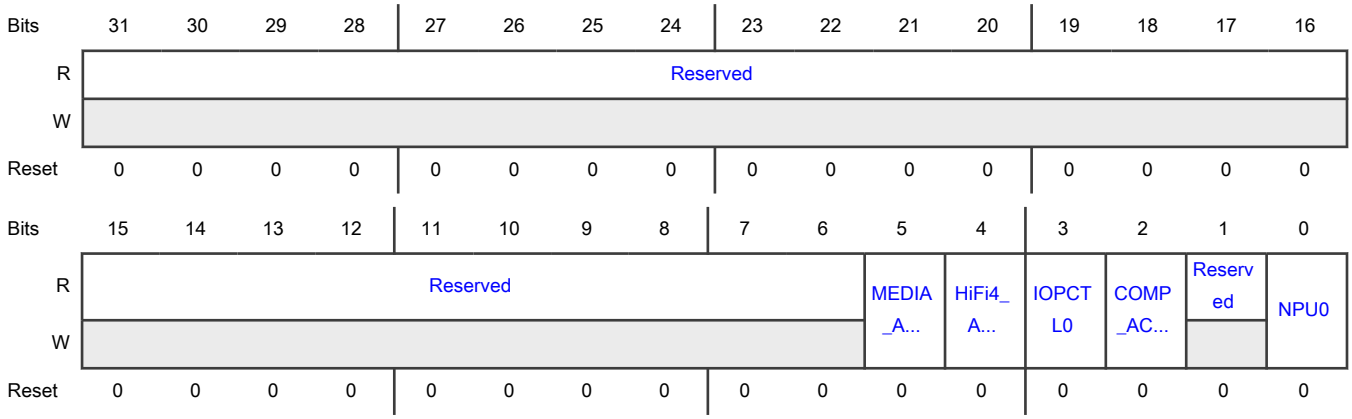
Offset

Register	Offset
PSCCTL5_CLR	84h

Function

Writing an one to a bit position in PSCCTL5_CLR clears the corresponding position in PSCCTL5.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 MEDIA_ACCESS_RAM_ARBITER0	VDD2_Media Access RAM Arbiter0 Clock Disables VDD2_MEDIA access RAM arbiter0 clock. 0b - No effect 1b - Disable
4 HiFi4_ACCESS_RAM_ARBITER1	HiFi4 Access RAM Arbiter1 Clock Disables HiFi4 access RAM arbiter1 clock. 0b - No effect 1b - Disable
3 IOPCTL0	IOPCTL0 Clock Disables IOPCTL0 clock. 0b - No effect 1b - Disable
2 COMP_ACCESS_RAM_ARBITER1	VDD2_COMP Access RAM Arbiter1 Clock Disables VDD2_COMP access RAM arbiter1 clock. 0b - No effect 1b - Disable
1 —	Reserved
0	NPU0 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
NPU0	Disables NPU0 clock. 0b - No effect 1b - Disable

28.7.1.20 FRO_TUNER0 and FRO_TUNER1 Clock Status (FRO01CLKSTATUS)

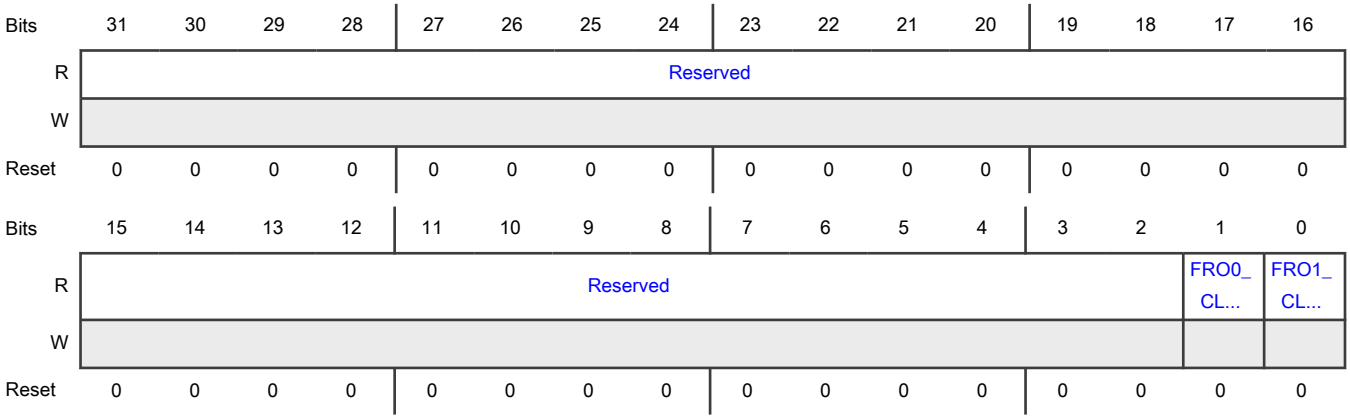
Offset

Register	Offset
FRO01CLKSTATUS	118h

Function

Indicates the status whether the FRO_TUNER0 and FRO_TUNER1 clocks have reached its final frequency.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 FRO0_CLK_OK	FRO_TUNER0 Clock OK Indicates the status whether the Free-Running Oscillator (FRO_TUNER0) reaches its final frequency. 0b - FRO_TUNER0 clock has not yet reached its final frequency. 1b - FRO_TUNER0 clock has reached its final frequency.
0	FRO_TUNER1 Clock OK

Table continues on the next page...

Table continued from the previous page...

Field	Function
FRO1_CLK_OK	Indicates the status whether the Free-Running Oscillator (FRO_TUNER1) reaches its final frequency. 0b - FRO_TUNER1 clock has not yet reached its final frequency. 1b - FRO_TUNER1 clock has reached its final frequency.

28.7.1.21 FRO0MAX Clock Domain Enable (FRO0MAXDOMAINEN)

Offset

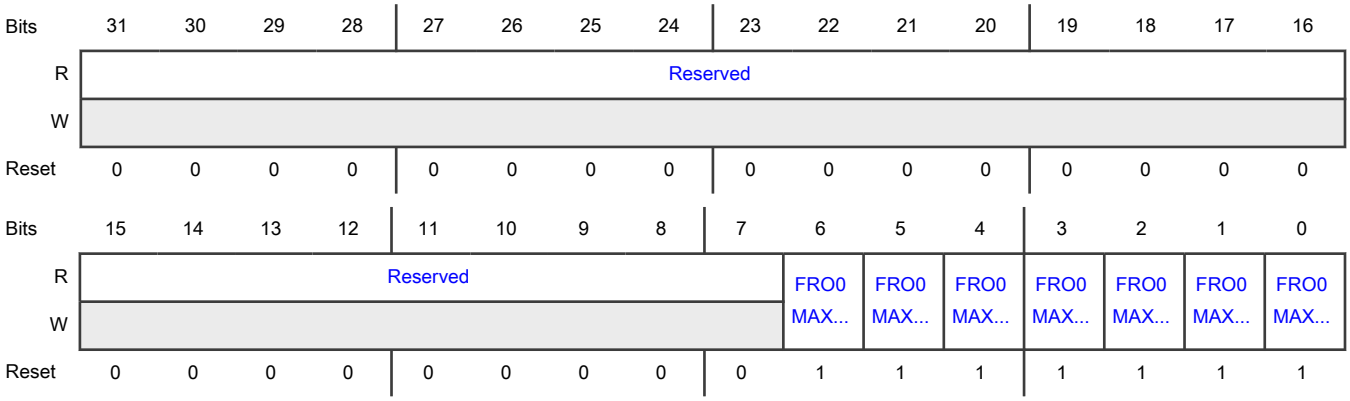
Register	Offset
FRO0MAXDOMAINEN	128h

Function

Enables fro0_max clock for various domains.

fro0_max clock can be sent to various domains for application. To reduce power consumption, when software configuration does not use this clock in one domain, turn off it in such domain.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 FRO0MAX_OF_COMN	fro0_max Clock of VDDN_COM Domain Enables fro0_max clock of VDDN_COM domain. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
5 FRO0MAX_OF_VDD2_COM	fro0_max Clock of VDD2_COM Domain Enables fro0_max clock of VDD2_COM domain. 0b - Disable 1b - Enable
4 FRO0MAX_OF_MDN	fro0_max Clock of VDDN_MEDIA Domain Enables fro0_max clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
3 FRO0MAX_OF_MD2	fro0_max Clock of VDD2_MEDIA Domain Enables fro0_max clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
2 FRO0MAX_OF_VDD2_DSP	fro0_max Clock of VDD2_DSP Domain Enables fro0_max clock of VDD2_DSP domain. 0b - Disable 1b - Enable
1 FRO0MAX_OF_SENSE	fro0_max Clock of VDD1_SENSE Domain Enables fro0_max clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
0 FRO0MAX_OF_CMPT	fro0_max Clock of VDD2_COMP Enables fro0_max clock of VDD2_COMP. 0b - Disable 1b - Enable

28.7.1.22 VDD2_COMP Main Clock Divider (MAINCLKDIV)

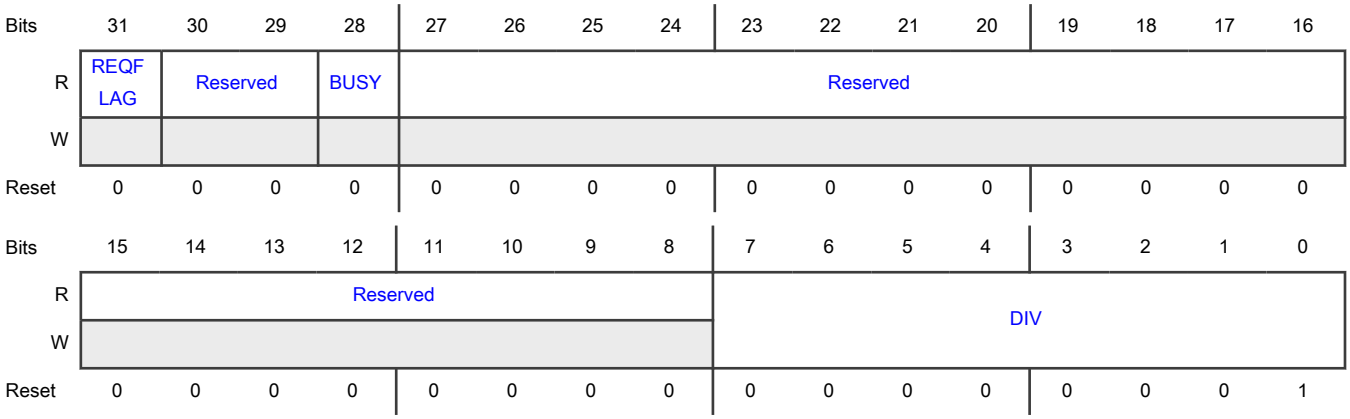
Offset

Register	Offset
MAINCLKDIV	400h

Function

Controls how the clock sources of the compute_main_clk is divided to provide system clock to the AHB bus, CPU, and memories.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.23 VDD2_COMP Base Clock Select Source (CMPTBASECLKSEL)

Offset

Register	Offset
CMPTBASECLKSEL	420h

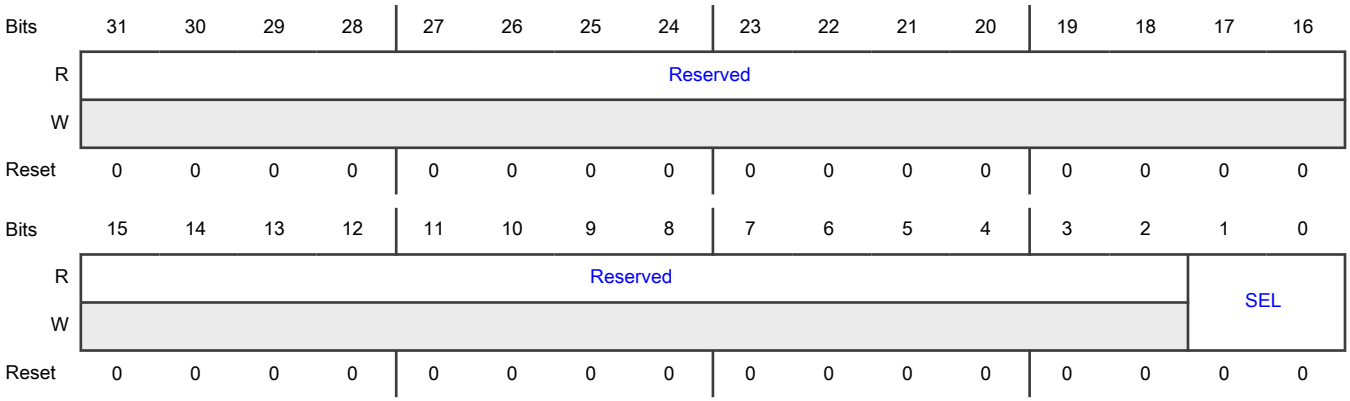
Function

Selects clock source for VDD2_COMP base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDD2_COMP Base Clock Select Source Selects a clock source for VDD2_COMP base clock. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.1.24 VDD2_DSP Base Clock Select Source (DSPBASECLKSEL)

Offset

Register	Offset
DSPBASECLKSEL	424h

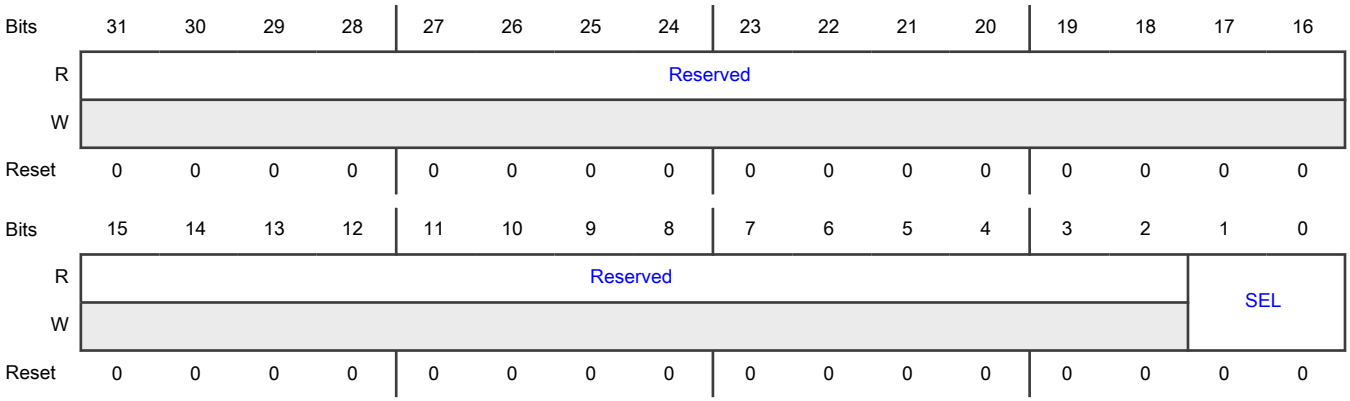
Function

Selects clock source for VDD2_DSP base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDD2_DSP Base Clock Select Source Selects clock source for VDD2_DSP base clock. When wakeup from VDD2_DSP domain power down, these two bit need to be re-written again to refresh the value which is written before power down. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.1.25 VDD2_COM Base Clock Select Source (VDD2COMBASECLKSEL)

Offset

Register	Offset
VDD2COMBASECLKSEL	428h

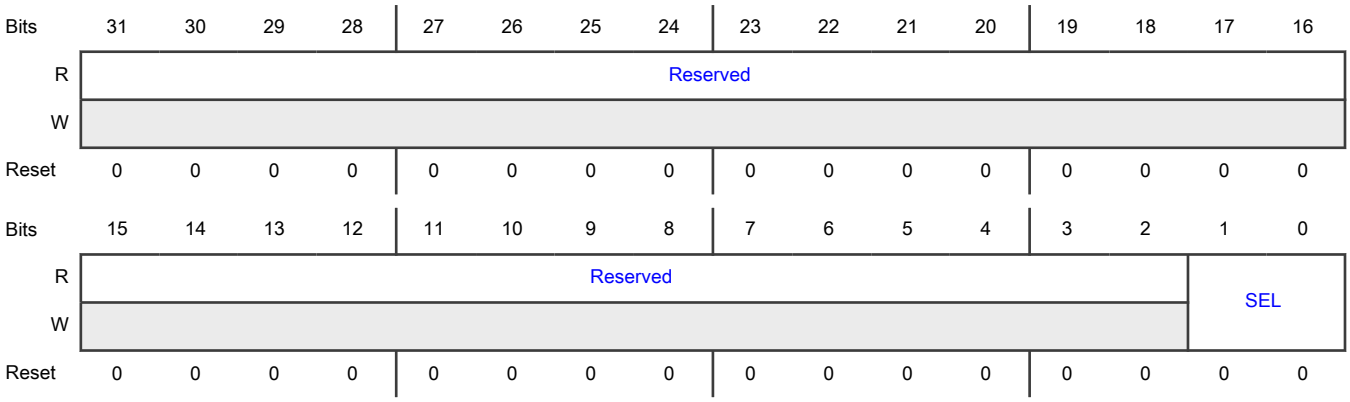
Function

Selects clock source for VDD2_COM base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDD2_COM Base Clock Select Source Selects clock source for VDD2_COM base clock. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.1.26 VDD2_COMP Main Clock Source Select (MAINCLKSEL)

Offset

Register	Offset
MAINCLKSEL	434h

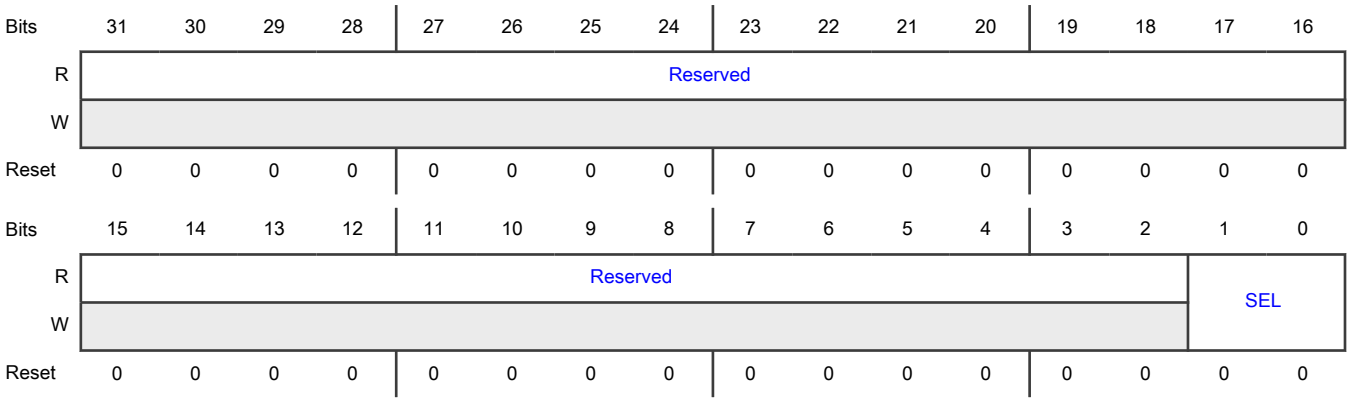
Function

Selects clock source for VDD2_COMP main clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDD2_COMP Main Clock Select Source Selects clock source for VDD2_COMP main clock. 00b - baseclk_cmpt 01b - main_pll_pfd0 10b - fro0_max 11b - audio_pll_pfd1

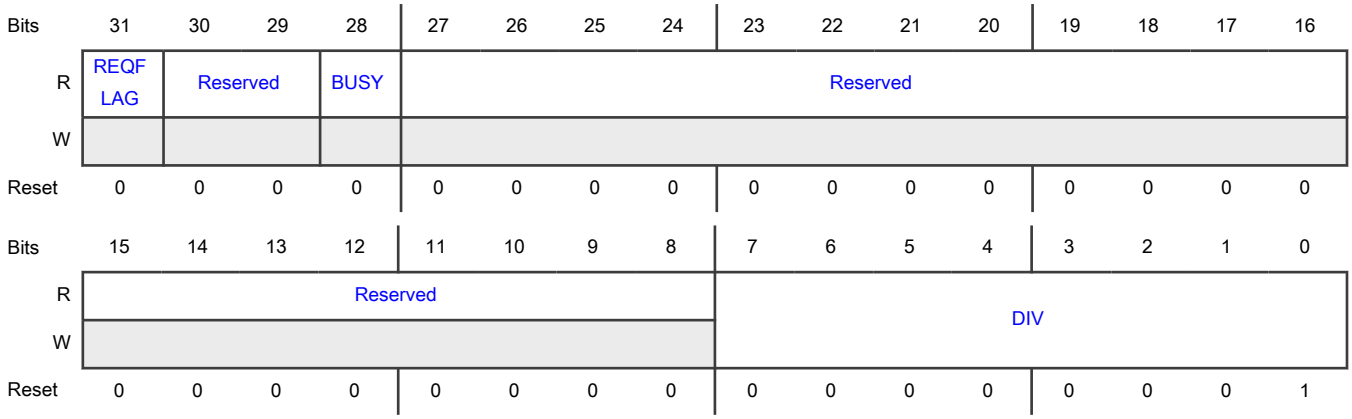
28.7.1.27 VDD2_DSP Clock Divider (DSPCPUCLKDIV)

Offset

Register	Offset
DSPCPUCLKDIV	440h

Function
Controls the divider for VDD2_DSP functional clock dsp_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0	Clock Divider Value Select

Table continues on the next page...

Table continued from the previous page...

Field	Function
DIV	Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.28 VDD2_DSP Clock Select Source (DSPCPUCLKSEL)

Offset

Register	Offset
DSPCPUCLKSEL	444h

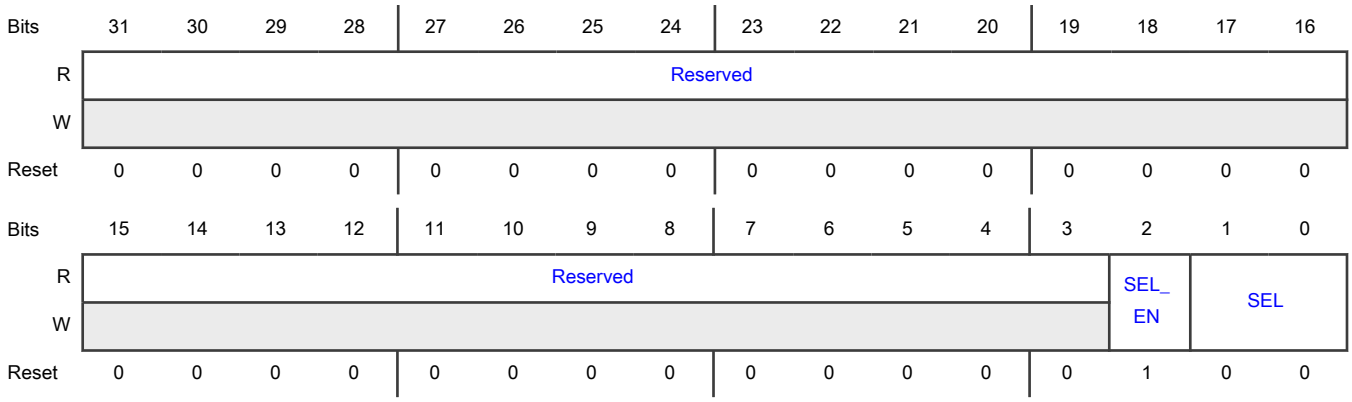
Function

Selects clock source for VDD2_DSP.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 SEL_EN	Clock Mux Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	VDD2_DSP Clock Select Source Selects clock source for VDD2_DSP. 00b - baseclk_dsp 01b - main_pll_pfd0 10b - fro0_max 11b - main_pll_pfd1

28.7.1.29 RAM Clock Select Source (RAMCLKSEL)

Offset

Register	Offset
RAMCLKSEL	450h

Function

Selects clock source for RAM in VDD2_COM. This clock is used for async access.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 SEL	RAM Clock Select Source Selects clock source for RAM. 00b - baseclk_com2 01b - main_pll_pfd0 10b - fro0_max 11b - fro1_max

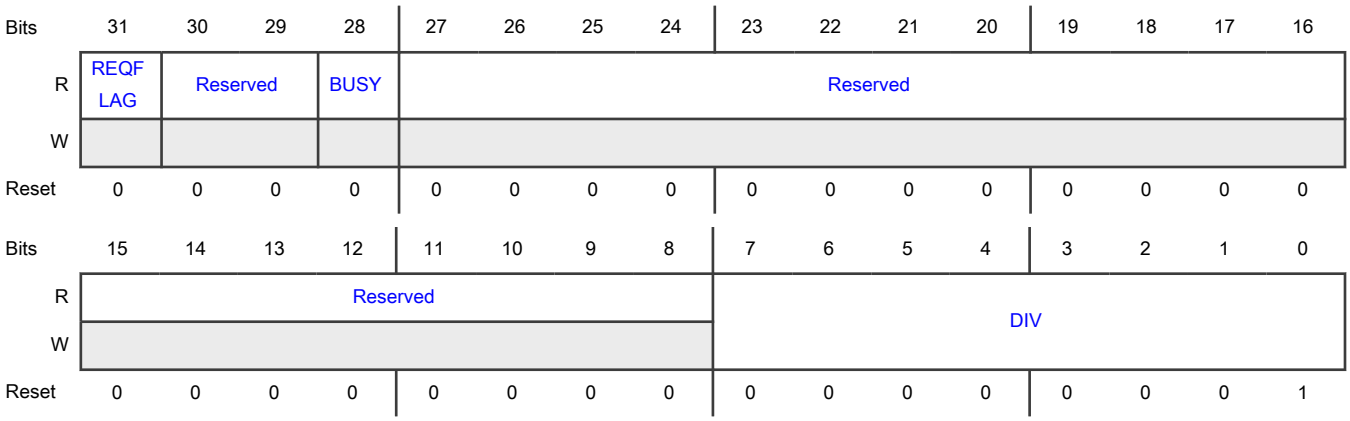
28.7.1.30 RAM Clock Divider (RAMCLKDIV)

Offset

Register	Offset
RAMCLKDIV	45Ch

Function
Controls the divider for the RAM function clock common_ram_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.31 TPIU (TRACE_RT700) Functional Clock Select Source (TPIUFCLKSEL)

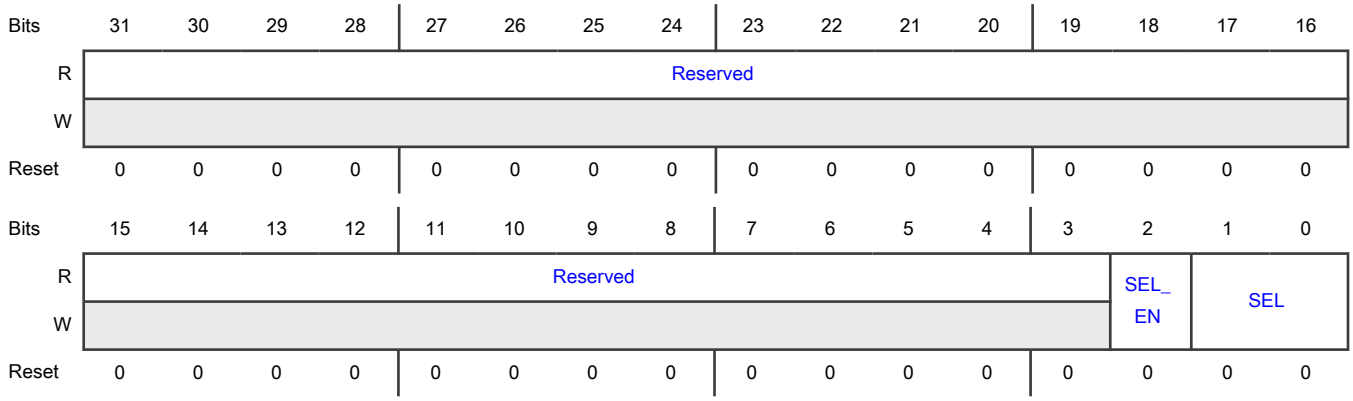
Offset

Register	Offset
TPIUFCLKSEL	560h

Function

Selects clock source for TPIU functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for TPIU clock to reduce power. 0b - Gates mux output for TPIU clock. 1b - Enables mux output for TPIU clock.
1-0 SEL	TPIU (TRACE_RT700) Clock Select Source Selects clock source for TPIU functional clock. 00b - baseclk_cmpt 01b - main_pll_pfd0 10b - fro0_max 11b - fro0_div6

28.7.1.32 TPIU (TRACE_RT700) Functional Clock Divider (TPIUCLKDIV)

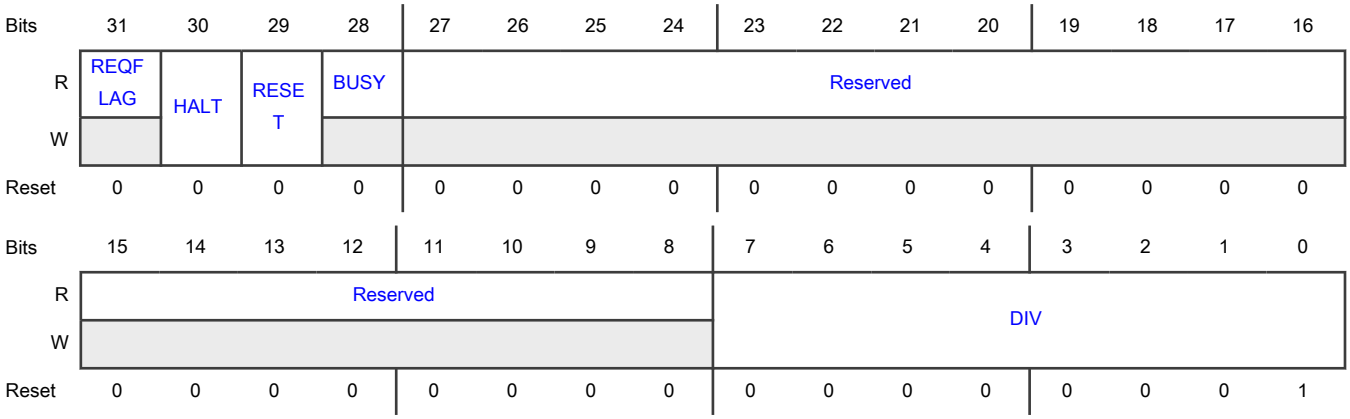
Offset

Register	Offset
TPIUCLKDIV	564h

Function

Controls the divider for the TPIU functional clock compute_tpiu_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.33 XSPI0 Functional Clock Source Select (XSPI0FCLKSEL)

Offset

Register	Offset
XSPI0FCLKSEL	600h

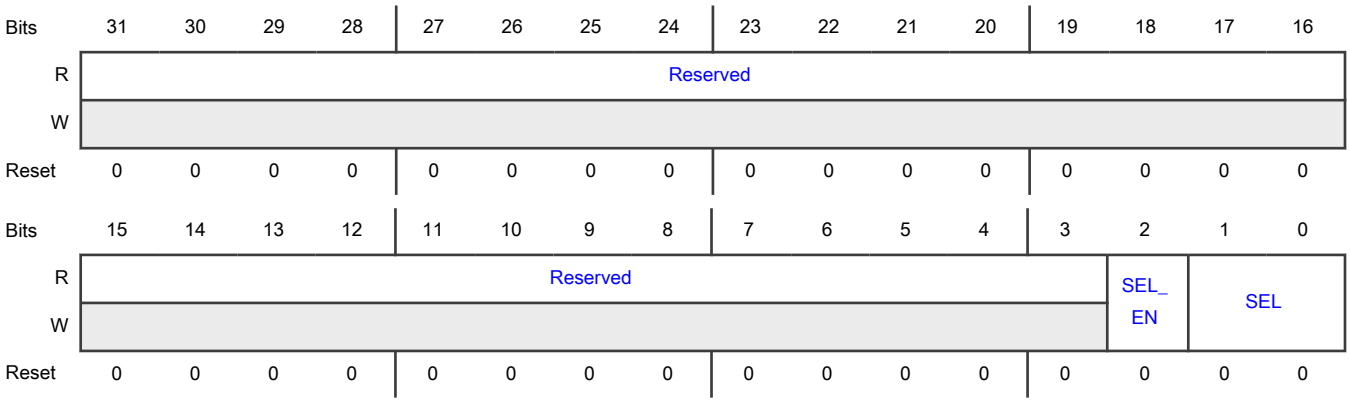
Function

Selects clock source for XSPI0 functional clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 SEL_EN	Clock Mux Output Enable Enables mux output for XSPI0 clock to reduce power. 0b - Gates mux output for XSPI0 clock. 1b - Enables mux output for XSPI0 clock.
1-0 SEL	XSPI0 Functional Clock Source Select Selects clock source for XSPI0 functional clock. 00b - baseclk_comn 01b - audio_pll_pfd0 10b - fro0_max 11b - main_pll_pfd1

28.7.1.34 XSPI0 Functional Clock Divider (XSPI0FCLKDIV)

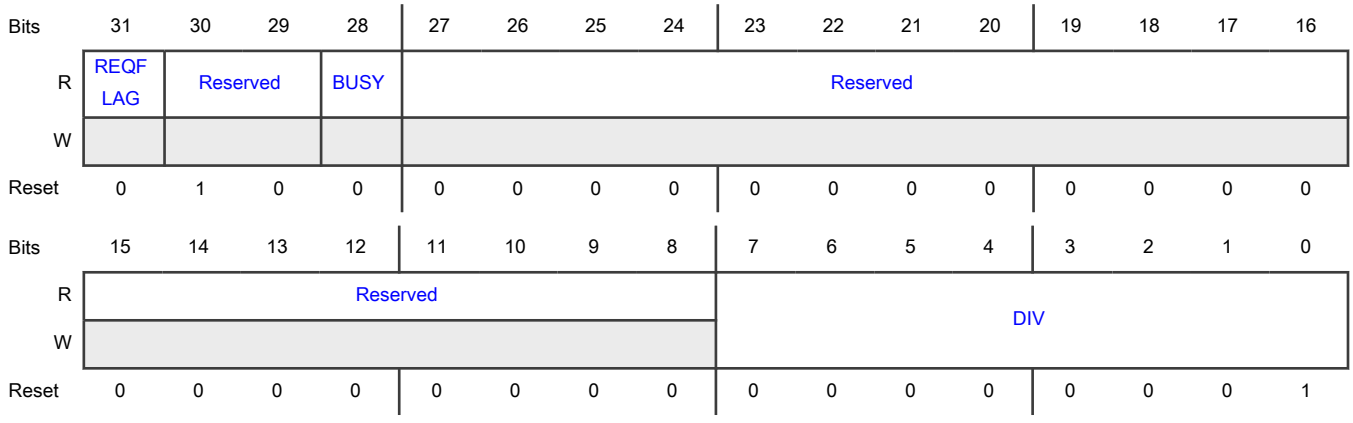
Offset

Register	Offset
XSPI0FCLKDIV	604h

Function

Controls the divider for the XSPI0 functional clock xspi0_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.35 XSPI1 Functional Clock Select Source (XSPI1FCLKSEL)

Offset

Register	Offset
XSPI1FCLKSEL	620h

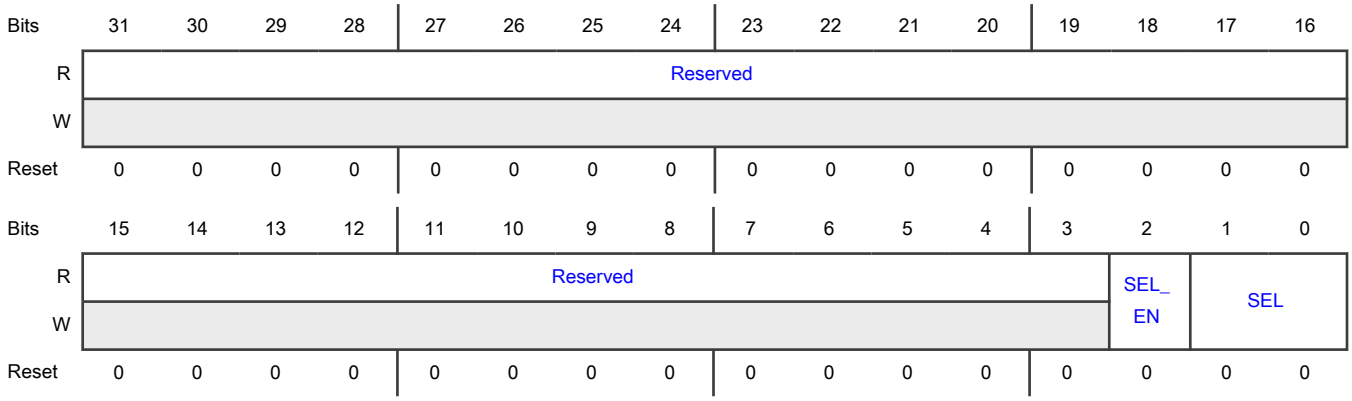
Function

Selects clock source for XSPI1 functional clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for XSPI 1 clock to reduce power. 0b - Gates mux output for XSPI1 clock. 1b - Enables mux output for XSPI1 clock.
1-0 SEL	Select Clock Source Selects clock source for XSPI1 functional clock. 00b - baseclk_comn 01b - audio_pll_pfd1 10b - fro0_max 11b - main_pll_pfd2

28.7.1.36 XSPI1 Functional Clock Divider (XSPI1FCLKDIV)

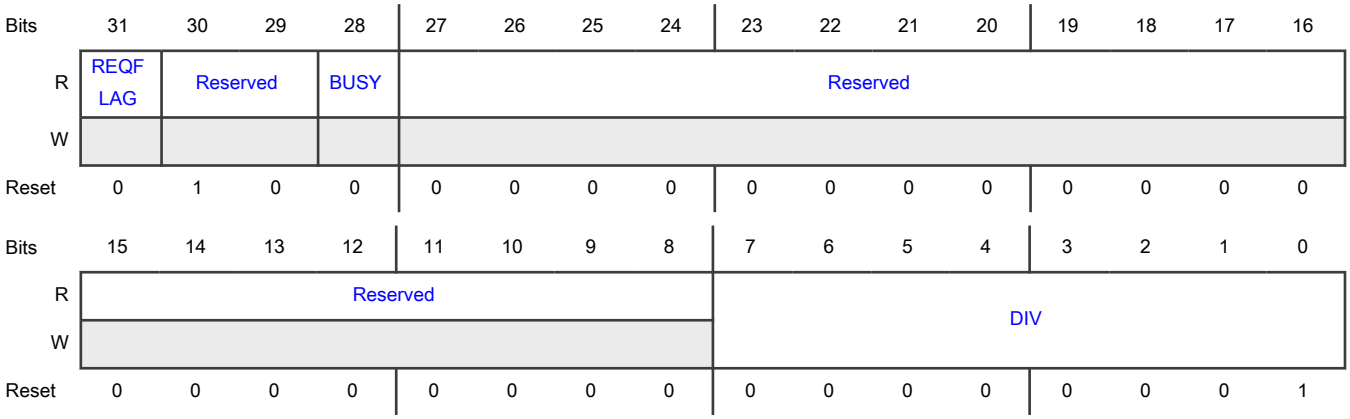
Offset

Register	Offset
XSPI1FCLKDIV	624h

Function

Controls the divider for the XSPI1 functional clock xspi1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

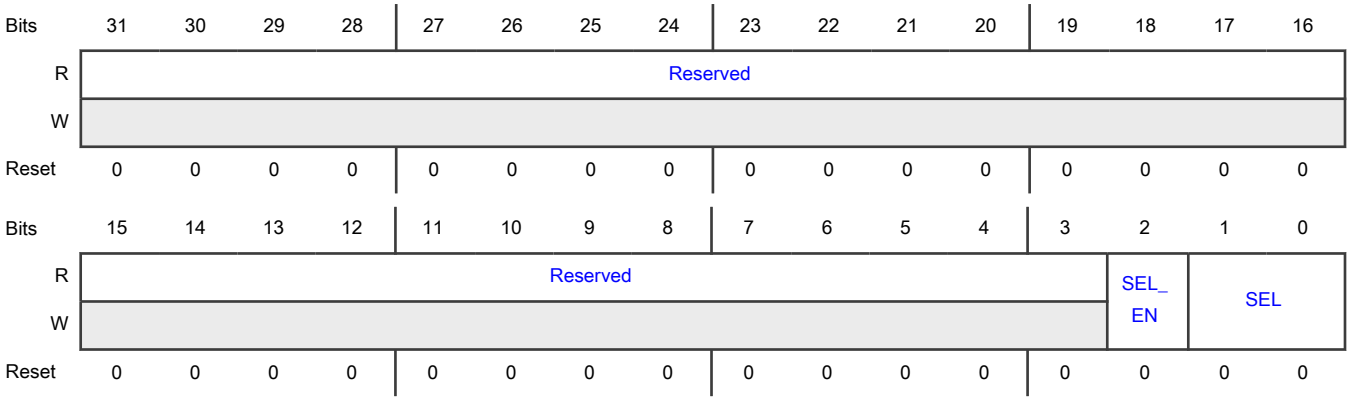
28.7.1.37 SCT Functional Clock Source Select (SCTFCLKSEL)

Offset

Register	Offset
SCTFCLKSEL	640h

Function
Selects the clock source for sct_fclk.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables mux output for SCT clock to reduce power. 0b - Gates mux output for SCT clock. 1b - Enables mux output for SCT clock.
1-0 SEL	SCT Functional Clock Source Select Selects the clock source for SCT clock only. 00b - baseclk_cmpt 01b - main_pll_pfd0 10b - fro0_max 11b - fro0_div6

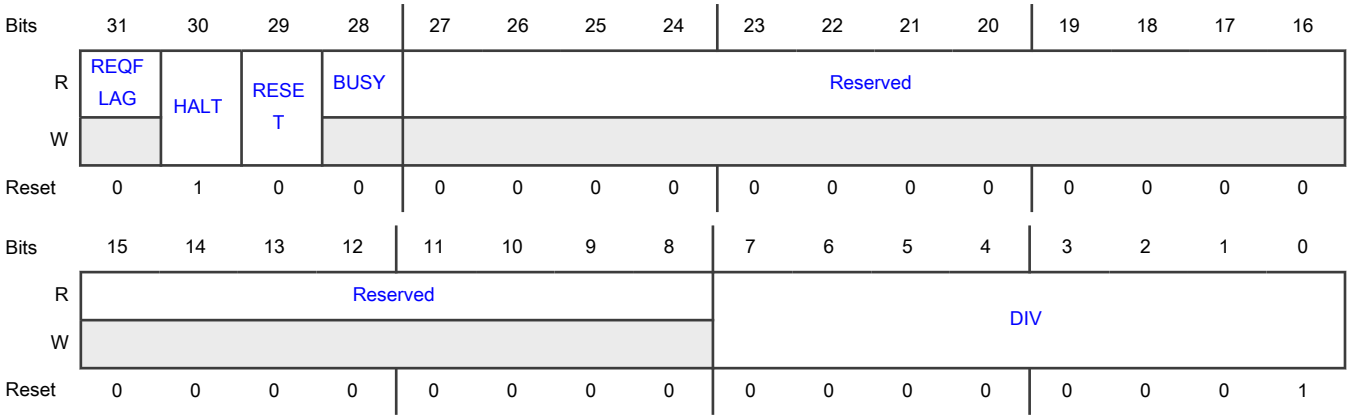
28.7.1.38 SCT Functional Clock Divider (SCTFCLKDIV)

Offset

Register	Offset
SCTFCLKDIV	644h

Function
Controls the divider for sct_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Reset the divider counter
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

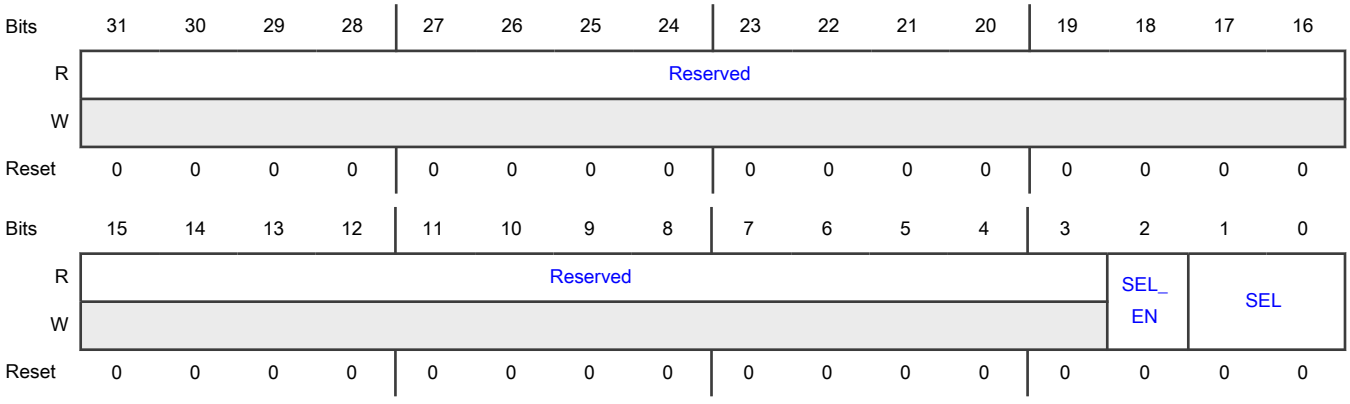
28.7.1.39 UTICK0 Functional Clock Select Source (UTICK0FCLKSEL)

Offset

Register	Offset
UTICK0FCLKSEL	700h

Function
Selects the clock source for UTICK0.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for UTICK0 clock to reduce power. 0b - Gates mux output for UTICK0 clock. 1b - Enables mux output for UTICK0 clock.
1-0 SEL	Select Clock Source Selects the clock source for UTICK0. 00b - baseclk_cmpt 01b - 1m_lposc 10b - fro0_max 11b - fro1_div2

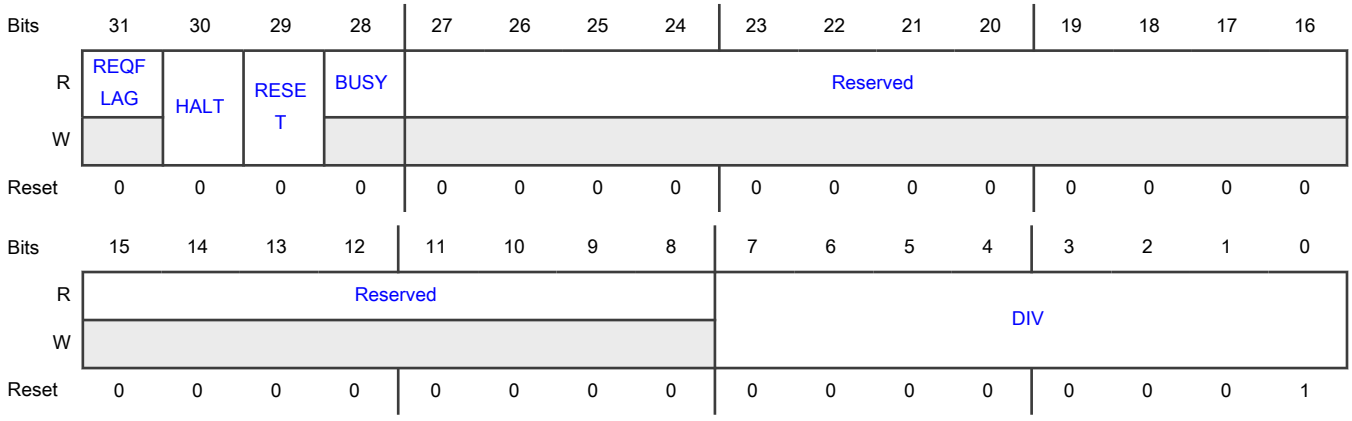
28.7.1.40 UTICK0 Functional Clock Divider (UTICK0FCLKDIV)

Offset

Register	Offset
UTICK0FCLKDIV	704h

Function
Controls the divider for the UTICK0 functional clock utick0_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

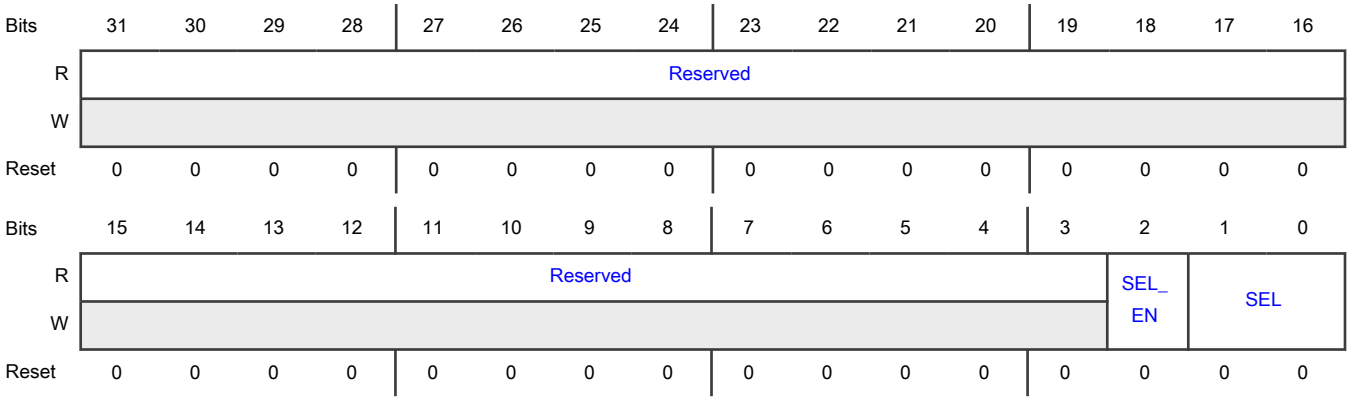
28.7.1.41 WWDT0 Functional Clock Source Select (WWDT0FCLKSEL)

Offset

Register	Offset
WWDT0FCLKSEL	720h

Function
Selects the clock source for WWDT0 functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables mux output for WWDT0 clock to reduce power. 0b - Gates mux output for WWDT0 clock. 1b - Enables mux output for WWDT0 clock.
1-0 SEL	Select Clock Source Selects the clock source for watchdog timer 0. 00b - 1m_lposc 01b - Reserved 10b - Reserved 11b - Reserved

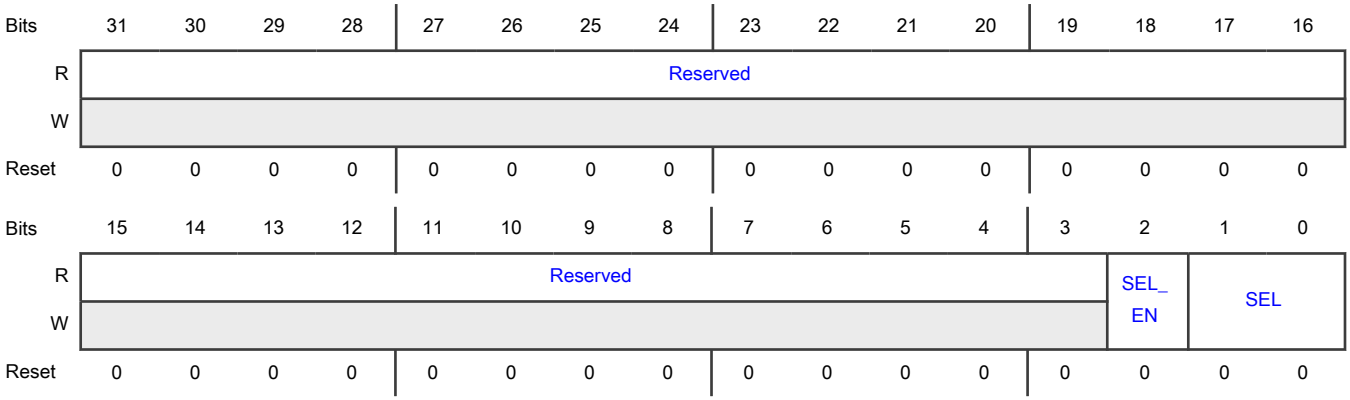
28.7.1.42 WWDT1 Functional Clock Source Select (WWDT1FCLKSEL)

Offset

Register	Offset
WWDT1FCLKSEL	740h

Function
Selects the clock source for the WWDT1 functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables mux output for WWDT1 clock to reduce power. 0b - Gates mux output for WWDT1 clock. 1b - Enables mux output for WWDT1 clock.
1-0 SEL	WWDT1 Functional Clock Source Select Selects the clock source for the WWDT1 functional clock. 00b - 1m_lposc 01b - Reserved 10b - Reserved 11b - Reserved

28.7.1.43 SYSTICK Functional Clock Source Select (SYSTICKFCLKSEL)

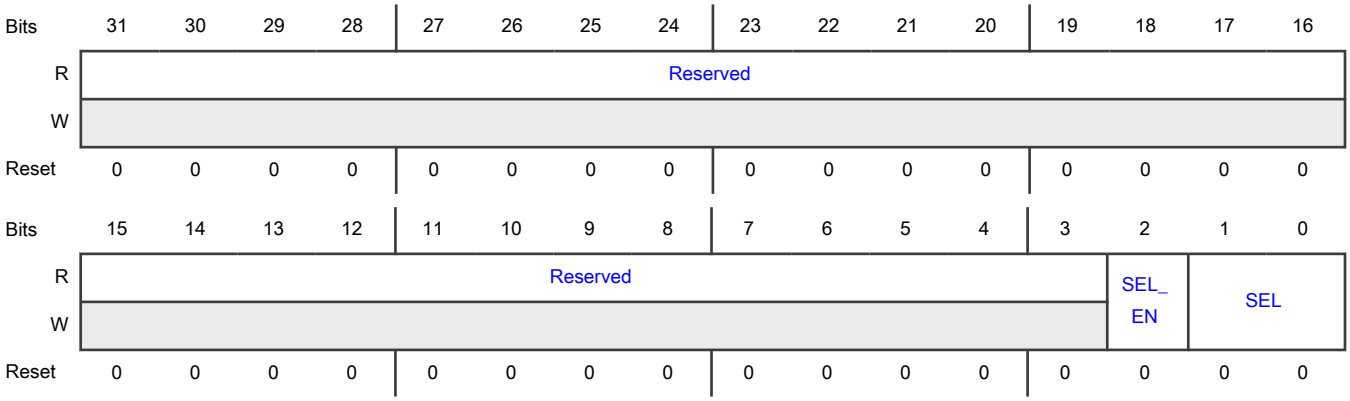
Offset

Register	Offset
SYSTICKFCLKSEL	760h

Function

Selects the clock source for the Systick timer.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for SYSTICK clock to reduce power. 0b - Gates mux output for SYSTICK clock. 1b - Enables mux output for SYSTICK clock.
1-0 SEL	Select Clock Source for SYSTICK Selects the clock source for the Systick timer. 00b - baseclk_cmpt 01b - 1m_lposc 10b - wake32k_clk (switch to other clock source before VDD2_COMP enters into the Deep Sleep Retention mode.) 11b - osc_clk

28.7.1.44 SYSTICK Functional Clock Divider (SYSTICKFCLKDIV)

Offset

Register	Offset
SYSTICKFCLKDIV	764h

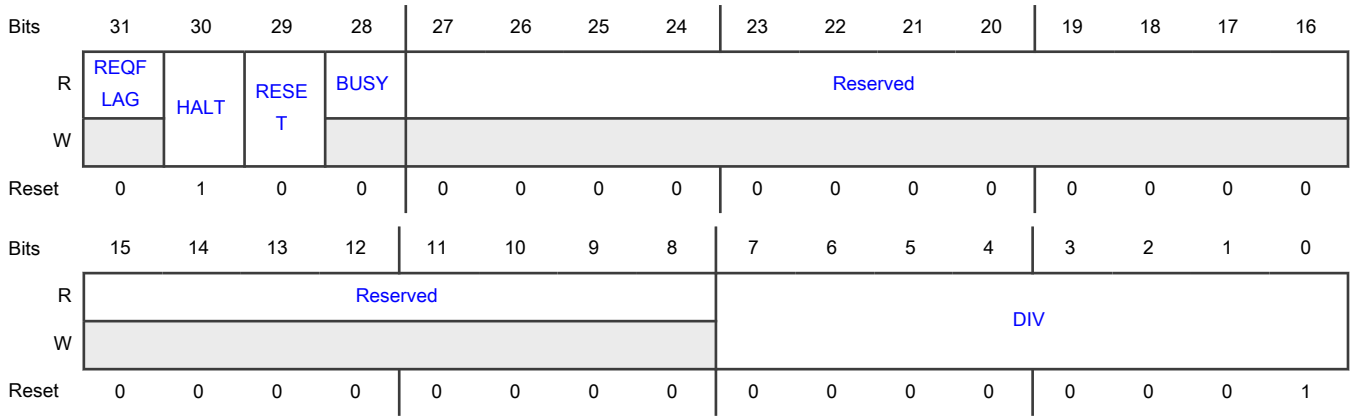
Function

Controls the divider for the Systick functional clock compute_systick_fclk.

NOTE

The clock frequency resulting from the Systick Clock Divider cannot be equal to or higher than the CPU clock frequency resulting from the CPU Clock Divider for divide by 1 to work.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

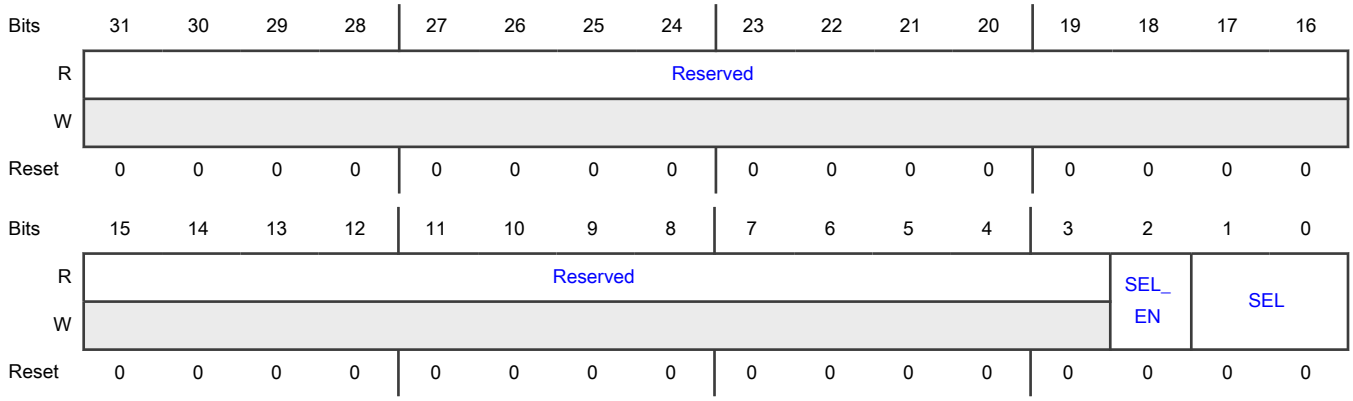
28.7.1.45 LP_FLEXCOMM 0 to 13 Clock Source 0 Select (FCCLK0SEL)

Offset

Register	Offset
FCCLK0SEL	800h

Function
Selects the clock source 0 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for FC clock 0 to reduce power. 0b - Gates mux output for FC clock 0. 1b - Enables mux output for FC clock 0.
1-0 SEL	LP_FLEXCOMM 0 to 13 Source Clock 0 Select Selects the clock source for FC Clock 0. 00b - baseclk_cmpt 01b - fro0_max 10b - main_pll_pfd3 11b - osc_clk

28.7.1.46 LP_FLEXCOMM 0 to 13 Clock Source 0 Divider (FCCLK0DIV)

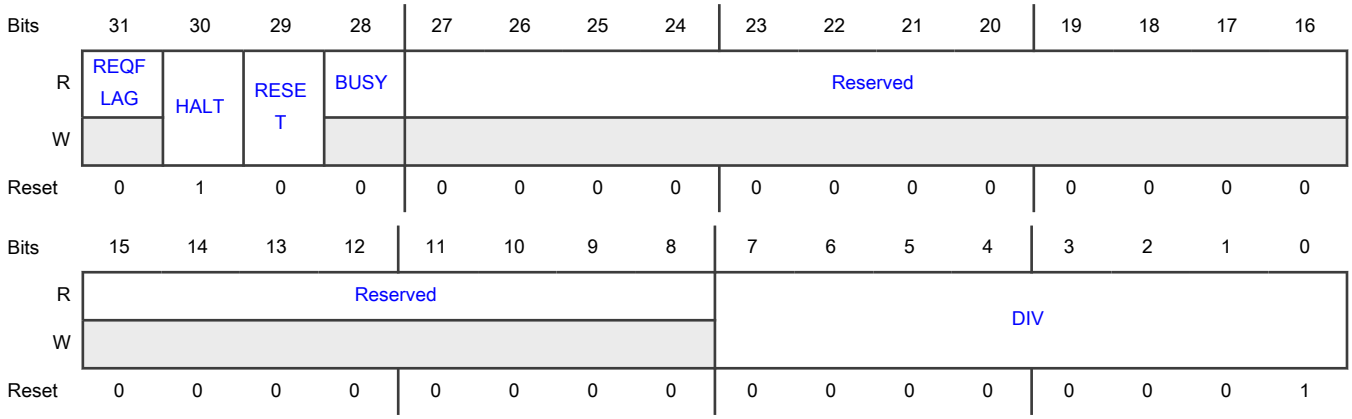
Offset

Register	Offset
FCCLK0DIV	804h

Function

Controls the divider of clock source 0 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.47 LP_FLEXCOMM0 Clock Source Select (FC0FCLKSEL)

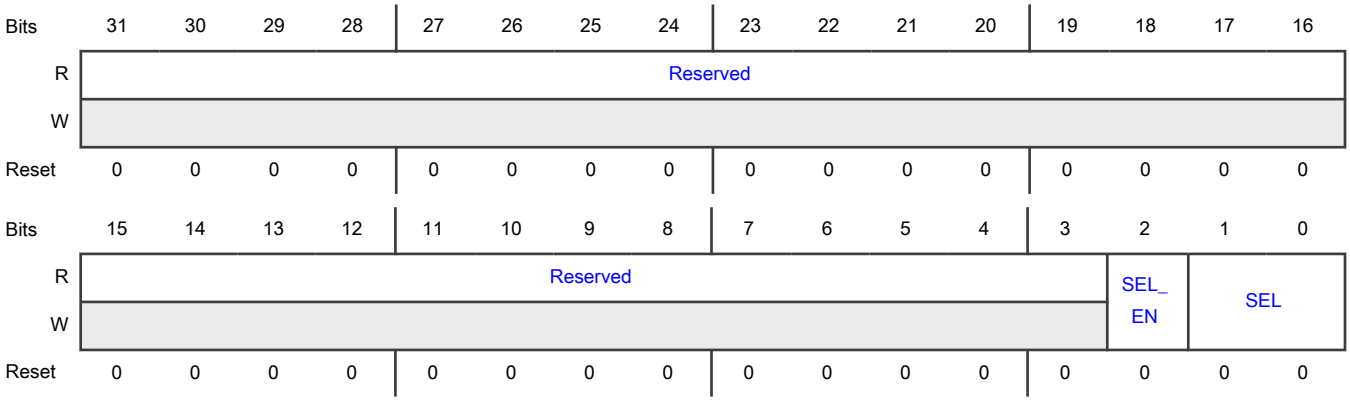
Offset

Register	Offset
FC0FCLKSEL	808h

Function

Selects the clock source for LP_FLEXCOMM0.

Diagram



Fields

Field	Function
31-3 —	Reserved
2	Clock Mux Output Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEL_EN	Enables mux output for LP_FLEXCOMM0 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM0 clock. 1b - Enables mux output for LP_FLEXCOMM0 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

28.7.1.48 LP_FLEXCOMM 0 to 13 Clock Source 1 Select (FCCLK1SEL)

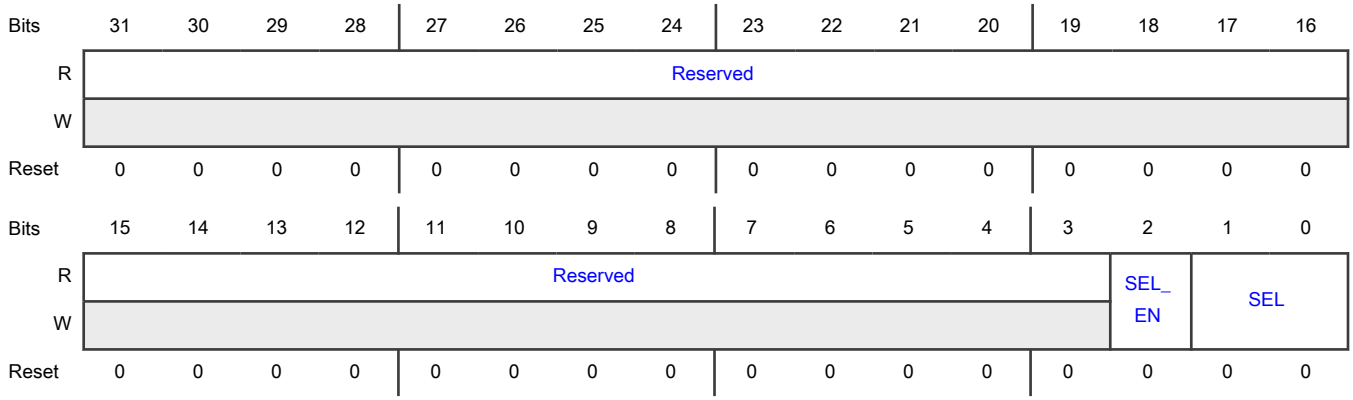
Offset

Register	Offset
FCCLK1SEL	820h

Function

Selects the clock source 1 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31-3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 SEL_EN	Clock Mux Output Enable Enables mux output for FC clock 1 to reduce power. 0b - Gates mux output for FC clock 1. 1b - Enables mux output for FC clock 1.
1-0 SEL	LP_FLEXCOMM 0 to 13 Source Clock 1 Select Selects the clock source for LP_FLEXCOMM 0 to 13 source clock 1. 00b - baseclk_cmpt 01b - fro0_max 10b - main_pll_pfd3 11b - osc_clk

28.7.1.49 LP_FLEXCOMM 0 to 13 Clock Source 1 Divider (FCCLK1DIV)

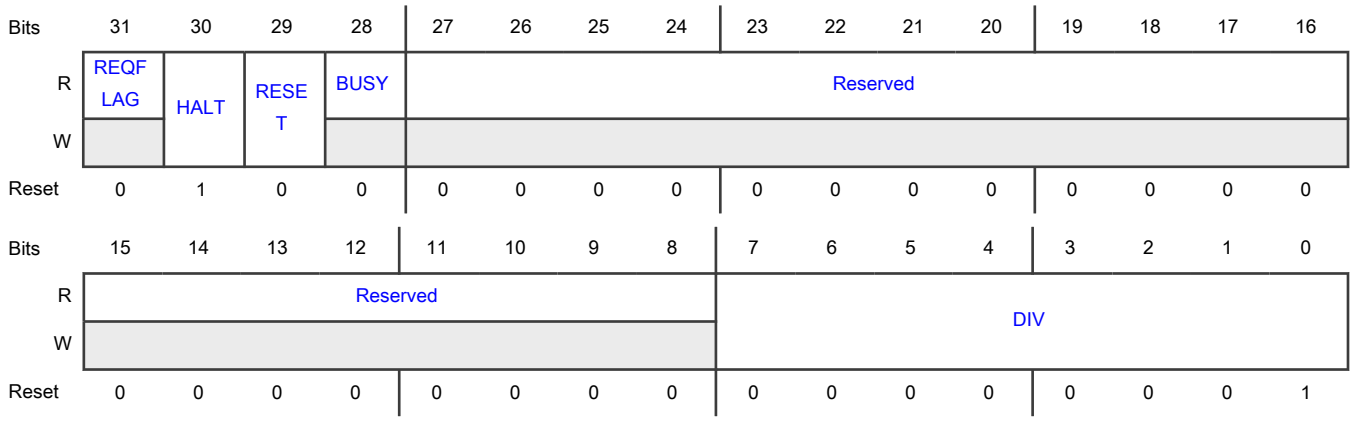
Offset

Register	Offset
FCCLK1DIV	824h

Function

Controls the divider of clock source 1 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

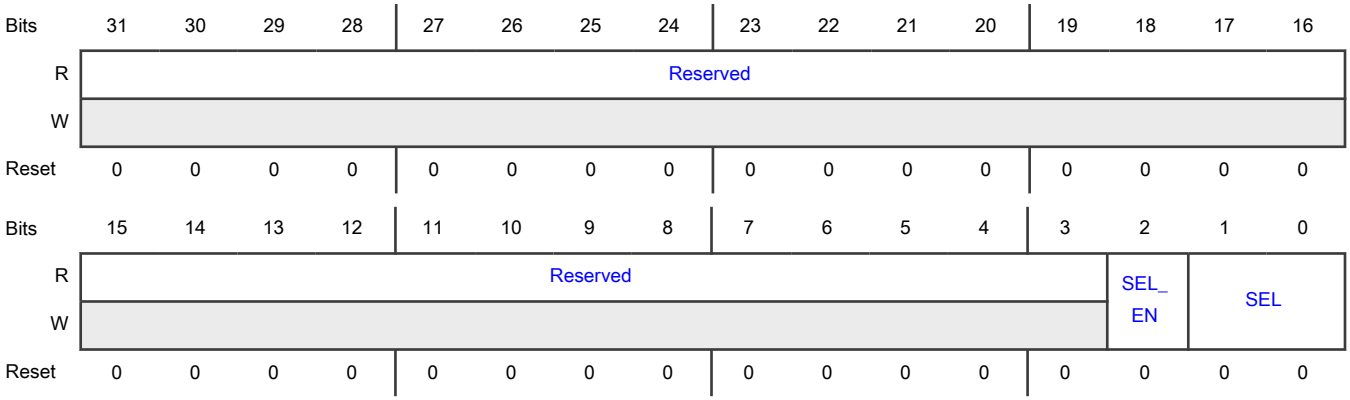
28.7.1.50 LP_FLEXCOMM1 Clock Source Select (FC1FCLKSEL)

Offset

Register	Offset
FC1FCLKSEL	828h

Function
Selects the clock source for LP_FLEXCOMM1.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM1 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM1 clock. 1b - Enables mux output for LP_FLEXCOMM1 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

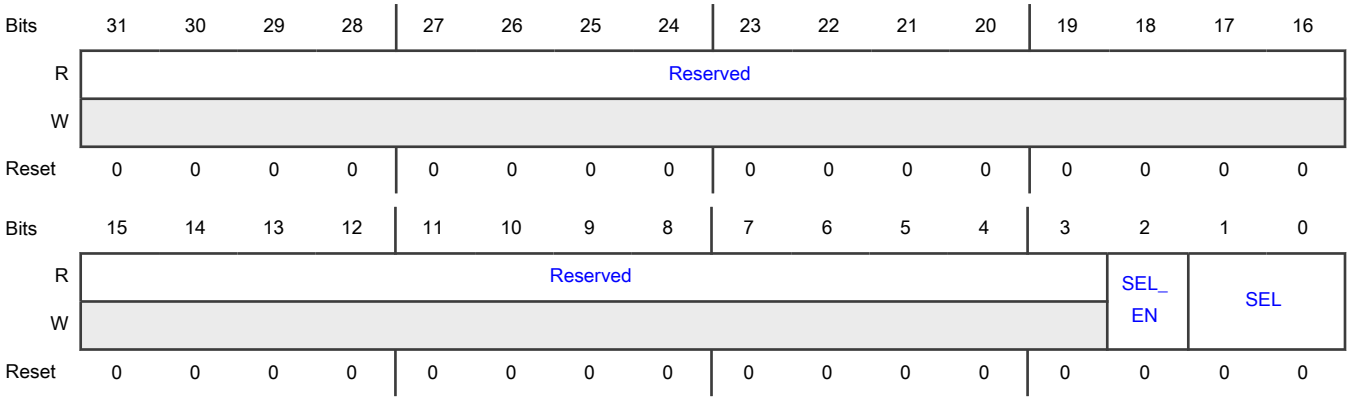
28.7.1.51 LP_FLEXCOMM 0 to 13 Clock Source 2 Select (FCCLK2SEL)

Offset

Register	Offset
FCCLK2SEL	840h

Function
Selects the clock source 2 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for FC clock 2 to reduce power. 0b - Gates mux output for FC clock 2. 1b - Enables mux output for FC clock 2.
1-0 SEL	LP_FLEXCOMM 0 to 13 Source Clock 2 Select Selects the clock source for LP_FLEXCOMM 0 to 13 source clock 2. 00b - baseclk_cmpt 01b - fro0_max 10b - main_pll_pfd3 11b - osc_clk

28.7.1.52 LP_FLEXCOMM 0 to 13 Clock Source 2 Divider (FCCLK2DIV)

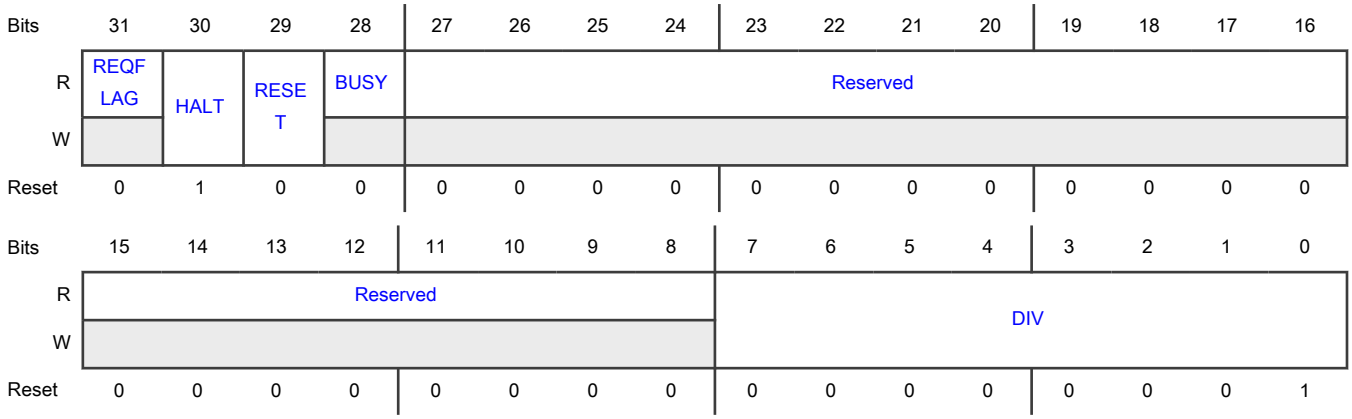
Offset

Register	Offset
FCCLK2DIV	844h

Function

Controls the divider of clock source 2 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.53 LP_FLEXCOMM2 Clock Source Select (FC2FCLKSEL)

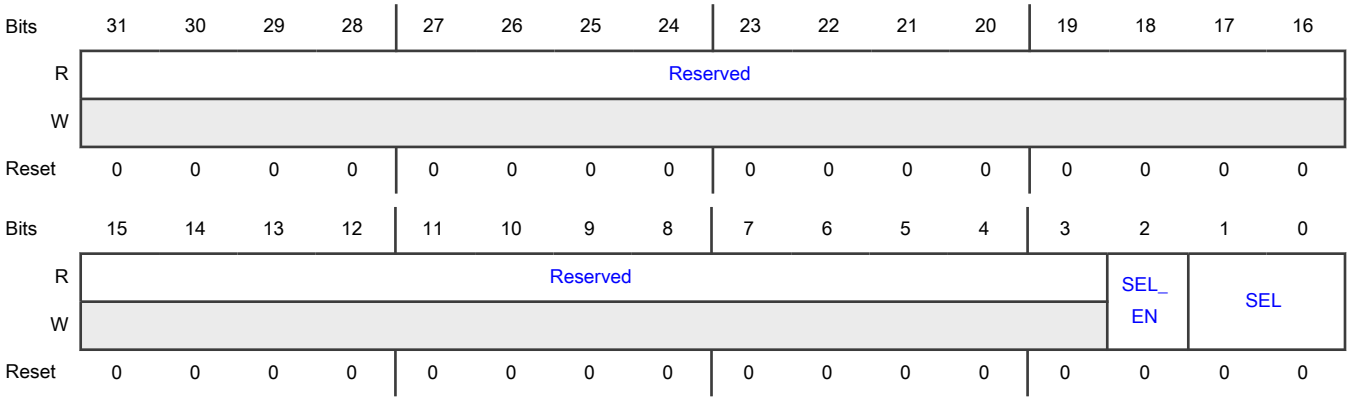
Offset

Register	Offset
FC2FCLKSEL	848h

Function

Selects the clock source for LP_FLEXCOMM2.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM2 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM2 clock. 1b - Enables mux output for LP_FLEXCOMM2 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

28.7.1.54 LP_FLEXCOMM 0 to 13 Clock Source 3 Select (FCCLK3SEL)

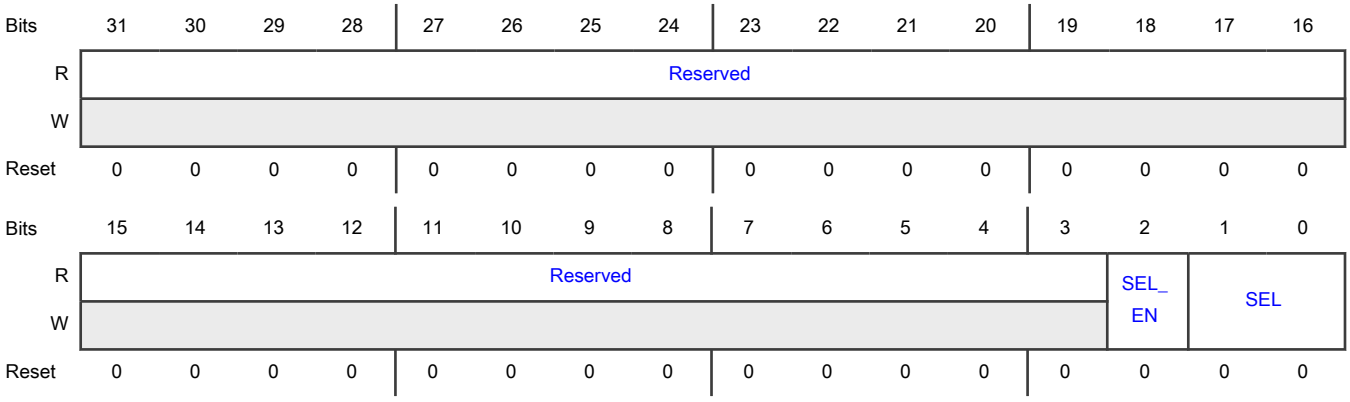
Offset

Register	Offset
FCCLK3SEL	860h

Function

Selects the clock source 3 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for FC clock 3 to reduce power. 0b - Gates mux output for FC clock 3. 1b - Enables mux output for FC clock 3.
1-0 SEL	LP_FLEXCOMM 0 to 13 Source Clock 3 Select Selects the clock source for LP_FLEXCOMM 0 to 13 source clock 3. 00b - baseclk_cmpt 01b - fro0_max 10b - main_pll_pfd3 11b - osc_clk

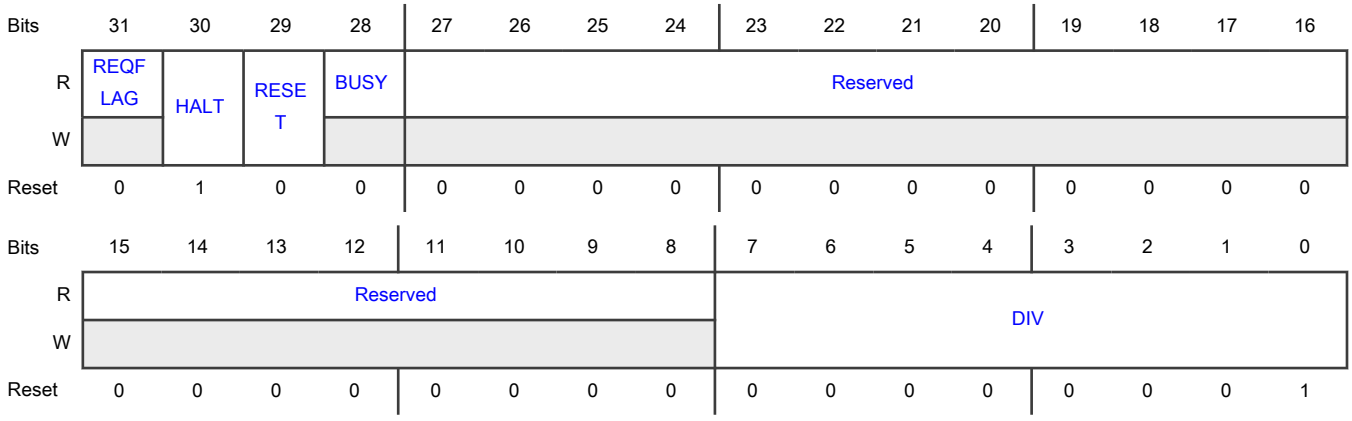
28.7.1.55 LP_FLEXCOMM 0 to 13 Clock Source 3 Divider (FCCLK3DIV)

Offset

Register	Offset
FCCLK3DIV	864h

Function
Controls the divider of clock source 3 for LP_FLEXCOMM 0 to 13.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

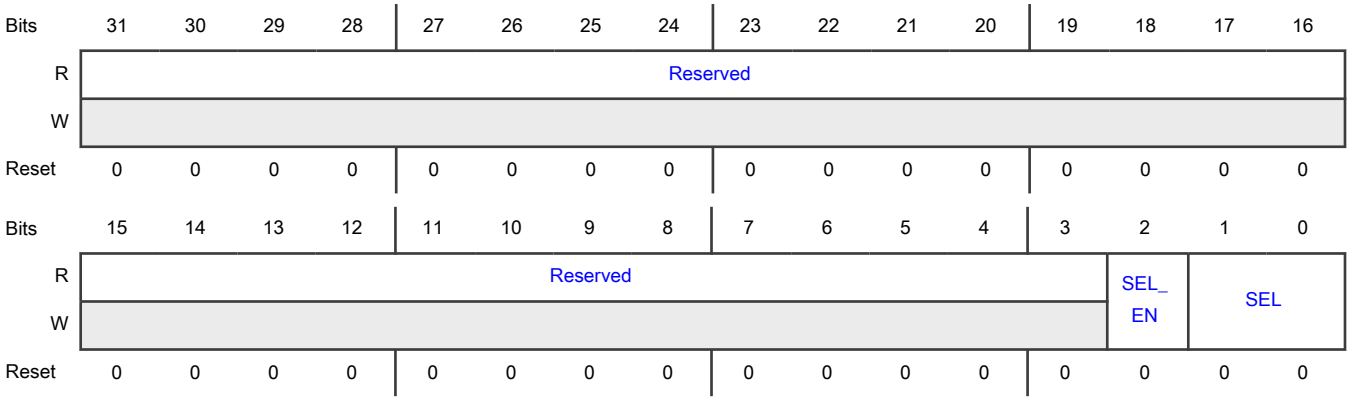
28.7.1.56 LP_FLEXCOMM3 Clock Source Select (FC3FCLKSEL)

Offset

Register	Offset
FC3FCLKSEL	868h

Function
Selects the clock source for LP_FLEXCOMM3.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM3 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM3 clock. 1b - Enables mux output for LP_FLEXCOMM3 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

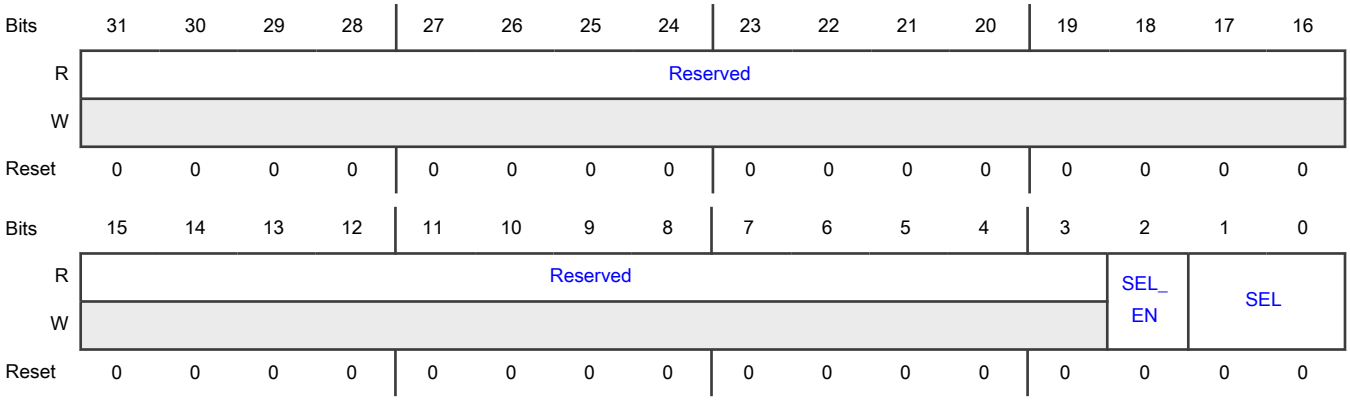
28.7.1.57 LP_FLEXCOMM4 Clock Source Select (FC4FCLKSEL)

Offset

Register	Offset
FC4FCLKSEL	888h

Function
Selects the clock source for LP_FLEXCOMM4.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM4 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM4 clock. 1b - Enables mux output for LP_FLEXCOMM4 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

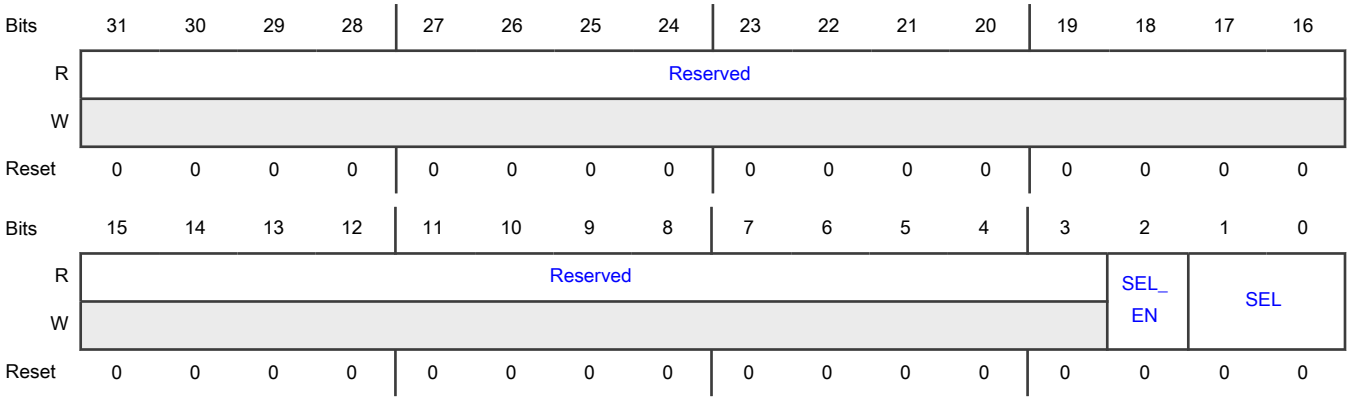
28.7.1.58 LP_FLEXCOMM5 Clock Source Select (FC5FCLKSEL)

Offset

Register	Offset
FC5FCLKSEL	8A8h

Function
Selects the clock source for LP_FLEXCOMM5.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM5 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM5 clock. 1b - Enables mux output for LP_FLEXCOMM5 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

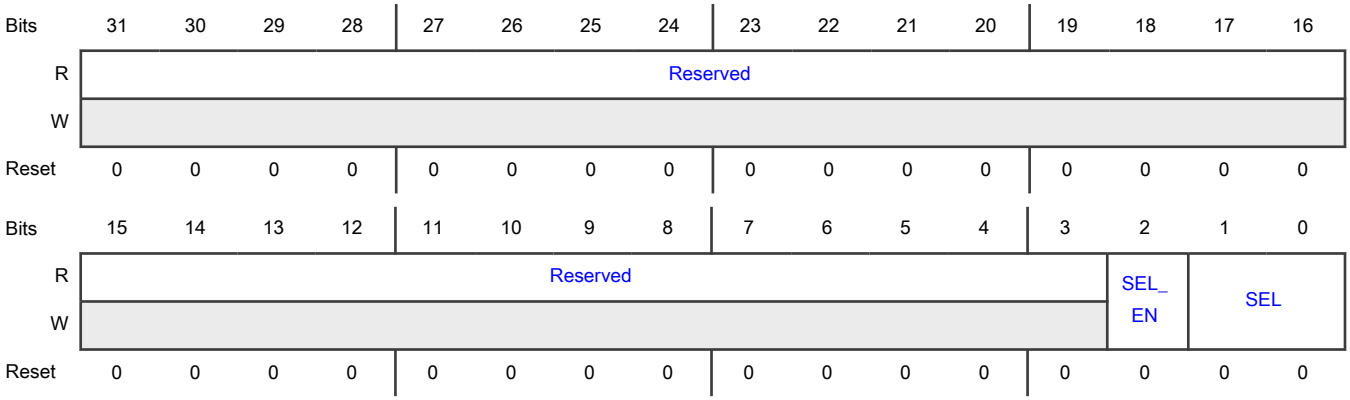
28.7.1.59 LP_FLEXCOMM6 Clock Source Select (FC6FCLKSEL)

Offset

Register	Offset
FC6FCLKSEL	8C8h

Function
Selects the clock source for LP_FLEXCOMM6.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM6 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM6 clock. 1b - Enables mux output for LP_FLEXCOMM6 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

28.7.1.60 LP_FLEXCOMM7 Clock Source Select (FC7FCLKSEL)

Offset

Register	Offset
FC7FCLKSEL	8E8h

Function
Selects the clock source for LP_FLEXCOMM7.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													SEL_EN	SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM7 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM7 clock. 1b - Enables mux output for LP_FLEXCOMM7 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

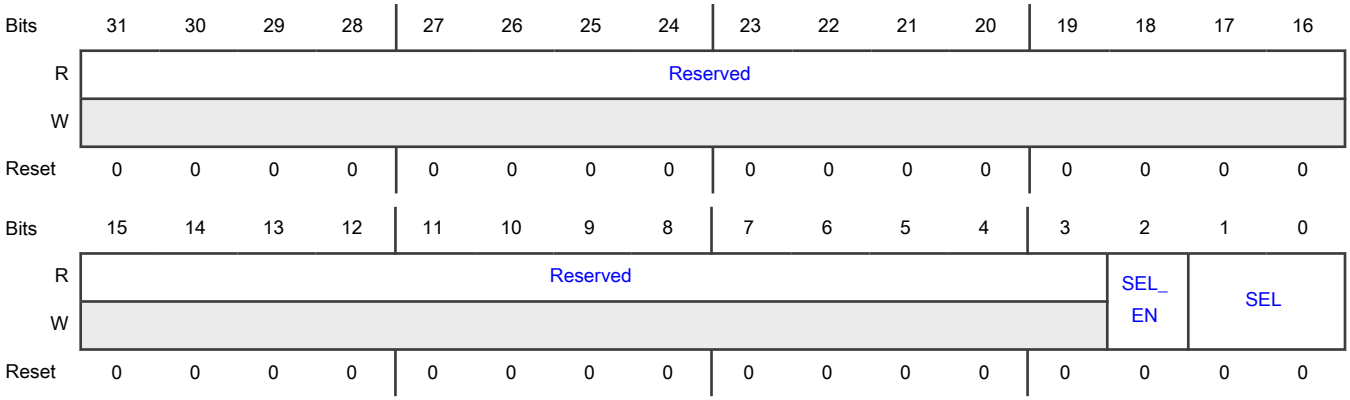
28.7.1.61 LP_FLEXCOMM8 Clock Source Select (FC8FCLKSEL)

Offset

Register	Offset
FC8FCLKSEL	908h

Function
Selects the clock source for LP_FLEXCOMM8.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM8 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM8 clock. 1b - Enables mux output for LP_FLEXCOMM8 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

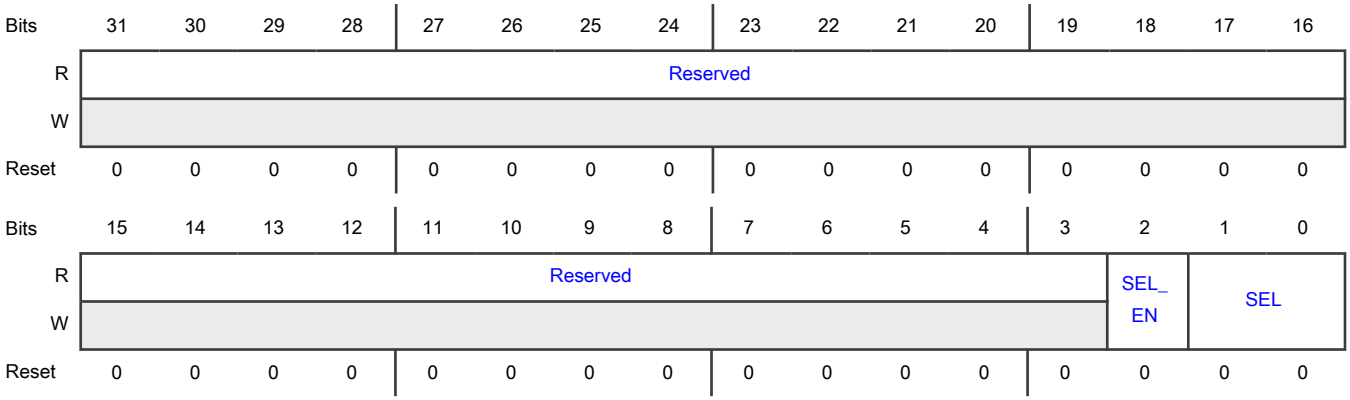
28.7.1.62 LP_FLEXCOMM9 Clock Source Select (FC9FCLKSEL)

Offset

Register	Offset
FC9FCLKSEL	928h

Function
Selects the clock source for LP_FLEXCOMM9.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM9 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM9 clock. 1b - Enables mux output for LP_FLEXCOMM9 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

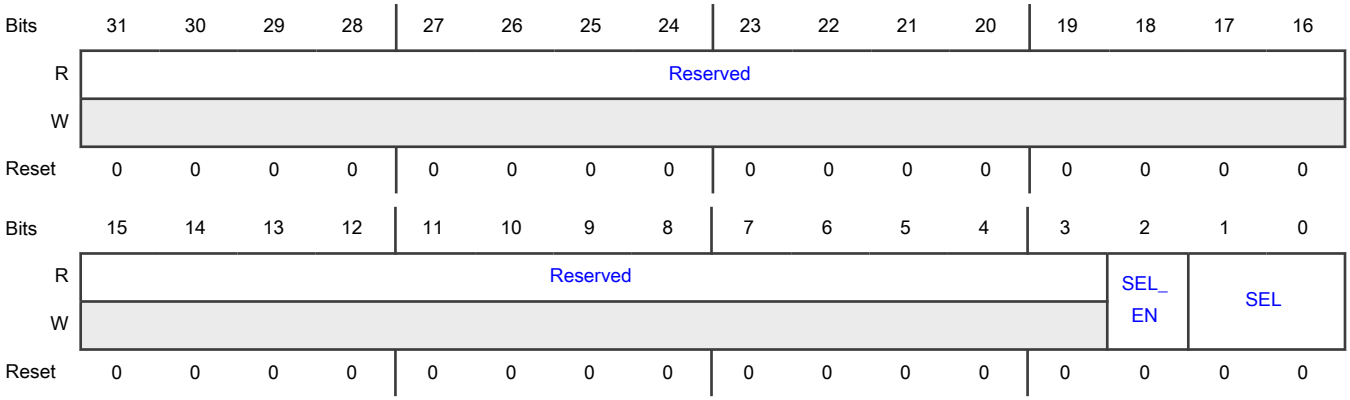
28.7.1.63 LP_FLEXCOMM10 Clock Source Select (FC10FCLKSEL)

Offset

Register	Offset
FC10FCLKSEL	948h

Function
Selects the clock source for LP_FLEXCOMM10.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM10 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM10 clock. 1b - Enables mux output for LP_FLEXCOMM10 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

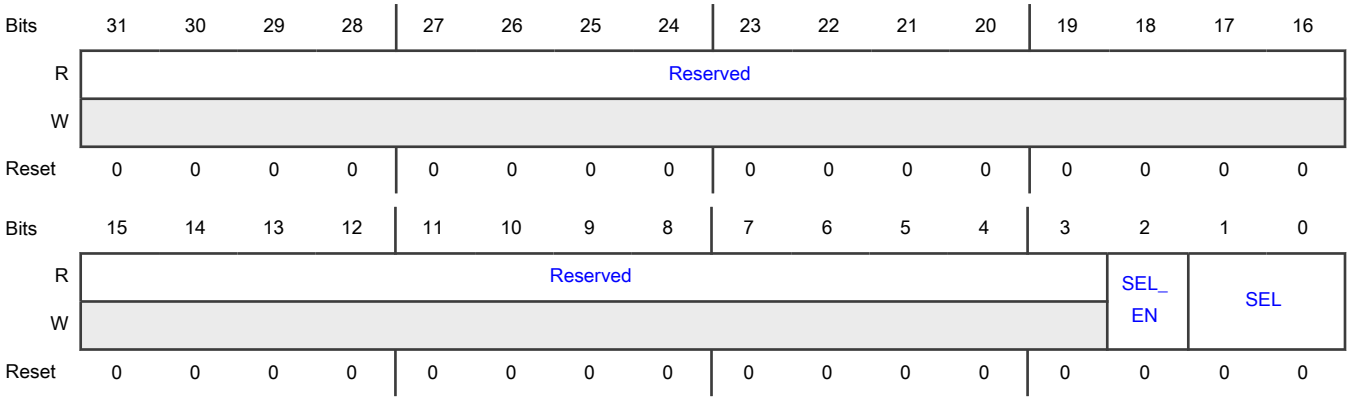
28.7.1.64 LP_FLEXCOMM11 Clock Source Select (FC11FCLKSEL)

Offset

Register	Offset
FC11FCLKSEL	968h

Function
Selects the clock source for LP_FLEXCOMM11.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM11 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM11 clock. 1b - Enables mux output for LP_FLEXCOMM11 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

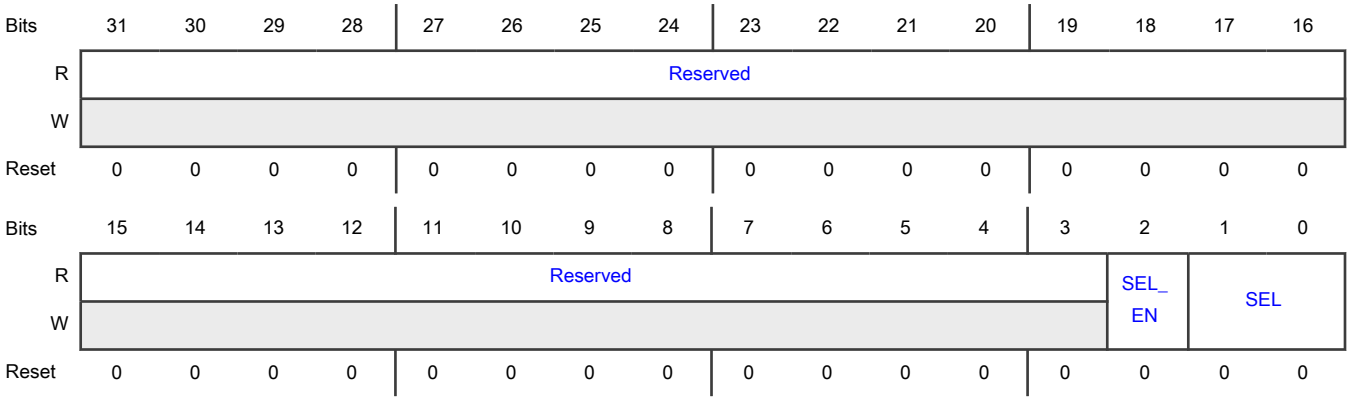
28.7.1.65 LP_FLEXCOMM12 Clock Source Select (FC12FCLKSEL)

Offset

Register	Offset
FC12FCLKSEL	988h

Function
Selects the clock source for LP_FLEXCOMM12.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM12 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM12 clock. 1b - Enables mux output for LP_FLEXCOMM12 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

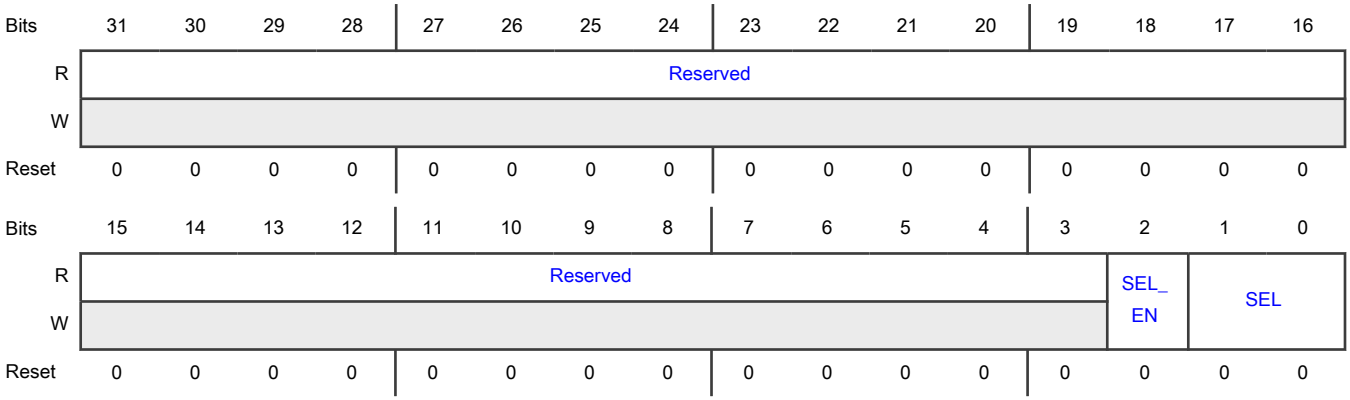
28.7.1.66 LP_FLEXCOMM13 Clock Source Select (FC13FCLKSEL)

Offset

Register	Offset
FC13FCLKSEL	9A8h

Function
Selects the clock source for LP_FLEXCOMM13.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for LP_FLEXCOMM13 clock to reduce power. 0b - Gates mux output for LP_FLEXCOMM13 clock. 1b - Enables mux output for LP_FLEXCOMM13 clock.
1-0 SEL	LP_FLEXCOMM Functional Clock Source Select Selects the clock source for LP_FLEXCOMM functional clock. 00b - fcclk0 01b - fcclk1 10b - fcclk2 11b - fcclk3

28.7.1.67 SAI0, SAI1 and SAI2 Clock Source Select (SAI012FCLKSEL)

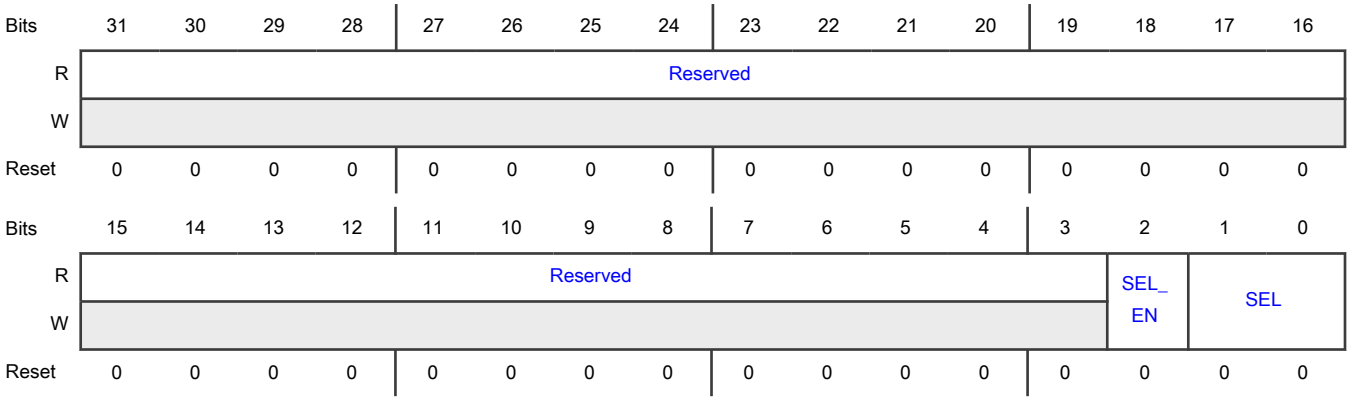
Offset

Register	Offset
SAI012FCLKSEL	9C8h

Function

Selects the clock source for SAI0, SAI1 and SAI2.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for SAI0, SAI1 and SAI2 clocks to reduce power. 0b - Gates mux output for SAI0, SAI1 and SAI2 clocks. 1b - Enables mux output for SAI0, SAI1 and SAI2 clocks.
1-0 SEL	SAI0, SAI1 and SAI2 Clock Source Select Selects the clock source for SAI0, SAI1 and SAI2. 00b - baseclk_cmpt 01b - fro2_div8 10b - fro0_max 11b - audio_clk_cmpt

28.7.1.68 SAI0, SAI1 and SAI2 Functional Clock Divider (SAI012CLKDIV)

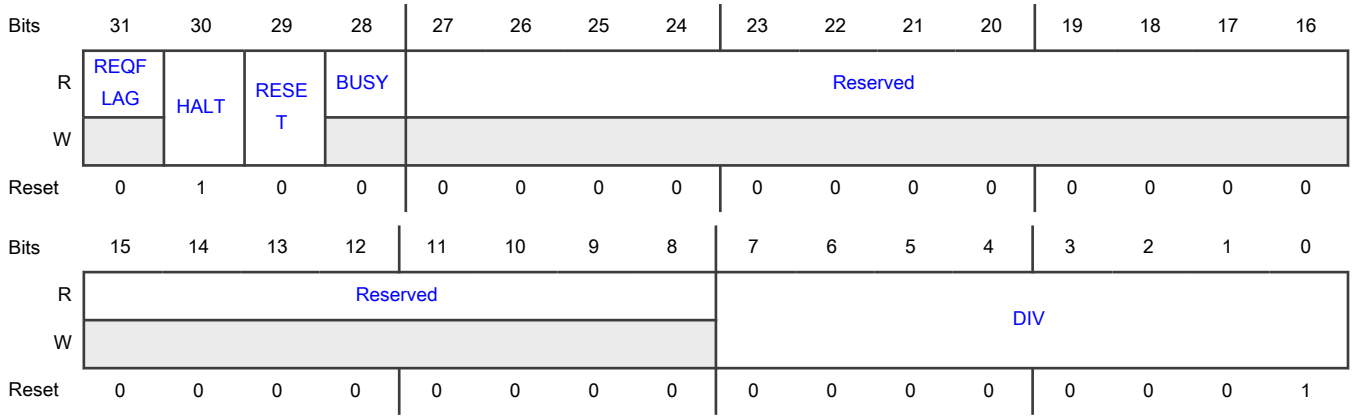
Offset

Register	Offset
SAI012CLKDIV	9CCh

Function

Controls the divider for the SAI0, SAI1 and SAI2s' functional clock sai012_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Reset the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

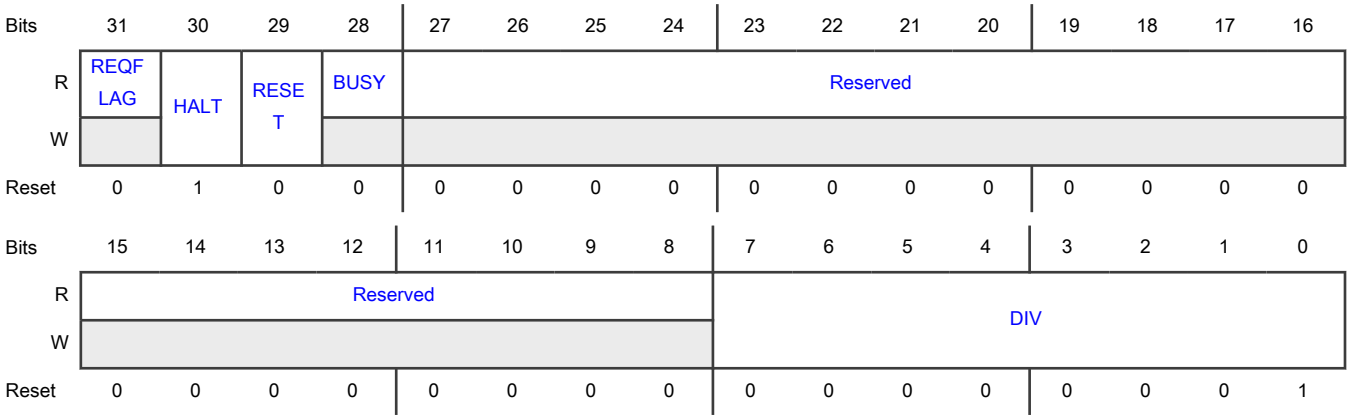
28.7.1.69 CTIMER0 Functional Clock Divider (CTIMER0CLKDIV)

Offset

Register	Offset
CTIMER0CLKDIV	A00h

Function
Controls the divider for the CTIMER0 functional clock ctimer0_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.70 CTIMER1 Functional Clock Divider (CTIMER1CLKDIV)

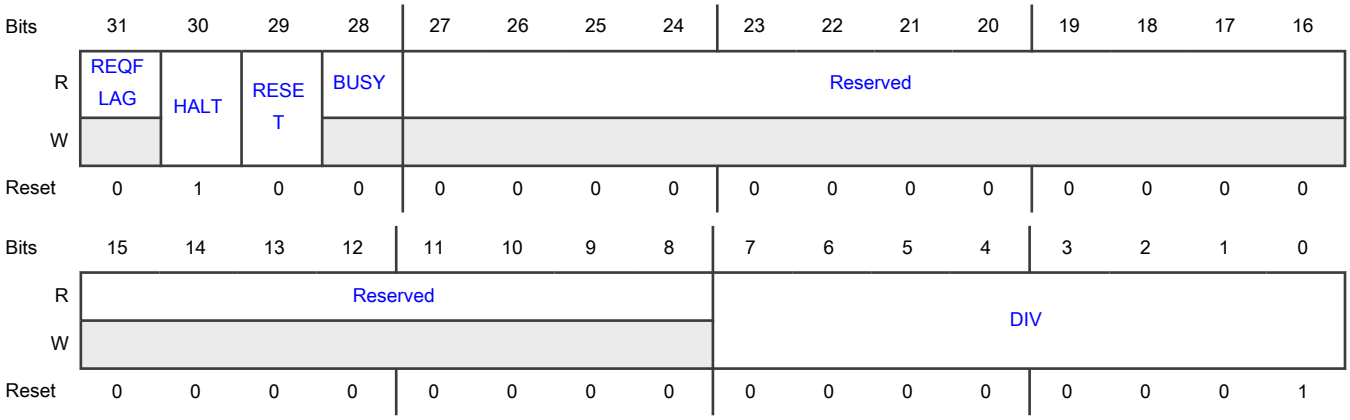
Offset

Register	Offset
CTIMER1CLKDIV	A04h

Function

Controls the divider for the CTIMER1 functional clock ctimer1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.71 CTIMER2 Functional Clock Divider (CTIMER2CLKDIV)

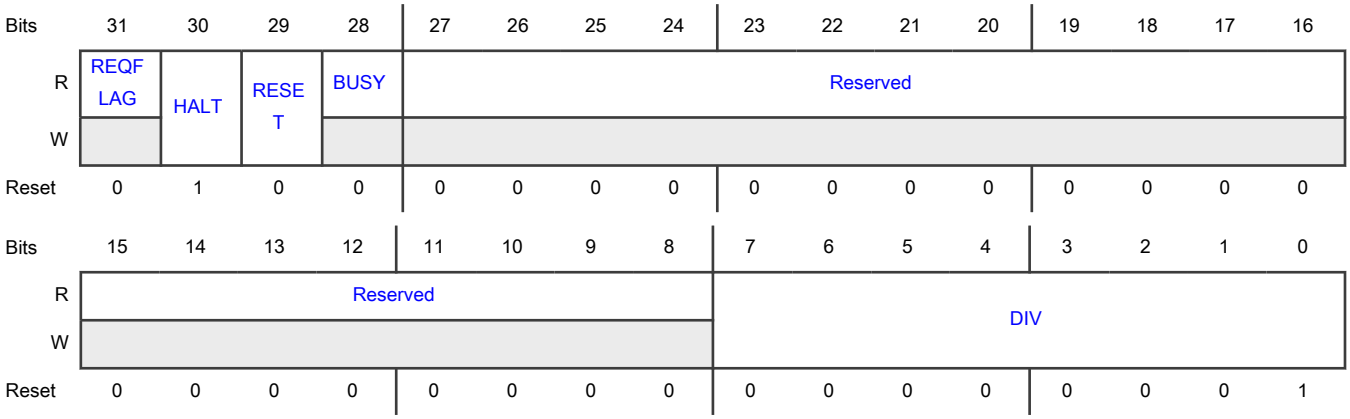
Offset

Register	Offset
CTIMER2CLKDIV	A08h

Function

Controls the divider for the CTIMER2 functional clock ctimer1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.72 CTIMER3 Functional Clock Divider (CTIMER3CLKDIV)

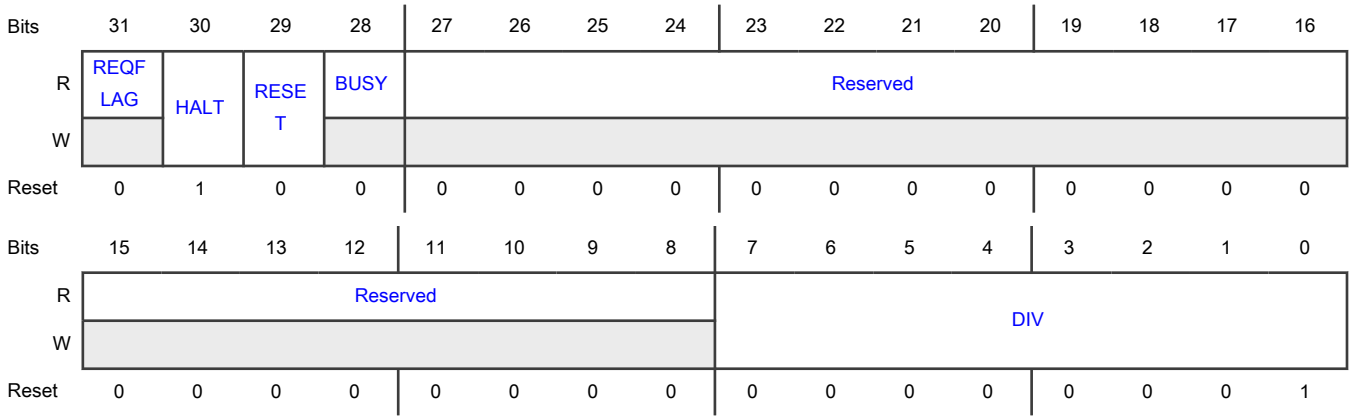
Offset

Register	Offset
CTIMER3CLKDIV	A0Ch

Function

Controls the divider for the CTIMER3 functional clock ctimer1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.73 CTIMER4 Functional Clock Divider (CTIMER4CLKDIV)

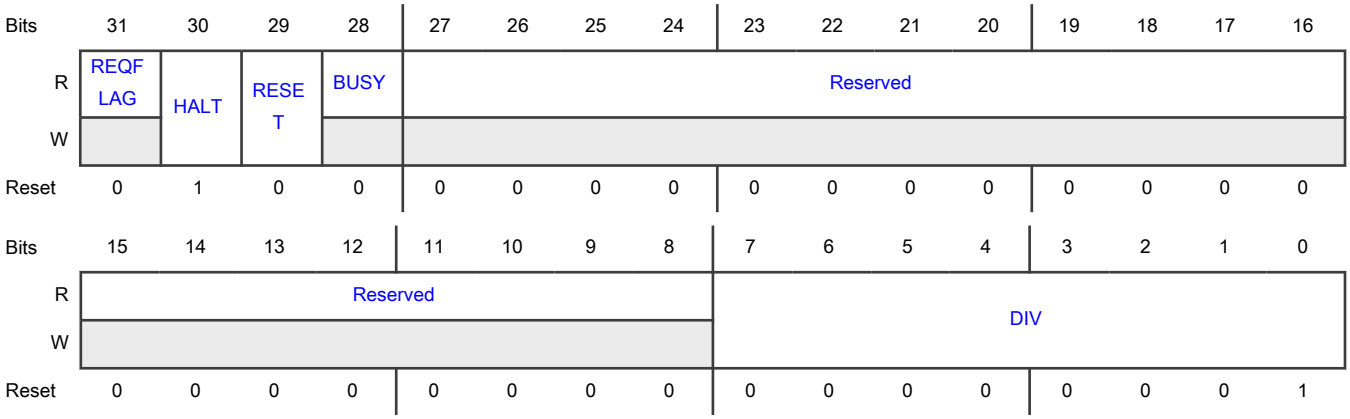
Offset

Register	Offset
CTIMER4CLKDIV	A10h

Function

Controls the divider for the CTIMER4 functional clock ctimer1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.74 CTIMER0 Functional Clock Source Select (CTIMER0FCLKSEL)

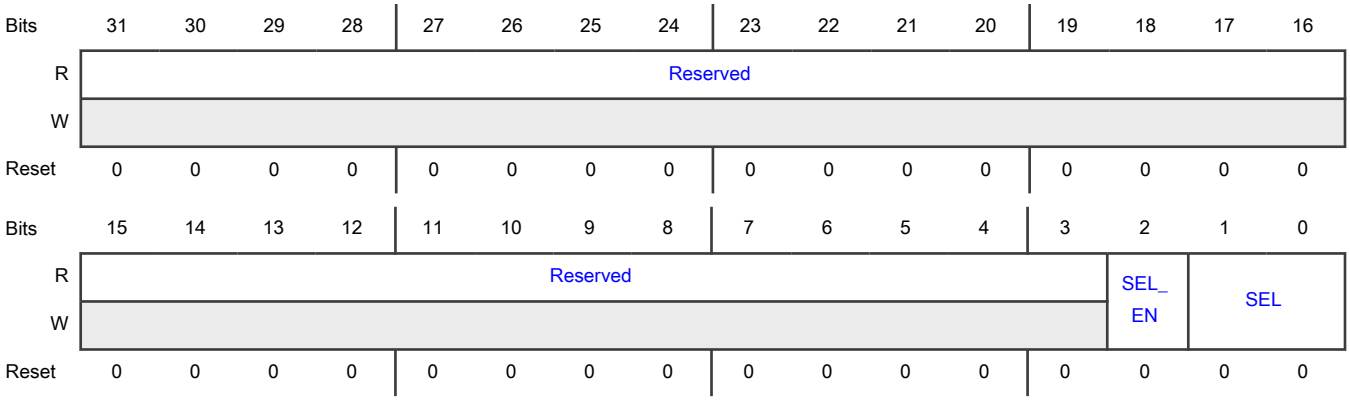
Offset

Register	Offset
CTIMER0FCLKSEL	AA0h

Function

Selects the clock source for CTIMER0.

Diagram



Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 SEL_EN	Clock Mux Output Enable Enables mux output for CTIMER0 clock to reduce power. 0b - Gates mux output for CTIMER0 clock. 1b - Enables mux output for CTIMER0 clock.
1-0 SEL	CTIMER0 Functional Clock Source Select Selects the clock source for CTIMER0. 00b - baseclk_cmpt 01b - audio_clk_cmpt 10b - fro0_max 11b - wake32k_clk

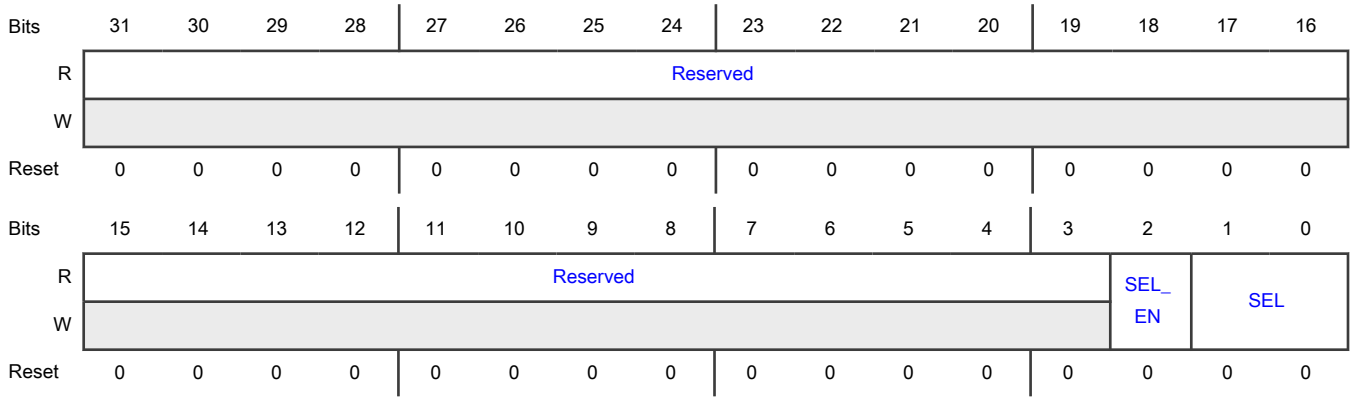
28.7.1.75 CTIMER1 Functional Clock Source Select (CTIMER1FCLKSEL)

Offset

Register	Offset
CTIMER1FCLKSEL	AA4h

Function
Selects the clock source for CTIMER1.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for CTIMER1 clock to reduce power. 0b - Gates mux output for CTIMER1 clock. 1b - Enables mux output for CTIMER1 clock.
1-0 SEL	CTIMER1 Functional Clock Source Select Selects the clock source for CTIMER1. 00b - baseclk_cmpt 01b - audio_clk_cmpt 10b - fro0_max 11b - wake32k_clk

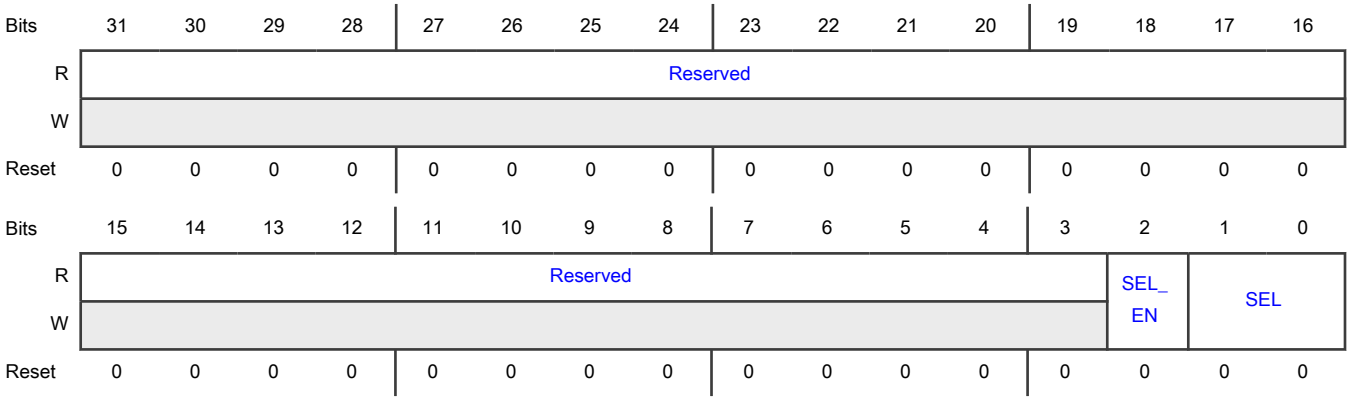
28.7.1.76 CTIMER2 Functional Clock Source Select (CTIMER2FCLKSEL)

Offset

Register	Offset
CTIMER2FCLKSEL	AA8h

Function
Selects the clock source for CTIMER2.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for CTIMER2 clock to reduce power. 0b - Gates mux output for CTIMER2 clock. 1b - Enables mux output for CTIMER2 clock.
1-0 SEL	CTIMER2 Functional Clock Source Select Selects the clock source for CTIMER2. 00b - baseclk_cmpt 01b - audio_clk_cmpt 10b - fro0_max 11b - wake32k_clk

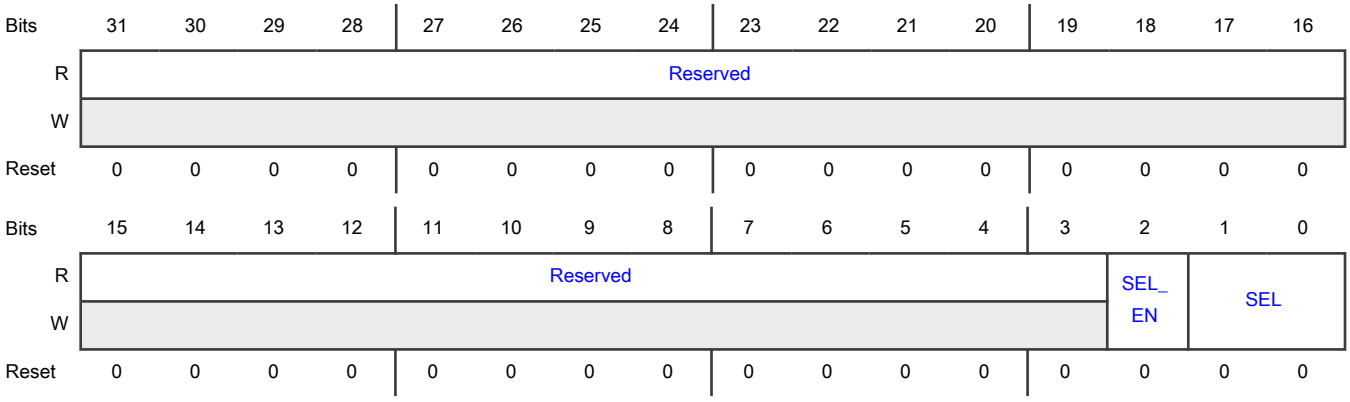
28.7.1.77 CTIMER3 Functional Clock Source Select (CTIMER3FCLKSEL)

Offset

Register	Offset
CTIMER3FCLKSEL	AACh

Function
Selects the clock source for CTIMER3.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for CTIMER3 clock to reduce power. 0b - Gates mux output for CTIMER3 clock. 1b - Enables mux output for CTIMER3 clock.
1-0 SEL	CTIMER3 Functional Clock Source Select Selects the clock source for CTIMER3. 00b - baseclk_cmpt 01b - audio_clk_cmpt 10b - fro0_max 11b - wake32k_clk

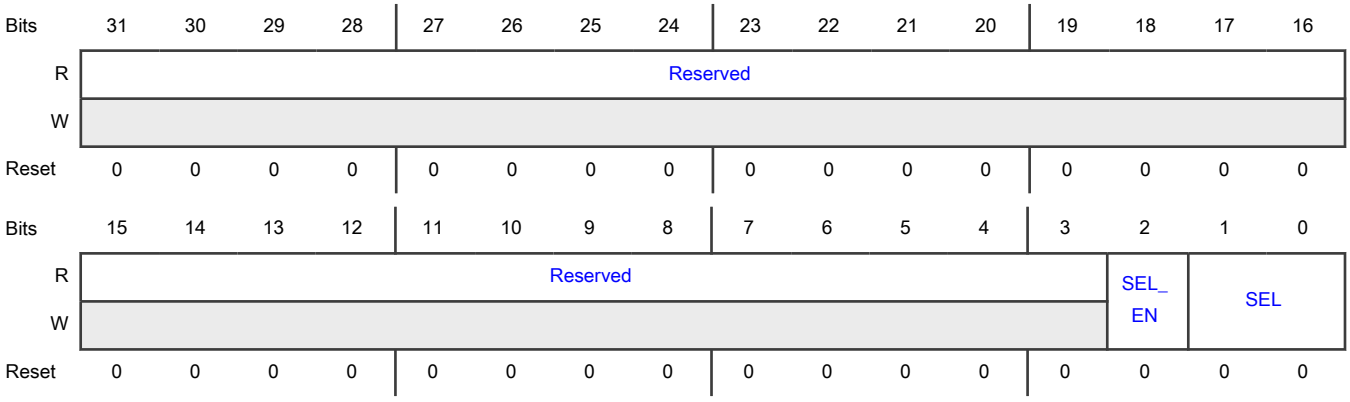
28.7.1.78 CTIMER4 Functional Clock Source Select (CTIMER4FCLKSEL)

Offset

Register	Offset
CTIMER4FCLKSEL	AB0h

Function
Selects the clock source for CTIMER4.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for CTIMER4 clock to reduce power. 0b - Gates mux output for CTIMER4 clock. 1b - Enables mux output for CTIMER4 clock.
1-0 SEL	CTIMER4 Functional Clock Source Select Selects the clock source for CTIMER4. 00b - baseclk_cmpt 01b - audio_clk_cmpt 10b - fro0_max 11b - wake32k_clk

28.7.1.79 I3C0 and I3C1 Functional Clock Source Select (I3C01FCLKSEL)

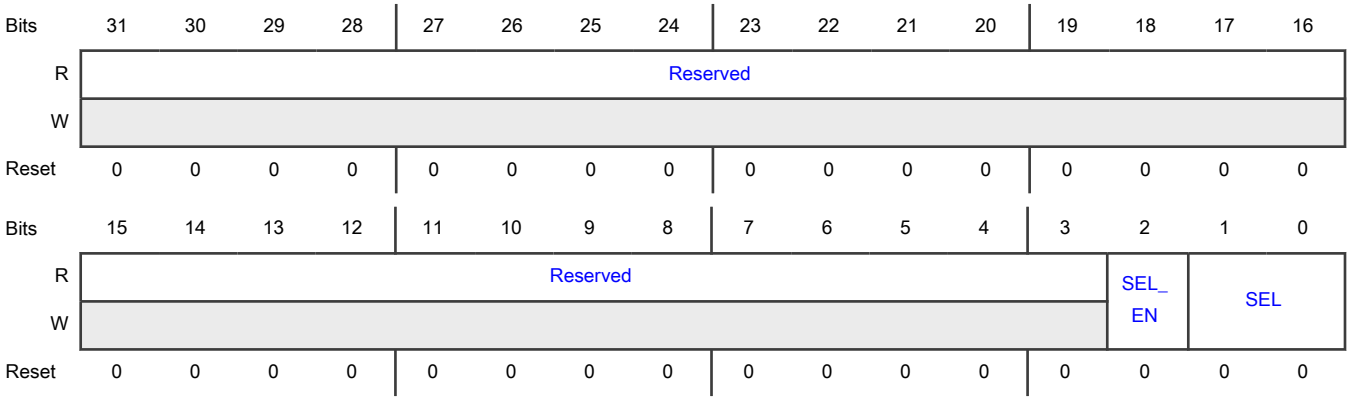
Offset

Register	Offset
I3C01FCLKSEL	B00h

Function

Selects the clock source for the I3C0 and I3C1 functional clocks.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables mux output for I3C0 and I3C1 clocks to reduce power. 0b - Gates mux output for I3C0 and I3C1 clocks. 1b - Enables mux output for I3C0 and I3C1 clocks.
1-0 SEL	I3C0 and I3C1 Functional Clock Source Select Selects the clock source for the I3C0 and I3C1 functional clocks. 00b - baseclk_cmpt 01b - fro0_max 10b - fro1_div8 11b - osc_clk

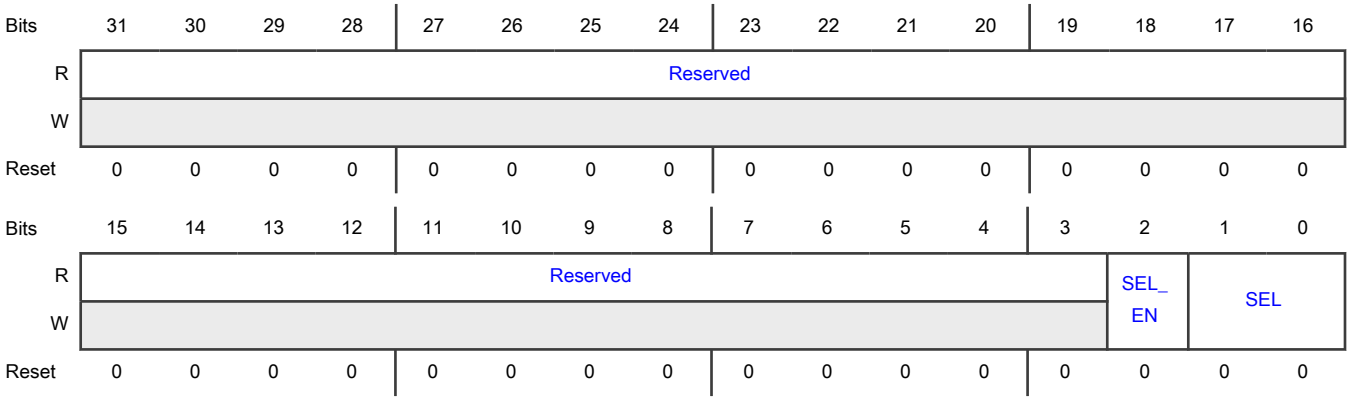
28.7.1.80 I3C0 and I3C1 P-CLK Source Select (I3C01PCLKSEL)

Offset

Register	Offset
I3C01PCLKSEL	B04h

Function
Selects the clock source for the I3C0 and I3C1 P-CLK.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for I3C0 and I3C1 P-CLK to reduce power. 0b - Gates mux output for I3C0 and I3C1 P-CLK. 1b - Enables mux output for I3C0 and I3C1 P-CLK.
1-0 SEL	I3C0 and I3C1 P-CLK Source Select Selects the clock source for the I3C0 and I3C1 P-CLK. 00b - baseclk_cmpt 01b - main_pll_pfd0 10b - fro0_max 11b - fro1_max

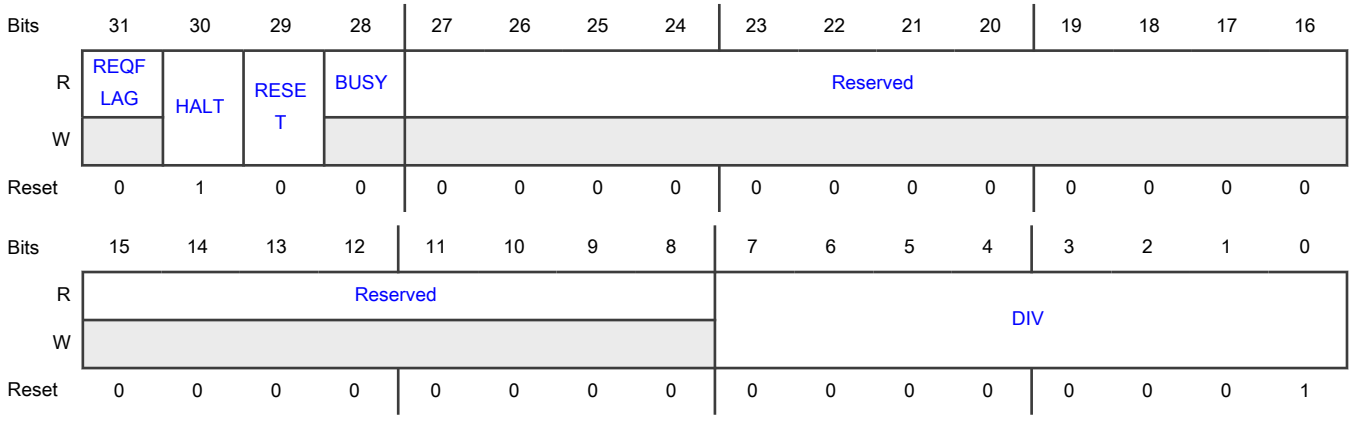
28.7.1.81 I3C0 and I3C1 P-CLK Divider (I3C01PCLKDIV)

Offset

Register	Offset
I3C01PCLKDIV	B08h

Function
Controls the divider for the I3C0 and I3C1 P-CLK i3c01_pclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. Clock being divided must be running for this status to change. 0b - The Divider change has been finished. 1b - The change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. The intent is to allow the divider's clock source to be changed, without causing a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	I3C0, I3C1 Clock Divider Value Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

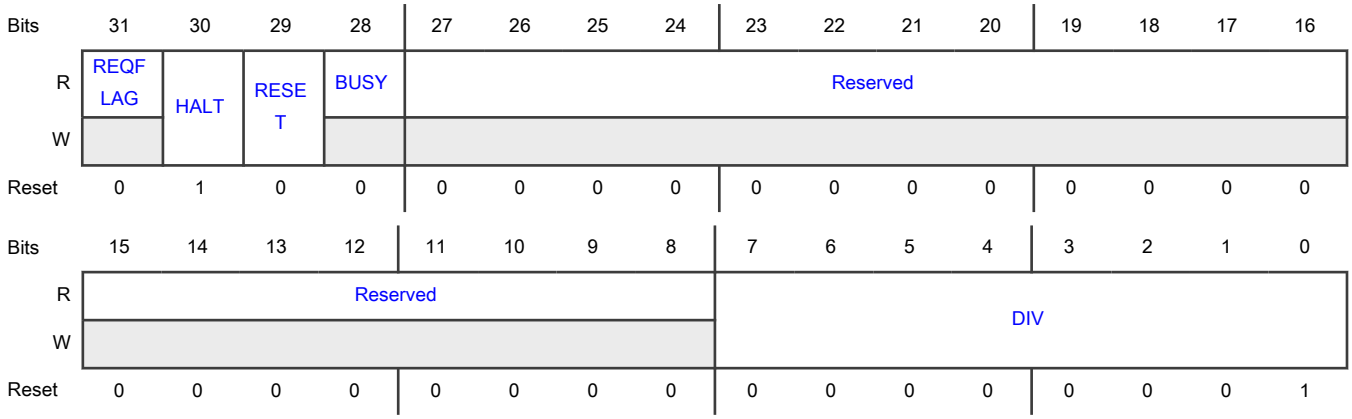
28.7.1.82 I3C0 and I3C1 Functional Clock Divider (I3C01FCLKDIV)

Offset

Register	Offset
I3C01FCLKDIV	B10h

Function
Controls the divider for the I3C0 and I3C1 function clock i3c01_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change has been finished. 1b - The change is being made to the divider counter.
30 HALT	Halt Divider Counter Stops the divider counter. The intent is to allow the divider's clock source to be changed, without causing a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	I3C0 and I3C1 Clock Divider Value Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

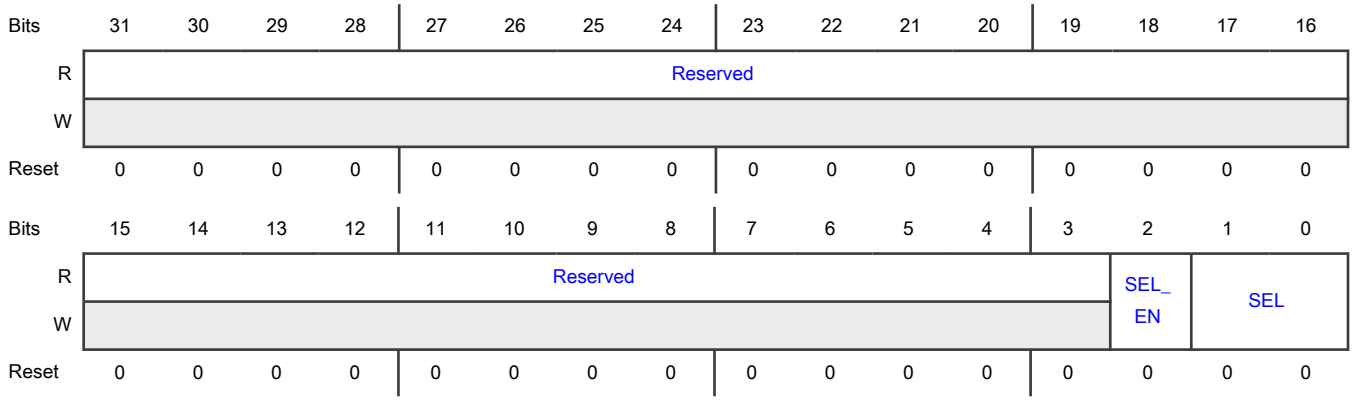
28.7.1.83 CLKOUT_VDD2 Clock Select Source (CLKOUTCLKSEL)

Offset

Register	Offset
CLKOUTCLKSEL	B20h

Function
Selects the clock source for CLKOUT_VDD2.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables mux output for CLKOUT_VDD2 clock to reduce power. 0b - Gates mux output for CLKOUT_VDD2 clock. 1b - Enables mux output for CLKOUT_VDD2 clock.
1-0 SEL	Select Clock Source 00b - baseclk_com2 01b - main_pll_pfd0 10b - fro0_max 11b - fro1_max

28.7.1.84 CLKOUT_VDD2 Clock Divider (CLKOUTCLKDIV)

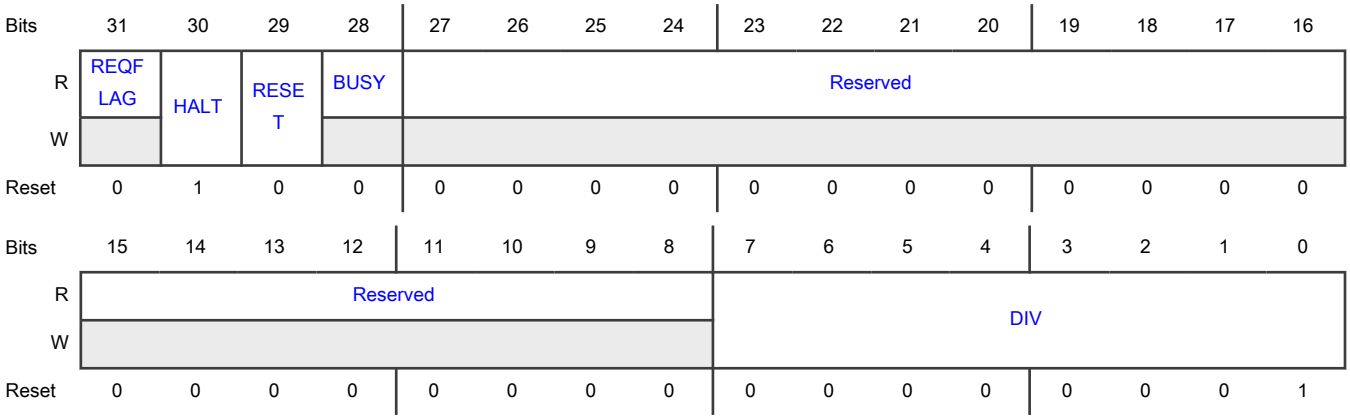
Offset

Register	Offset
CLKOUTCLKDIV	B24h

Function

Controls the divider for the CLKOUT_VDD2 clock comm2_clkout.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Halt Divider Counter Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Reset the Divider Counter Resets the divider counter to use a new divider value immediately, instead of completing the previous count. 0b - No effect 1b - Reset the Divider Counter
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT_VDD2 is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.1.85 VDD2_COMP Audio Clock Source Select (AUDIOVDD2CLKSEL)

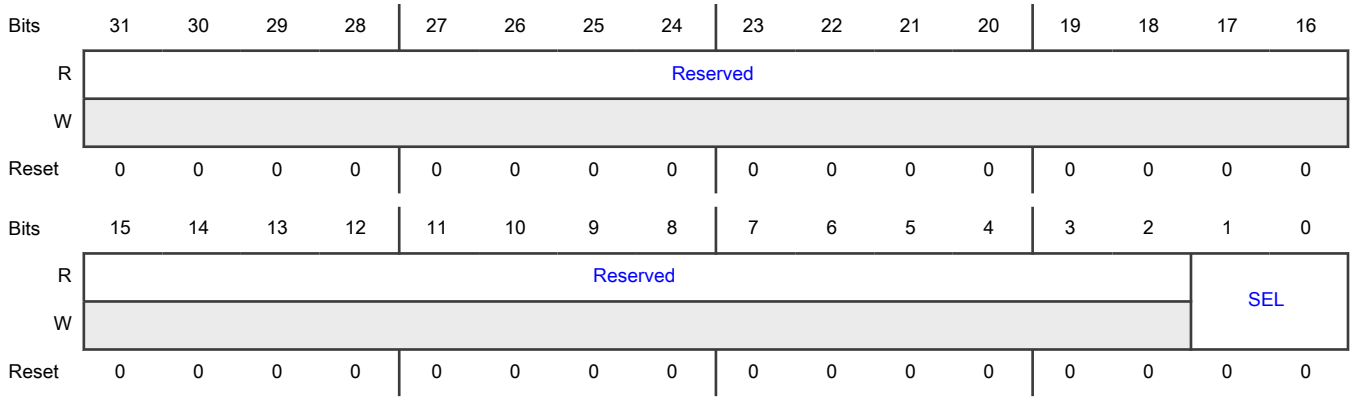
Offset

Register	Offset
AUDIOVDD2CLKSEL	B30h

Function

Selects the clock source for VDD2_COMP audio clock.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Clock Source Select Selects the clock source for VDD2_COMP audio clock. 00b - mclk_in

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - osc_clk 10b - fro2_div8 11b - audio_pll_pfd3

28.7.2 CLKCTL1 register descriptions

This section includes the memory map and register descriptions for Clock Controller 1.

28.7.2.1 CLKCTL1 memory map

CLKCTL1 base address: 4004_1000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0)	32	RW	0000_00C0h
14h	VDD1_SENSE Peripheral Clock Control 1 (PSCCTL1)	32	RW	0000_0000h
40h	VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_SET)	32	RW	0000_0000h
44h	VDD1_SENSE Peripheral Clock Control 1 Set (PSCCTL1_SET)	32	RW	0000_0000h
70h	VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_CLR)	32	RW	0000_0000h
74h	VDD1_SENSE Peripheral Clock Control 1 Clear (PSCCTL1_CLR)	32	RW	0000_0000h
438h	VDD1_SENSE Base Clock Select Source (SENSEBASECLKSEL)	32	RW	0000_0000h
440h	CPU Clock of DSP in VDD1_SENSE Clock Divider (SENSEDSPCPUCLKDIV)	32	RW	0000_0001h
444h	CPU Clock of DSP in VDD1_SENSE Clock Select (SENSEDSPCPUCLKSEL)	32	RW	0000_0000h
500h	SAI3 Clock Select Source (SAI3FCLKSEL)	32	RW	0000_0000h
504h	SAI3 Functional Clock Divider (SAI3CLKDIV)	32	RW	4000_0001h
700h	UTICK1 Functional Clock Source Select (UTICK1FCLKSEL)	32	RW	0000_0000h
704h	UTICK1 Functional Clock Divider (UTICK1CLKDIV)	32	RW	0000_0001h
720h	WWDT2 Functional Clock Source Select (WWDT2FCLKSEL)	32	RW	0000_0000h
740h	WWDT3 Functional Clock Source Select (WWDT3FCLKSEL)	32	RW	0000_0000h
760h	SYSTICK Functional Clock Select Source (SYSTICKFCLKSEL)	32	RW	0000_0004h
764h	SYSTICK Functional Clock Divider (SYSTICKFCLKDIV)	32	RW	4000_0001h
7A0h	CTIMER5 Functional Clock Source Select (CTIMER5FCLKSEL)	32	RW	0000_0000h
7A4h	CTIMER6 Functional Clock Source Select (CTIMER6FCLKSEL)	32	RW	0000_0000h
7A8h	CTIMER7 Functional Clock Source Select (CTIMER7FCLKSEL)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
7B0h	CTIMER5 Functional Clock Divider (CTIMER5CLKDIV)	32	RW	4000_0001h
7B4h	CTIMER6 Functional Clock Divider (CTIMER6CLKDIV)	32	RW	4000_0001h
7B8h	CTIMER7 Functional Clock Divider (CTIMER7CLKDIV)	32	RW	4000_0001h
800h	I3C2 and I3C3 Functional Clock Source Select (I3C23FCLKSEL)	32	RW	0000_0000h
810h	I3C2 and I3C3 Functional Clock Divider (I3C23FCLKDIV)	32	RW	4000_0001h
A00h	LP_FLEXCOMM17 Clock Source Select (FC17FCLKSEL)	32	RW	0000_0000h
A04h	LP_FLEXCOMM17 Clock Divider (FC17FCLKDIV)	32	RW	0000_0001h
A20h	LP_FLEXCOMM18 Clock Source Select (FC18FCLKSEL)	32	RW	0000_0000h
A24h	LP_FLEXCOMM18 Clock Divider (FC18FCLKDIV)	32	RW	0000_0001h
A40h	LP_FLEXCOMM19 Clock Source Select (FC19FCLKSEL)	32	RW	0000_0000h
A44h	LP_FLEXCOMM19 Clock Divider (FC19FCLKDIV)	32	RW	0000_0001h
A60h	LP_FLEXCOMM20 Clock Source Select (FC20FCLKSEL)	32	RW	0000_0000h
A64h	LP_FLEXCOMM20 Clock Divider (FC20FCLKDIV)	32	RW	0000_0001h
AA0h	VDD1_SENSE Audio Clock Source (AUDIOVDD1CLKSEL)	32	RW	0000_0000h

28.7.2.2 VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0)

Offset

Register	Offset
PSCCTL0	10h

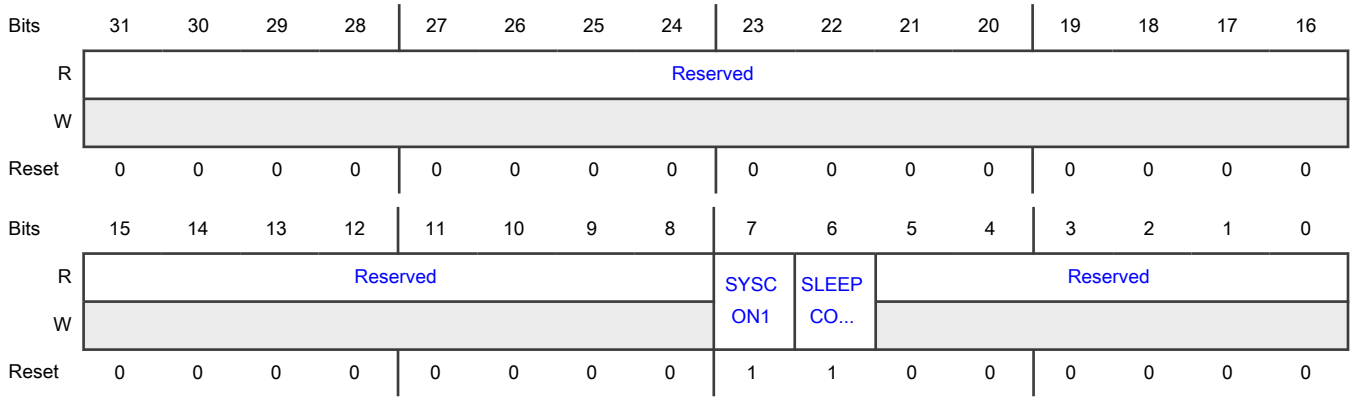
Function

Allows software to control specific peripheral's clock. Writing one to any assigned bit in this register enables the clock and the specified peripheral to operate. Writing zero disables the clock. If VDD1_SENSE power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 SYSCON1	SYSCON1 Clock Enables SYSCON1 clock. 0b - Disable 1b - Enable
6 SLEEPCON1	SLEEPCON1 Clock Enables SLEEPCON1 clock. 0b - Disable 1b - Enable
5-0 —	Reserved

28.7.2.3 VDD1_SENSE Peripheral Clock Control 1 (PSCCTL1)

Offset

Register	Offset
PSCCTL1	14h

Function

Allows software to control specific peripheral's clock. Writing one to any assigned bit in this register enables the clock and the specified peripheral to operate. Writing zero disables the clock. If VDD1_SENSE power rail is OFF, reset after power on.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	INPUTMU...	Reserved	PVTS1	WWDT3	WWDT2	SEMA42_3	MU3	Reserved		UTICK1	MRT1	CTIMER7	CTIMER6	CTIMER5	PINT1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO10	GPIO9	GPIO8	I3C3	I3C2	SAI3	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	eDMA3	eDMA2	Reserved		HiFi1	SENSE_A...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 INPUTMUX1	INPUTMUX1 Clock Enables INPUTMUX1 clock. 0b - Disable 1b - Enable
29 —	Reserved
28 PVTS1	PVTS1 Clock Enables PVTS1 clock. 0b - Disable 1b - Enable
27 WWDT3	WWDT3 clock Enables WWDT3 clock. 0b - Disable 1b - Enable
26 WWDT2	WWDT2 Clock Enables WWDT2 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
25 SEMA42_3	SEMA42_3 Clock Enables SEMA42_3 clock. 0b - Disable 1b - Enable
24 MU3	MU3 Clock Enables MU3 clock. 0b - Disable 1b - Enable
23-22 —	Reserved
21 UTICK1	UTICK1 Clock Enables UTICK1 clock. 0b - Disable 1b - Enable
20 MRT1	MRT1 Clock Enables MRT1 clock. 0b - Disable 1b - Enable
19 CTIMER7	CTIMER7 Clock CTIMER7 clock control 0b - Disable 1b - Enable
18 CTIMER6	CTIMER6 Clock Enables CTIMER6 clock. 0b - Disable 1b - Enable
17 CTIMER5	CTIMER5 Clock Enables CTIMER5 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
16 PINT1	PINT1 Clock Enables PINT1 clock. 0b - Disable 1b - Enable
15 GPIO10	GPIO10 Clock Enables GPIO10 clock. 0b - Disable 1b - Enable
14 GPIO9	GPIO9 Clock Enables GPIO9 clock. 0b - Disable 1b - Enable
13 GPIO8	GPIO8 Clock Enables GPIO8 clock. 0b - Disable 1b - Enable
12 I3C3	I3C3 Clock Enables I3C3 clock. 0b - Disable 1b - Enable
11 I3C2	I3C2 Clock Enables I3C2 clock. 0b - Disable 1b - Enable
10 SAI3	SAI3 Clock Enables SAI3 clock. 0b - Disable 1b - Enable
9	LP_FLEXCOMM20 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM20	Enables LP_FLEXCOMM20 clock. 0b - Disable 1b - Enable
8 LP_FLEXCOMM19	LP_FLEXCOMM19 Clock Enables LP_FLEXCOMM19 clock. 0b - Disable 1b - Enable
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Clock Enables LP_FLEXCOMM18 clock. 0b - Disable 1b - Enable
6 LP_FLEXCOMM17	LP_FLEXCOMM17 Clock Enables LP_FLEXCOMM17 clock. 0b - Disable 1b - Enable
5 eDMA3	eDMA3 Clock Enables eDMA3 clock. 0b - Disable 1b - Enable
4 eDMA2	eDMA2 Clock Enables eDMA2 clock. 0b - Disable 1b - Enable
3-2 —	Reserved
1 HiFi1	HiFi1 Clock Enables HiFi1 clock. 0b - Disable 1b - Enable
0	VDD1_SENSE Access RAM Arbiter0 Clock Enables VDD1_SENSE access RAM arbiter0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
SENSE_ACCESS_RAM_ARBITER0	0b - Disable 1b - Enable

28.7.2.4 VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_SET)

Offset

Register	Offset
PSCCTL0_SET	40h

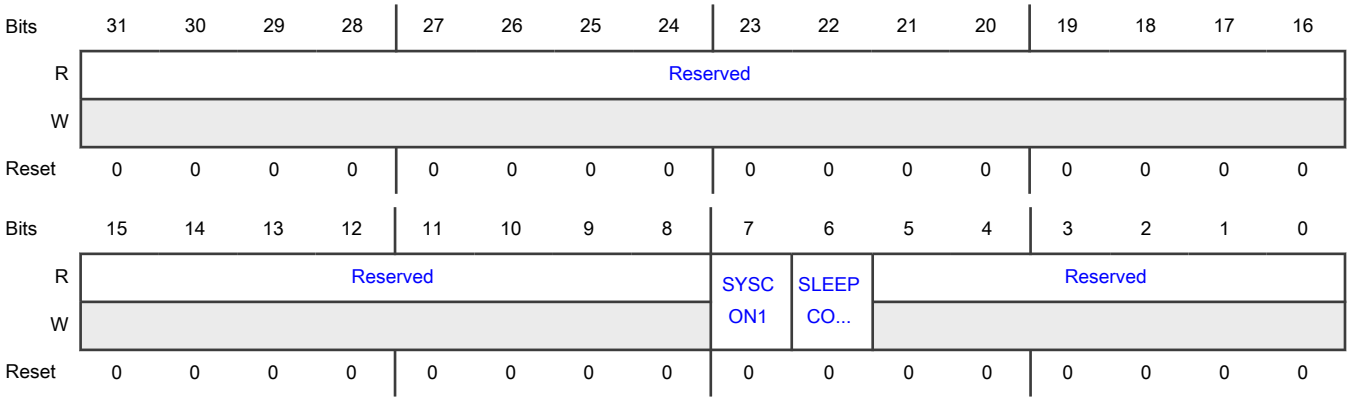
Function

Enables the corresponding position in PSCCTL0 by writing 1 to a bit position in PSCCTL0_SET.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 SYSCON1	SYSCON1 Clock Enables SYSCON1 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
6 SLEEPCON1	SLEEPCON1 Clock Enables SLEEPCON1 clock. 0b - No effect 1b - Enable
5-0 —	Reserved

28.7.2.5 VDD1_SENSE Peripheral Clock Control 1 Set (PSCCTL1_SET)

Offset

Register	Offset
PSCCTL1_SET	44h

Function

Enables the corresponding position in PSCCTL1 by writing 1 to a bit position in PSCCTL1_SET.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	INPUTMU...	Reserved	PVTS1	WWDT3	WWDT2	SEMA42_3	MU3	Reserved		UTICK1	MRT1	CTIMER7	CTIMER6	CTIMER5	PINT1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO10	GPIO9	GPIO8	I3C3	I3C2	SAI3	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	eDMA3	eDMA2	Reserved		HiFi1	SENSE_A...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 INPUTMUX1	INPUTMUX1 Clock Enables INPUTMUX1 clock. 0b - No effect 1b - Enable
29 —	Reserved
28 PVTS1	PVTS1 Clock Enables PVTS1 clock. 0b - No effect 1b - Enable
27 WWDT3	WWDT3 Clock Enables WWDT3 clock. 0b - No effect 1b - Enable
26 WWDT2	WWDT2 Clock Enables WWDT2 clock. 0b - No effect 1b - Enable
25 SEMA42_3	SEMA42_3 Clock Enables SEMA42_3 clock. 0b - No effect 1b - Enable
24 MU3	MU3 Clock Enables MU3 clock. 0b - No effect 1b - Enable
23-22 —	Reserved
21	UTICK1 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
UTICK1	Enables UTICK1 clock. 0b - No effect 1b - Enable
20 MRT1	MRT1 Clock Enables MRT1 clock. 0b - No effect 1b - Enable
19 CTIMER7	CTIMER7 Clock Enables CTIMER7 clock. 0b - No effect 1b - Enable
18 CTIMER6	CTIMER6 Clock Enables CTIMER6 clock. 0b - No effect 1b - Enable
17 CTIMER5	CTIMER5 Clock Enables CTIMER5 clock. 0b - No effect 1b - Enable
16 PINT1	PINT1 Clock Enables PINT1 clock. 0b - No effect 1b - Enable
15 GPIO10	GPIO10 Clock Enables GPIO10 clock. 0b - No effect 1b - Enable
14 GPIO9	GPIO9 Clock Enables GPIO9 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 GPIO8	GPIO8 Clock Enables GPIO8 clock. 0b - No effect 1b - Enable
12 I3C3	I3C3 Clock Enables I3C3 clock. 0b - No effect 1b - Enable
11 I3C2	I3C2 Clock Enables I3C2 clock. 0b - No effect 1b - Enable
10 SAI3	SAI3 Clock Enables SAI3 clock. 0b - No effect 1b - Enable
9 LP_FLEXCOMM20	LP_FLEXCOMM20 Clock Enables or LP_FLEXCOMM20 clock. 0b - No effect 1b - Enable
8 LP_FLEXCOMM19	LP_FLEXCOMM19 Clock Enables LP_FLEXCOMM19 clock. 0b - No effect 1b - Enable
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Clock Enables LP_FLEXCOMM18 clock. 0b - No effect 1b - Enable
6 LP_FLEXCOMM17	LP_FLEXCOMM17 Clock Enables LP_FLEXCOMM17 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
5 eDMA3	eDMA3 Clock Enables eDMA3 clock. 0b - No effect 1b - Enable
4 eDMA2	eDMA2 Clock Enables eDMA2 clock. 0b - No effect 1b - Enable
3-2 —	Reserved
1 HiFi1	HiFi1 Clock Enables HiFi1 clock. 0b - No effect 1b - Enable
0 SENSE_ACCESS_RAM_ARBITER0	VDD1_SENSE Access RAM Arbiter0 Clock Enables VDD1_SENSE access RAM arbiter0 clock. 0b - No effect 1b - Enable

28.7.2.6 VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_CLR)

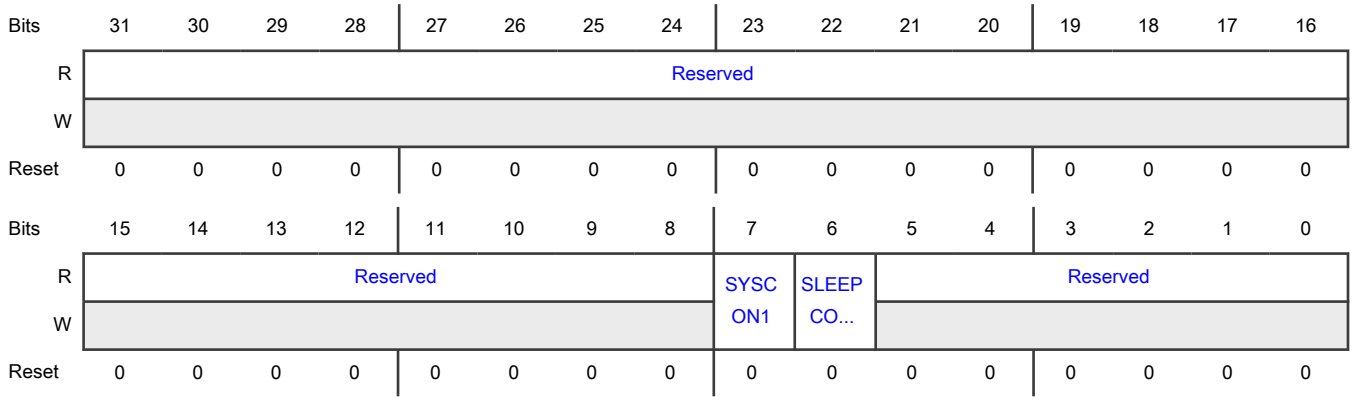
Offset

Register	Offset
PSCCTL0_CLR	70h

Function

Clears the corresponding bit in PSCCTL0 by writing 1 to a bit position in PSCCTL0_CLR.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 SYSCON1	SYSCON1 Clock Disables SYSCON1 clock. 0b - No effect 1b - Disable
6 SLEEPCON1	SLEEPCON1 Clock Disables SLEEPCON1 clock. 0b - No effect 1b - Disable
5-0 —	Reserved

28.7.2.7 VDD1_SENSE Peripheral Clock Control 1 Clear (PSCCTL1_CLR)

Offset

Register	Offset
PSCCTL1_CLR	74h

Function

Clears the corresponding bit in PSCCTL1 by writing 1 to a bit position in PSCCTL1_CLR.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	INPUTMU...	Reserved	PVTS1	WWDT3	WWDT2	SEMA42_3	MU3	Reserved		UTICK1	MRT1	CTIMER7	CTIMER6	CTIMER5	PINT1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO10	GPIO9	GPIO8	I3C3	I3C2	SAI3	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	eDMA3	eDMA2	Reserved		HiFi1	SENSE_A...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 INPUTMUX1	INPUTMUX1 Clock Disables INPUTMUX1 clock. 0b - No effect 1b - Disable
29 —	Reserved
28 PVTS1	PVTS1 Clock Disables PVTS1 clock. 0b - No effect 1b - Disable
27 WWDT3	WWDT3 Clock Disables WWDT3 clock. 0b - No effect 1b - Disable
26 WWDT2	WWDT2 Clock Disables WWDT2 clock. 0b - No effect 1b - Disable
25	SEMA42_3 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEMA42_3	Disables SEMA42_3 clock. 0b - No effect 1b - Disable
24 MU3	MU3 Clock Disables MU3 clock. 0b - No effect 1b - Disable
23-22 —	Reserved
21 UTICK1	UTICK1 Clock Disables UTICK1 clock. 0b - No effect 1b - Disable
20 MRT1	MRT1 Clock Disables MRT1 clock. 0b - No effect 1b - Disable
19 CTIMER7	CTIMER7 Clock Disables CTIMER7 clock. 0b - No effect 1b - Disable
18 CTIMER6	CTIMER6 Clock Disables CTIMER6 clock. 0b - No effect 1b - Disable
17 CTIMER5	CTIMER5 Clock Disables CTIMER5 clock. 0b - No effect 1b - Disable
16 PINT1	PINT1 Clock Disables PINT1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Disable
15 GPIO10	GPIO10 Clock Disables GPIO10 clock. 0b - No effect 1b - Disable
14 GPIO9	GPIO9 Clock Disables GPIO9 clock. 0b - No effect 1b - Disable
13 GPIO8	GPIO8 Clock Disables GPIO8 clock. 0b - No effect 1b - Disable
12 I3C3	I3C3 Clock Disables I3C3 clock. 0b - No effect 1b - Disable
11 I3C2	I3C2 Clock Disables I3C2 clock. 0b - No effect 1b - Disable
10 SAI3	SAI3 Clock Disables SAI3 clock. 0b - No effect 1b - Disable
9 LP_FLEXCOMM20	LP_FLEXCOMM20 Clock Disables LP_FLEXCOMM20 clock. 0b - No effect 1b - Disable
8	LP_FLEXCOMM19 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM19	Disables LP_FLEXCOMM19 clock. 0b - No effect 1b - Disable
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Clock Disables LP_FLEXCOMM18 clock. 0b - No effect 1b - Disable
6 LP_FLEXCOMM17	LP_FLEXCOMM17 Clock Disables LP_FLEXCOMM17 clock. 0b - No effect 1b - Disable
5 eDMA3	eDMA3 Clock Disables eDMA3 clock. 0b - No effect 1b - Disable
4 eDMA2	eDMA2 Clock Disables eDMA2 clock. 0b - No effect 1b - Disable
3-2 —	Reserved
1 HiFi1	HiFi1 Clock Disables HiFi1 clock. 0b - No effect 1b - Disable
0 SENSE_ACCESS_RAM_ARBITER0	VDD1_SENSE Access RAM Arbiter0 Clock Disables VDD1_SENSE access RAM arbiter0 clock. 0b - No effect 1b - Disable

28.7.2.8 VDD1_SENSE Base Clock Select Source (SENSEBASECLKSEL)

Offset

Register	Offset
SENSEBASECLKSEL	438h

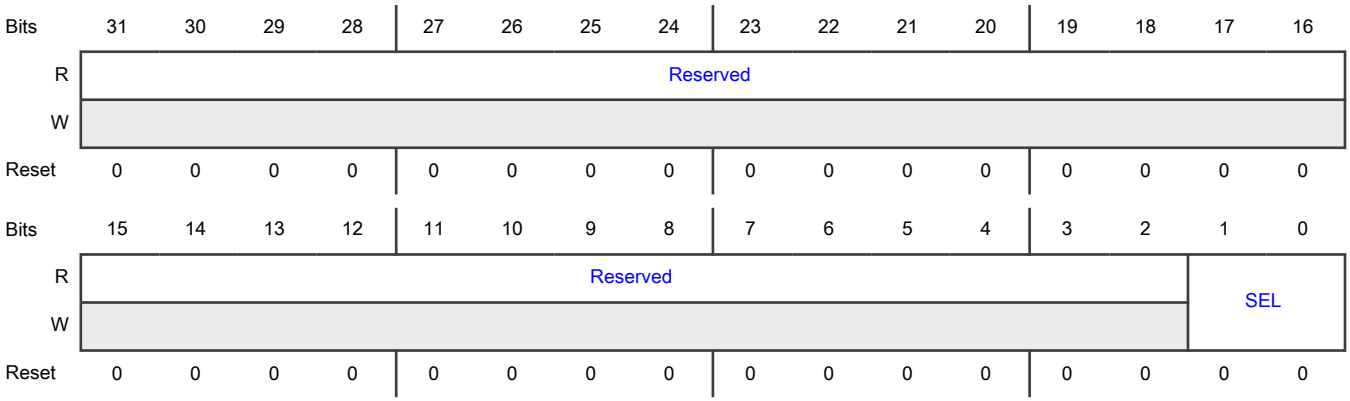
Function

Selects the clock source for VDD1_SENSE base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Clock Source Select for Main VDD1_SENSE Base Clock Selects clock source for main VDD1_SENSE base clock. 00b - fro1_div3 01b - fro1_max 10b - fro2_div3 11b - 1m_lposc

28.7.2.9 CPU Clock of DSP in VDD1_SENSE Clock Divider (SENSEDSPCPUCLKDIV)

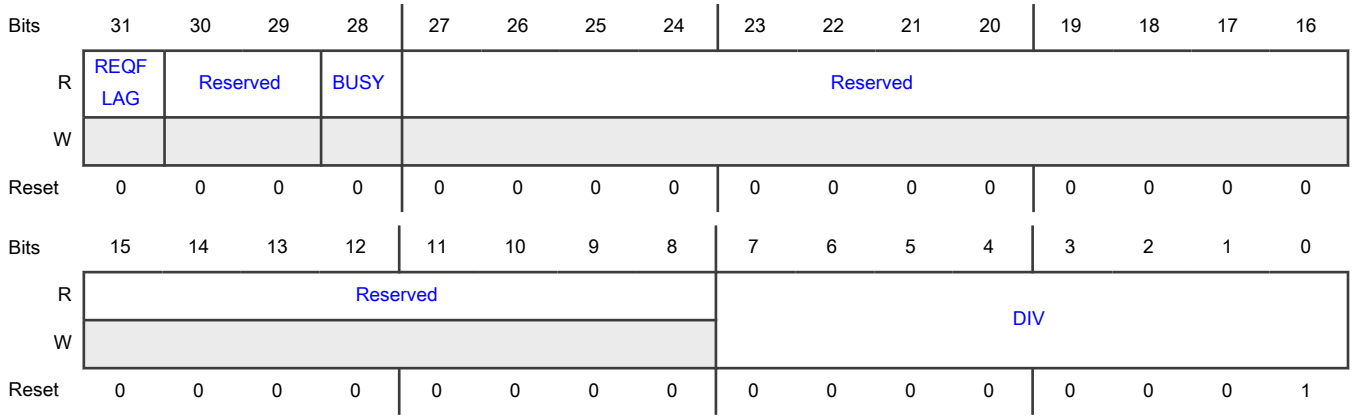
Offset

Register	Offset
SENSEDSPCPUCLKDIV	440h

Function

Controls the divider for the CPU clock of DSP in VDD1_SENSE sense_dsp_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0	Clock Divider Value Select

Table continues on the next page...

Table continued from the previous page...

Field	Function
DIV	Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.2.10 CPU Clock of DSP in VDD1_SENSE Clock Select (SENSEDSPCPUCLKSEL)

Offset

Register	Offset
SENSEDSPCPUCLKSEL	444h

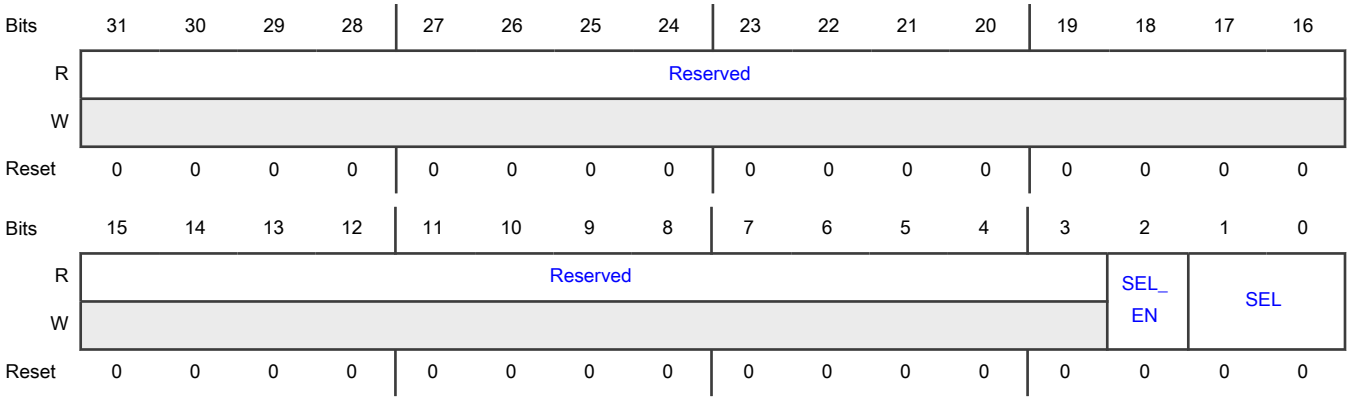
Function

Selects a clock source for CPU of DSP in VDD1_SENSE.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects a clock source for CPU of DSP in VDD1_SENSE. 00b - baseclk_sense 01b - fro2_max 10b - audio_pll_pfd1 11b - fro1_max

28.7.2.11 SAI3 Clock Select Source (SAI3FCLKSEL)

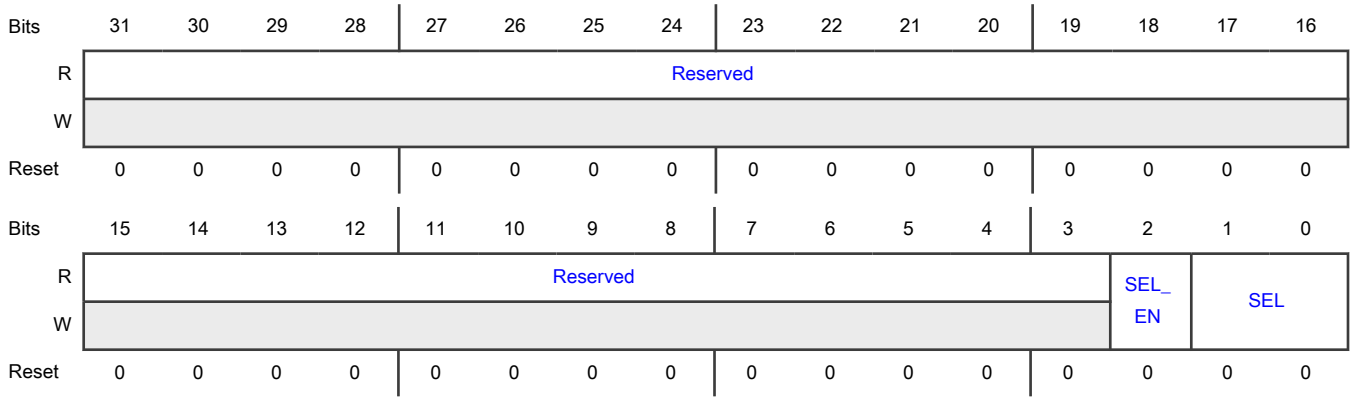
Offset

Register	Offset
SAI3FCLKSEL	500h

Function

Selects a clock source for SAI3 functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects a clock source for SAI3 clock. 00b - baseclk_sense 01b - fro2_div8 10b - fro2_max 11b - audio_clk

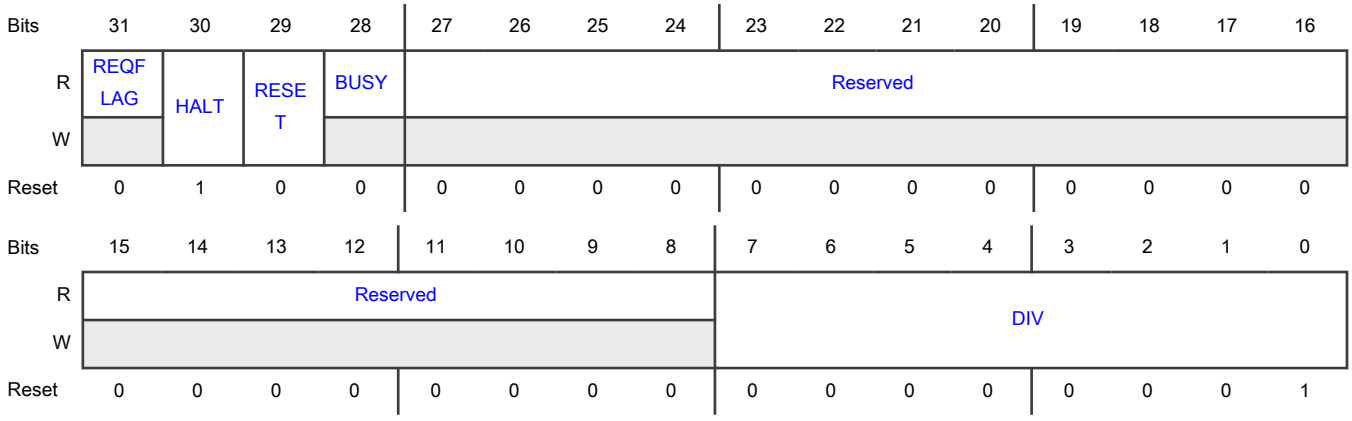
28.7.2.12 SAI3 Functional Clock Divider (SAI3CLKDIV)

Offset

Register	Offset
SAI3CLKDIV	504h

Function
Controls the divider for SAI3 functional clock sai3_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Halts the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when user does not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

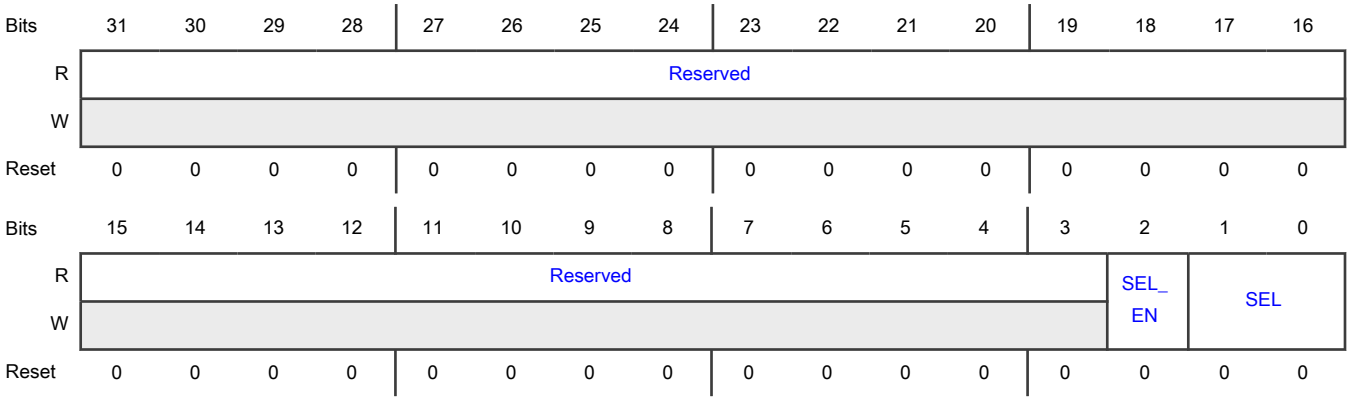
28.7.2.13 UTICK1 Functional Clock Source Select (UTICK1FCLKSEL)

Offset

Register	Offset
UTICK1FCLKSEL	700h

Function
Selects the clock source for UTICK1.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	UTICK Clock Source Select Selects the clock source for UTICK timer. 00b - baseclk_sense 01b - 1m_lposc 10b - fro2_max 11b - fro1_div2

28.7.2.14 UTICK1 Functional Clock Divider (UTICK1CLKDIV)

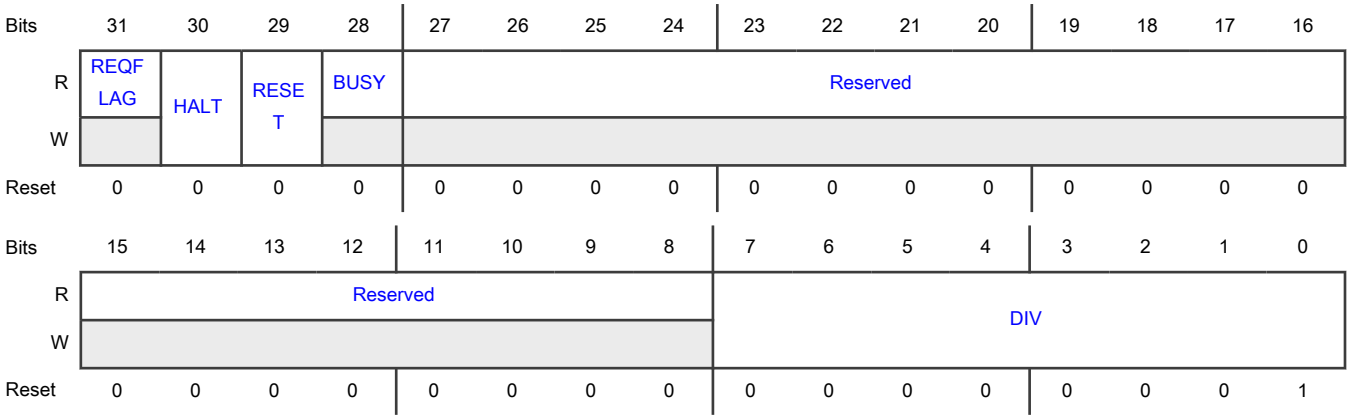
Offset

Register	Offset
UTICK1CLKDIV	704h

Function

Controls the divider for the UTICK1 functional clock utick1_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Select a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.2.15 WWDT2 Functional Clock Source Select (WWDT2FCLKSEL)

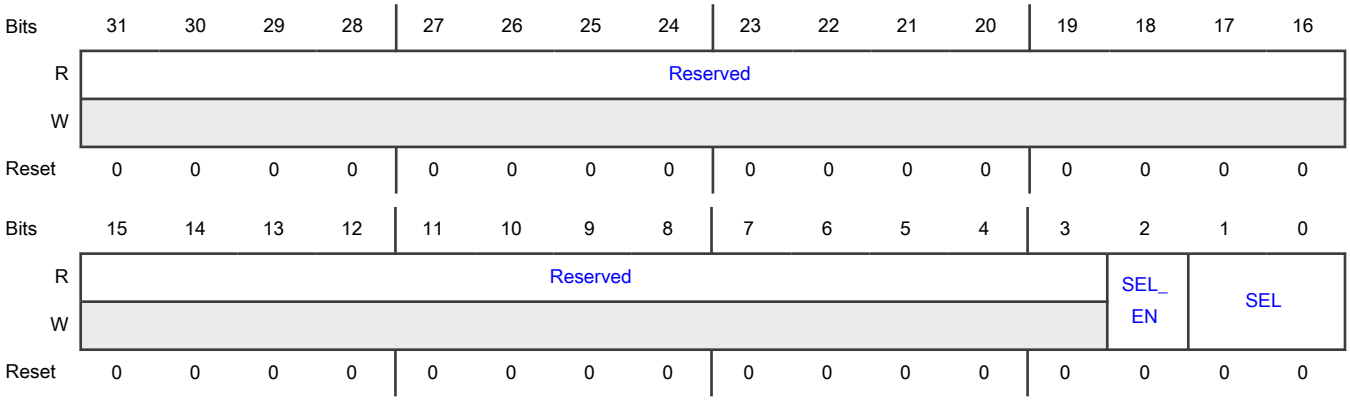
Offset

Register	Offset
WWDT2FCLKSEL	720h

Function

Selects the clock source for WWDT2.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Enable Clock Mux Output Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects clock source for WWDT2. 00b - 1m_lposc 01b - Reserved 10b - Reserved 11b - Reserved

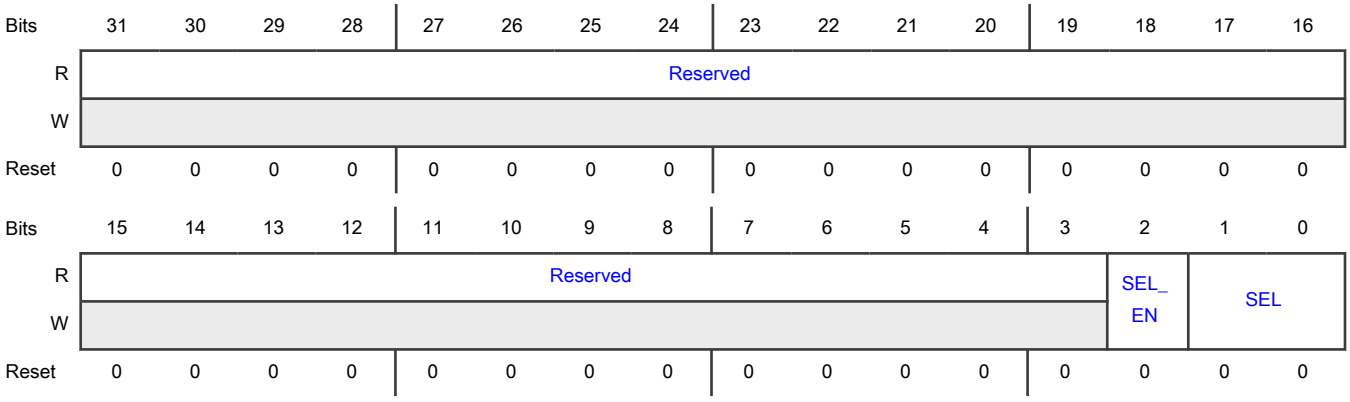
28.7.2.16 WWDT3 Functional Clock Source Select (WWDT3FCLKSEL)

Offset

Register	Offset
WWDT3FCLKSEL	740h

Function
Selects the clock source for WWDT3.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects clock source for WWDT3. 00b - 1m_lposc 01b - Reserved 10b - Reserved 11b - Reserved

28.7.2.17 SYSTICK Functional Clock Select Source (SYSTICKFCLKSEL)

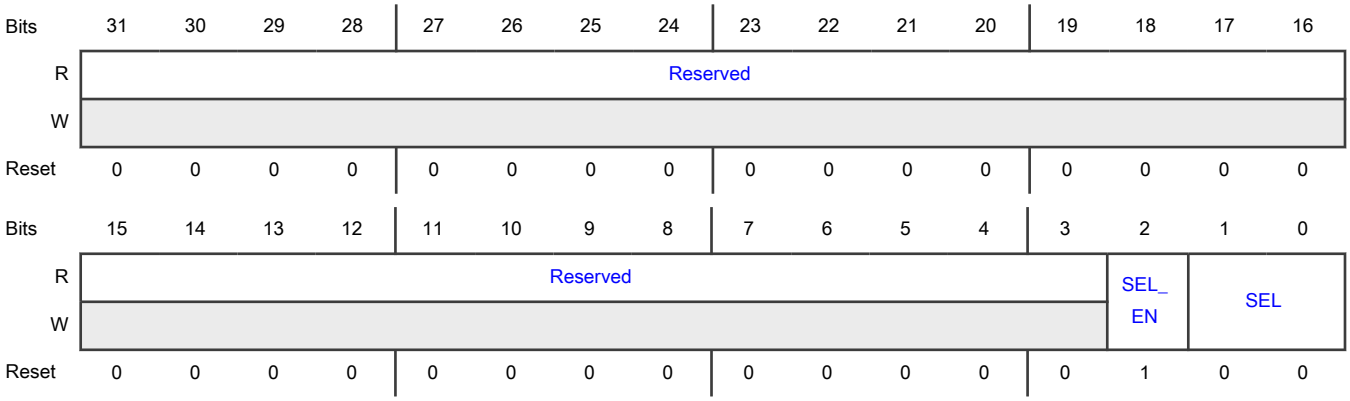
Offset

Register	Offset
SYSTICKFCLKSEL	760h

Function

Selects clock source for Systick timer.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output for Systick timer to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects clock source for Systick timer. 00b - baseclk_sense 01b - 1m_lposc 10b - wake32k_clk (switch to other clock source before VDD1_SENSE enters into the Deep Sleep Retention mode.) 11b - osc_clk

28.7.2.18 SYSTICK Functional Clock Divider (SYSTICKFCLKDIV)

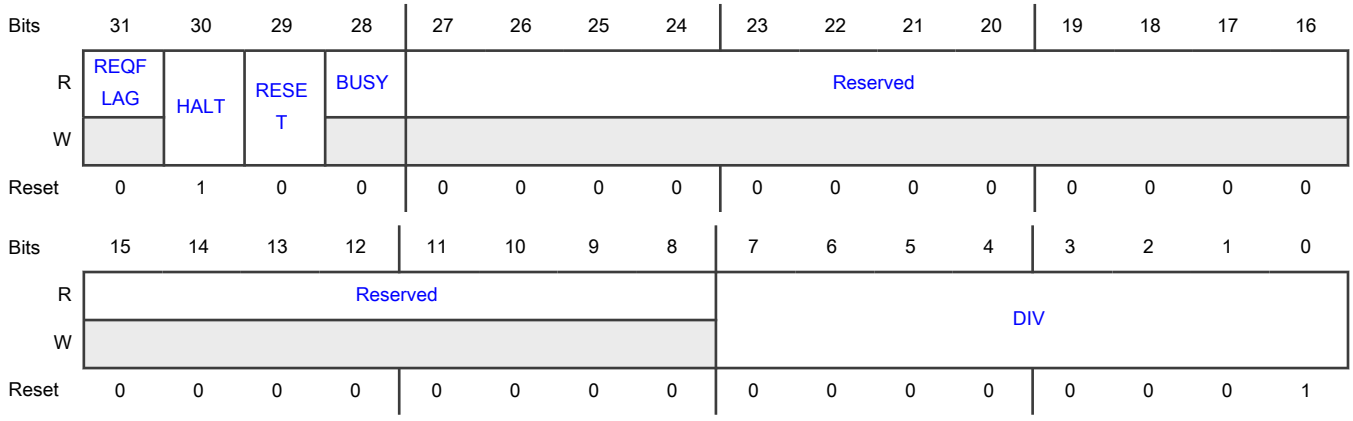
Offset

Register	Offset
SYSTICKFCLKDIV	764h

Function

Controls the divider for the Systick functional clock sense_systick_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Systick Clock Divider Counter Reset Resets the divider counter for Systick clock to a new divider value immediately instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

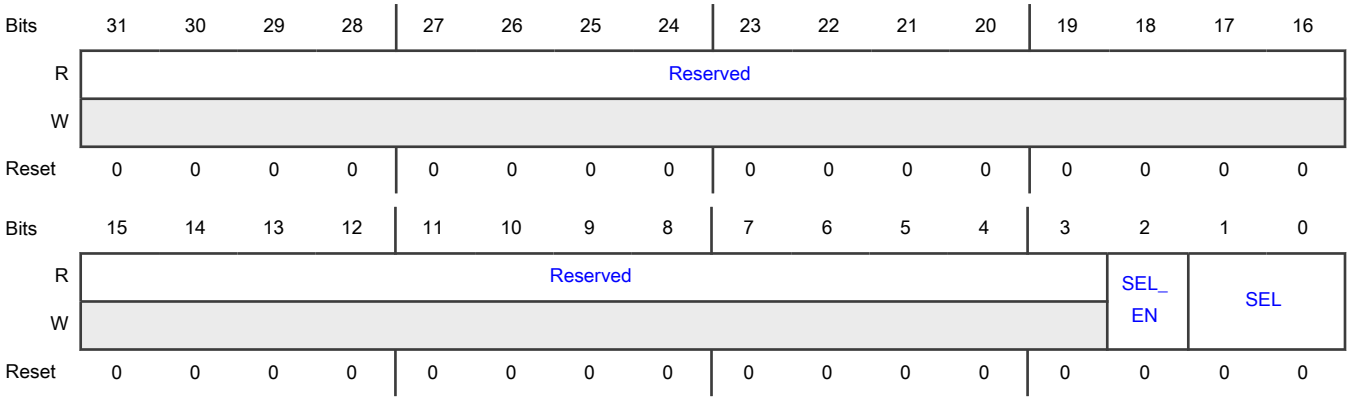
28.7.2.19 CTIMER5 Functional Clock Source Select (CTIMER5FCLKSEL)

Offset

Register	Offset
CTIMER5FCLKSEL	7A0h

Function
Selects the clock source for CTIMER5.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	CTIMER5 Functional Clock Source Select Selects the clock source for CTIMER5. 00b - baseclk_sense 01b - audio_clk 10b - fro2_max 11b - wake32k_clk

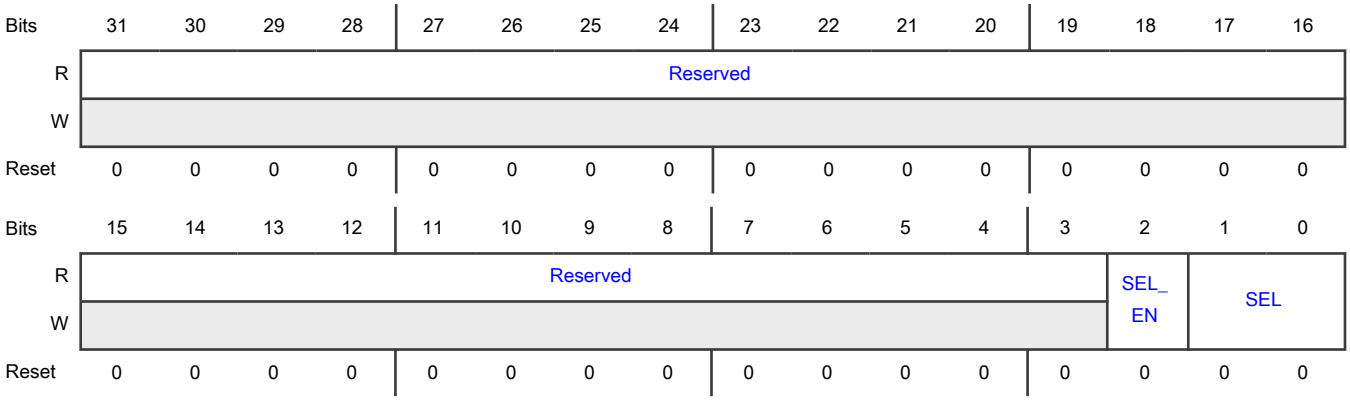
28.7.2.20 CTIMER6 Functional Clock Source Select (CTIMER6FCLKSEL)

Offset

Register	Offset
CTIMER6FCLKSEL	7A4h

Function
Selects the clock source for CTIMER6.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	CTIMER6 Functional Clock Source Select Selects the clock source for CTIMER6. 00b - baseclk_sense 01b - audio_clk 10b - fro2_max 11b - wake32k_clk

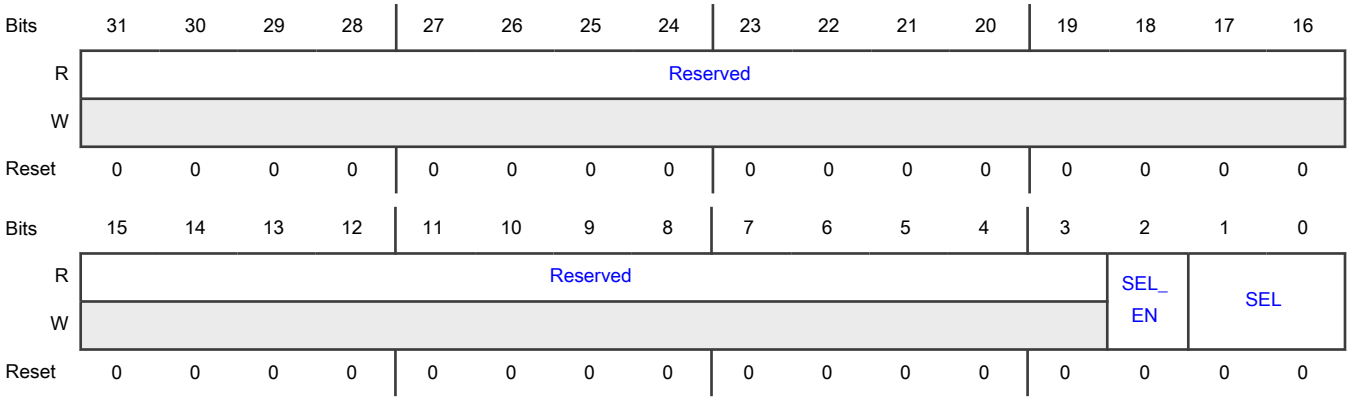
28.7.2.21 CTIMER7 Functional Clock Source Select (CTIMER7FCLKSEL)

Offset

Register	Offset
CTIMER7FCLKSEL	7A8h

Function
Selects the clock source for CTIMER7.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	CTIMER7 Functional Clock Source Select Selects the clock source for CTIMER7. 00b - baseclk_sense 01b - audio_clk 10b - fro2_max 11b - wake32k_clk

28.7.2.22 CTIMER5 Functional Clock Divider (CTIMER5CLKDIV)

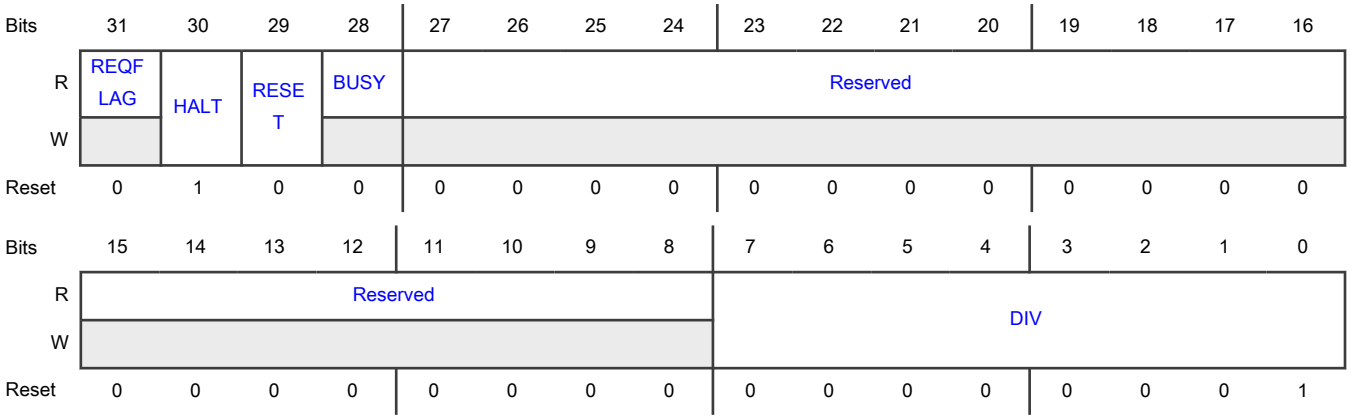
Offset

Register	Offset
CTIMER5CLKDIV	7B0h

Function

Controls the divider for CTIMER5 functional clock ctimer5_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter for CTIMER5 functional clock to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Divider Value Selection Selects a divider counter value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

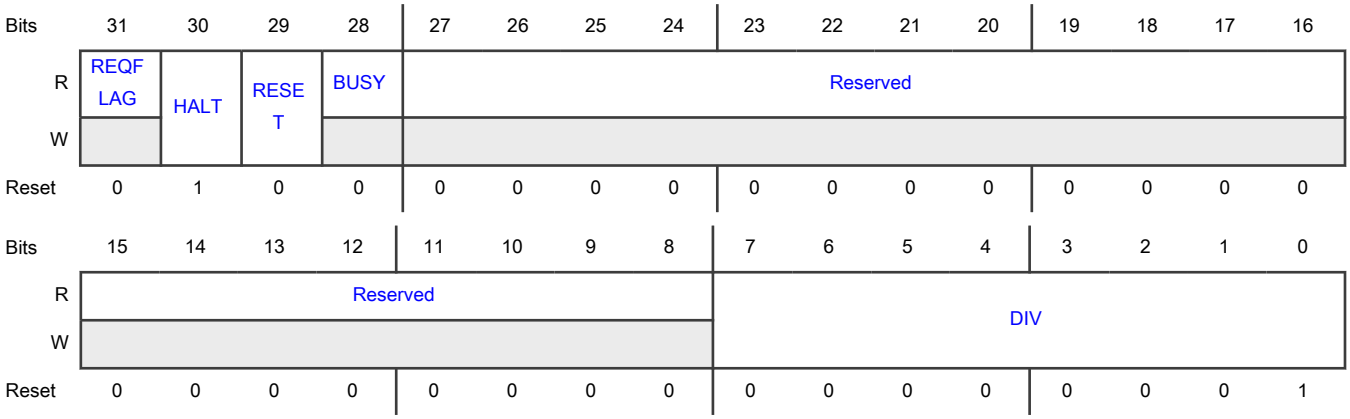
28.7.2.23 CTIMER6 Functional Clock Divider (CTIMER6CLKDIV)

Offset

Register	Offset
CTIMER6CLKDIV	7B4h

Function
Controls the divider for the CTIMER6 functional clock ctimer6_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Selection Selects a divider counter value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.2.24 CTIMER7 Functional Clock Divider (CTIMER7CLKDIV)

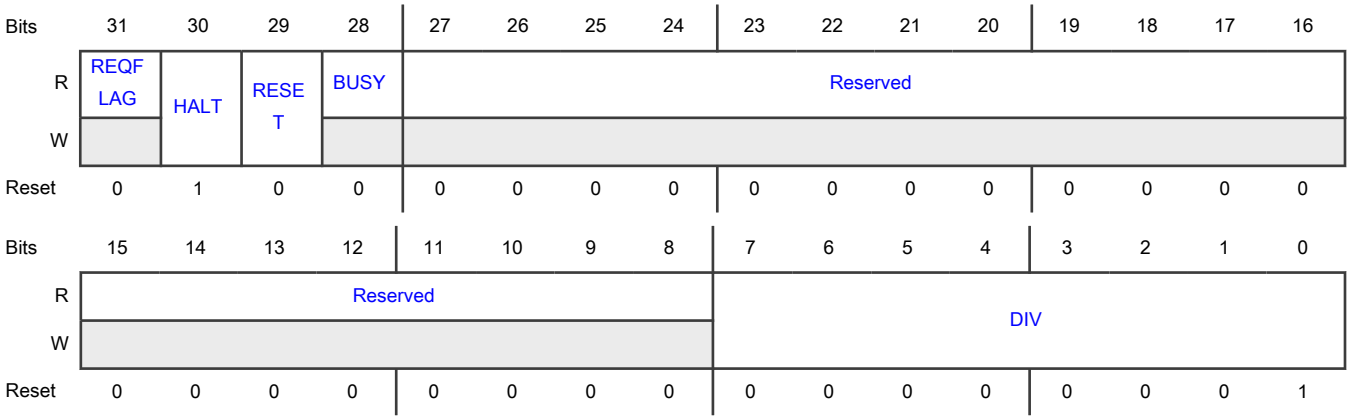
Offset

Register	Offset
CTIMER7CLKDIV	7B8h

Function

Controls the divider for CTIMER7 functional clock ctimer7_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has finished

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.2.25 I3C2 and I3C3 Functional Clock Source Select (I3C23FCLKSEL)

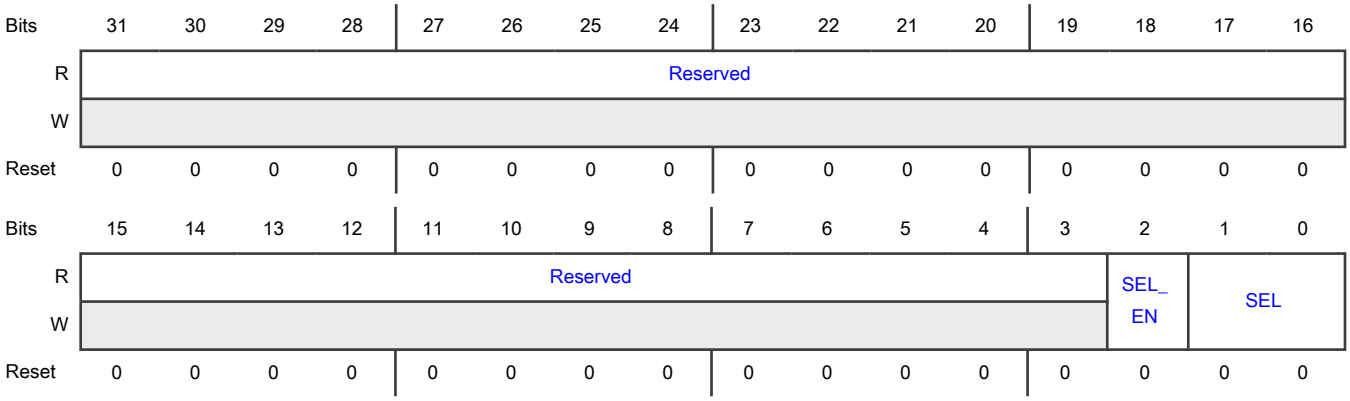
Offset

Register	Offset
I3C23FCLKSEL	800h

Function

Selects the clock source for the I3C0 and I3C1 functional clocks.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output for I3C functional clocks to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	I3C2 and I3C3 Clock Source Select Selects the clock source for I3C2 and I3C3 clocks. 00b - baseclk_sense 01b - fro2_max 10b - fro1_div8 11b - osc_clk

28.7.2.26 I3C2 and I3C3 Functional Clock Divider (I3C23FCLKDIV)

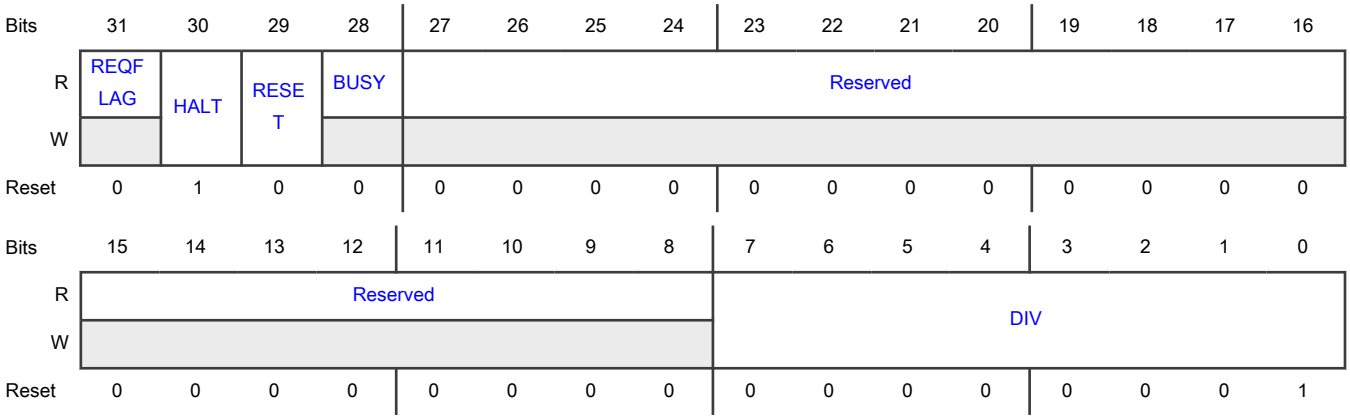
Offset

Register	Offset
I3C23FCLKDIV	810h

Function

Controls the divider for the I3C0 and I3C1 functional clock i3c23_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The Divider change has been finished (The clock to be divided must be running for this status to change). 1b - The divider value is being changed.
30 HALT	Divider Counter Halt Halts the Divider Counter. The intent is to allow the divider's clock source to be changed, without causing a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
7-0 DIV	I3C2 and I3C3 Clock Divider Value Selects the divider value: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

28.7.2.27 LP_FLEXCOMM17 Clock Source Select (FC17FCLKSEL)

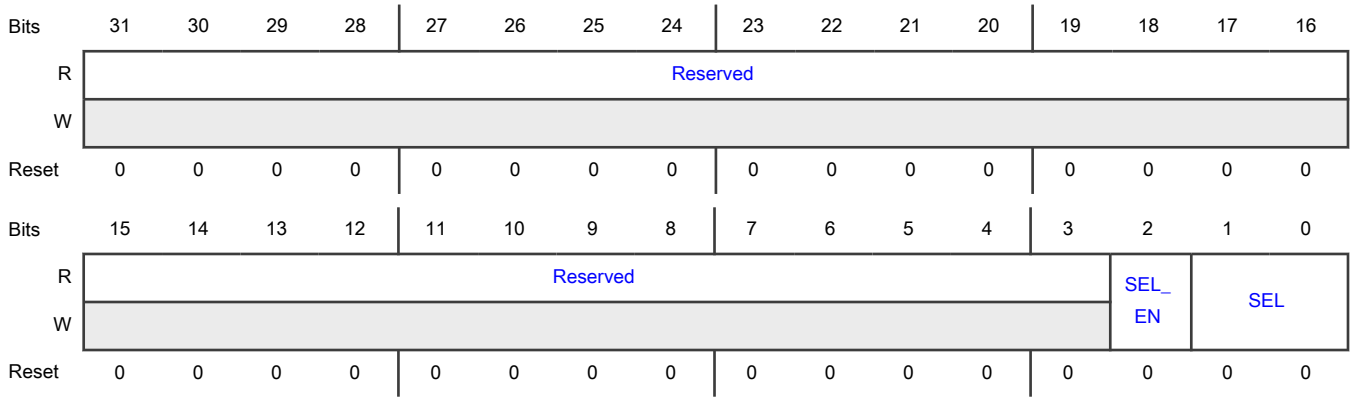
Offset

Register	Offset
FC17FCLKSEL	A00h

Function

Selects the clock source for LP_FLEXCOMM17.

Diagram



Fields

Field	Function
31-3 —	Reserved
2	Clock MUX Output Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEL_EN	Enables clock mux output for LP_FLEXCOMM17 to reduce power. 0b - Output of clkmux is gated. 1b - Enables output of clock mux.
1-0 SEL	LP_FLEXCOMM17 Clock Source Select Selects clock source for LP_FLEXCOMM 17 00b - baseclk_sense 01b - fro2_max 10b - fro1_max 11b - wake32k_clk

28.7.2.28 LP_FLEXCOMM17 Clock Divider (FC17FCLKDIV)

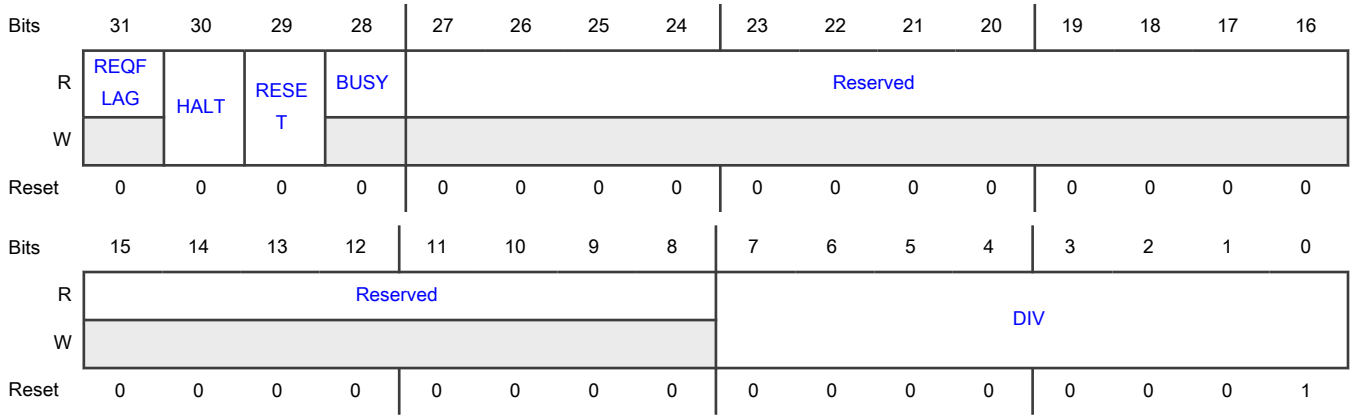
Offset

Register	Offset
FC17FCLKDIV	A04h

Function

Controls the divider for LP_FLEXCOMM17 clock flexcomm17_fclk.

Diagram



Fields

Field	Function
31	Request Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
REQFLAG	Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

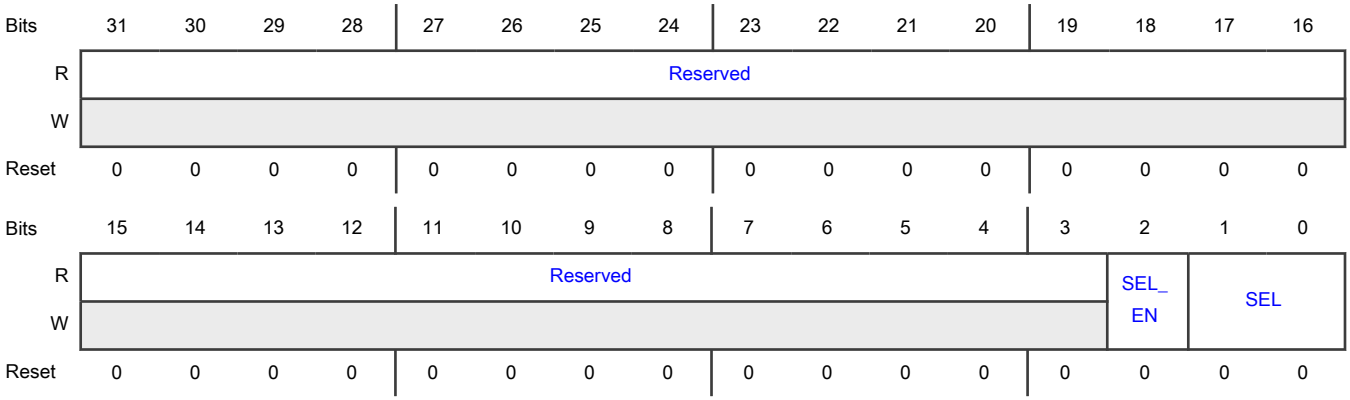
28.7.2.29 LP_FLEXCOMM18 Clock Source Select (FC18FCLKSEL)

Offset

Register	Offset
FC18FCLKSEL	A20h

Function
Selects the clock source for LP_FLEXCOMM18.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output for LP_FLEXCOMM18 to reduce power. 0b - Output of clkmux is gated. 1b - Enables output of clock mux.
1-0 SEL	LP_FLEXCOMM18 Clock Source Select Selects clock source for LP_FLEXCOMM 18 00b - baseclk_sense 01b - fro2_max 10b - fro1_max 11b - wake32k_clk

28.7.2.30 LP_FLEXCOMM18 Clock Divider (FC18FCLKDIV)

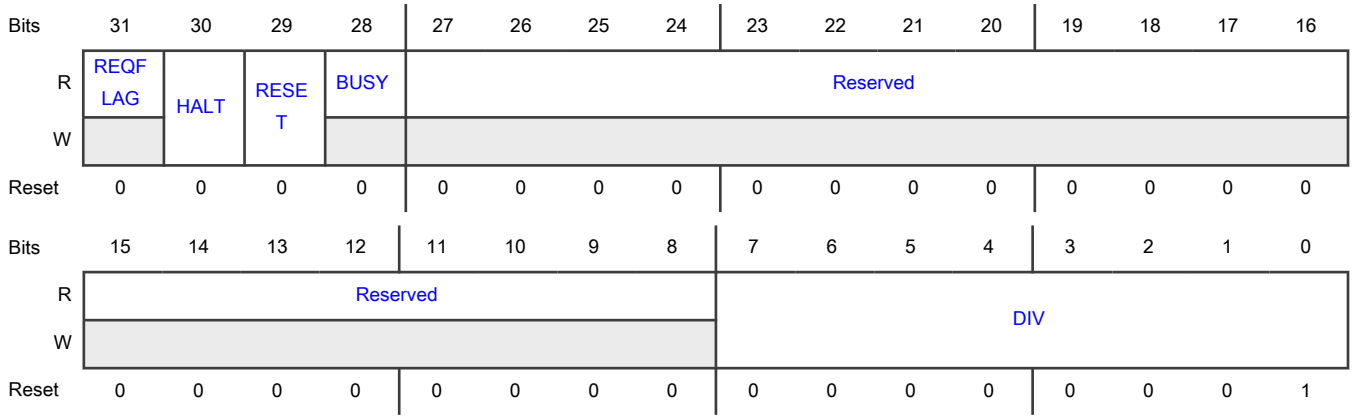
Offset

Register	Offset
FC18FCLKDIV	A24h

Function

Controls the divider for LP_FLEXCOMM18 clock flexcomm18_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

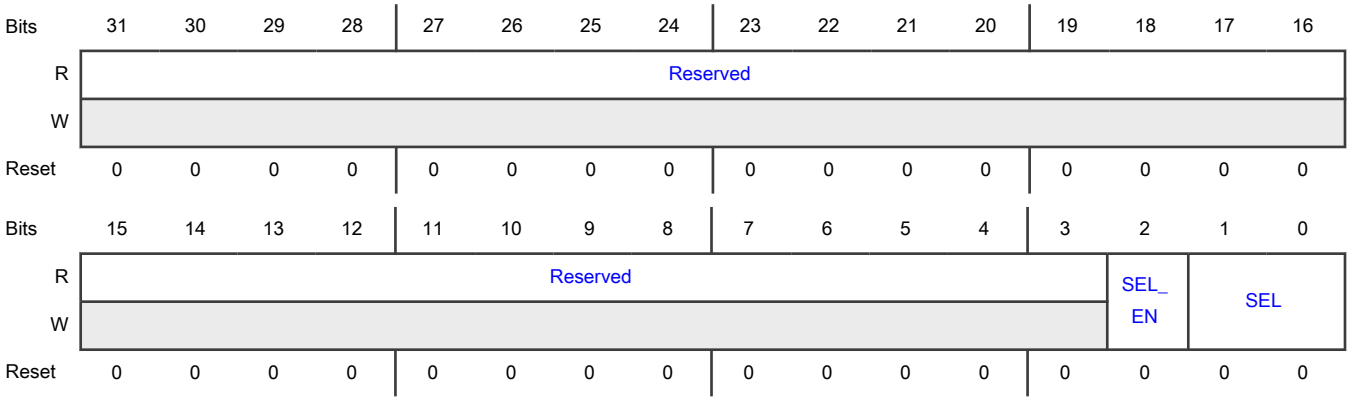
28.7.2.31 LP_FLEXCOMM19 Clock Source Select (FC19FCLKSEL)

Offset

Register	Offset
FC19FCLKSEL	A40h

Function
Selects the clock source for LP_FLEXCOMM19.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output for LP_FLEXCOMM19 to reduce power. 0b - Output of clkmux is gated. 1b - Enables output of clock mux.
1-0 SEL	LP_FLEXCOMM19 Clock Source Select Selects clock source for LP_FLEXCOMM 19 00b - baseclk_sense 01b - fro2_max 10b - fro1_max 11b - wake32k_clk

28.7.2.32 LP_FLEXCOMM19 Clock Divider (FC19FCLKDIV)

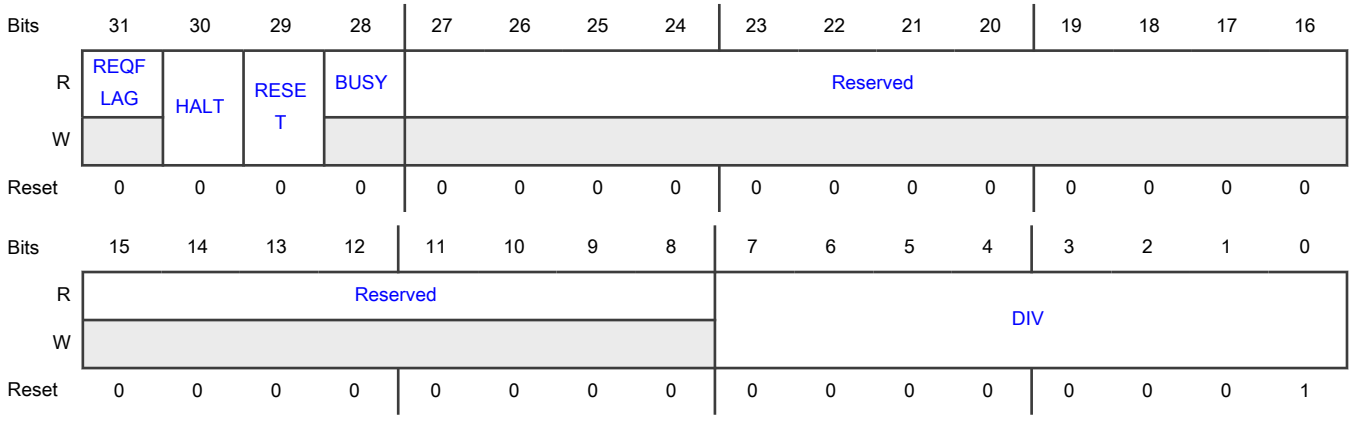
Offset

Register	Offset
FC19FCLKDIV	A44h

Function

Controls the divider for LP_FLEXCOMM19 clock flexcomm19_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

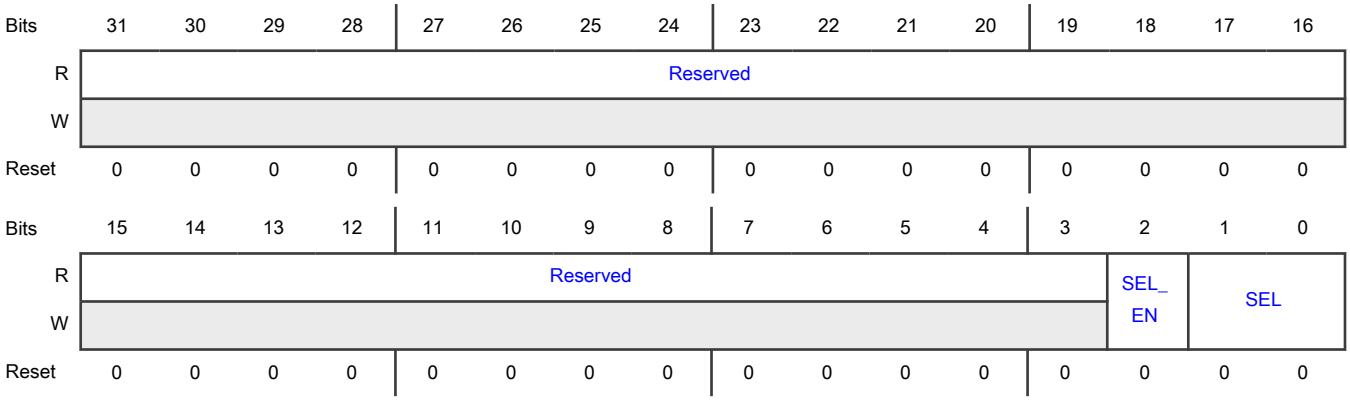
28.7.2.33 LP_FLEXCOMM20 Clock Source Select (FC20FCLKSEL)

Offset

Register	Offset
FC20FCLKSEL	A60h

Function
Selects the clock source for LP_FLEXCOMM20.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables clock mux output for LP_FLEXCOMM20 to reduce power. 0b - Output of clkmux is gated. 1b - Enables output of clock mux.
1-0 SEL	LP_FLEXCOMM20 Clock Source Select Selects clock source for LP_FLEXCOMM 20 00b - baseclk_sense 01b - fro2_max 10b - fro1_max 11b - wake32k_clk

28.7.2.34 LP_FLEXCOMM20 Clock Divider (FC20FCLKDIV)

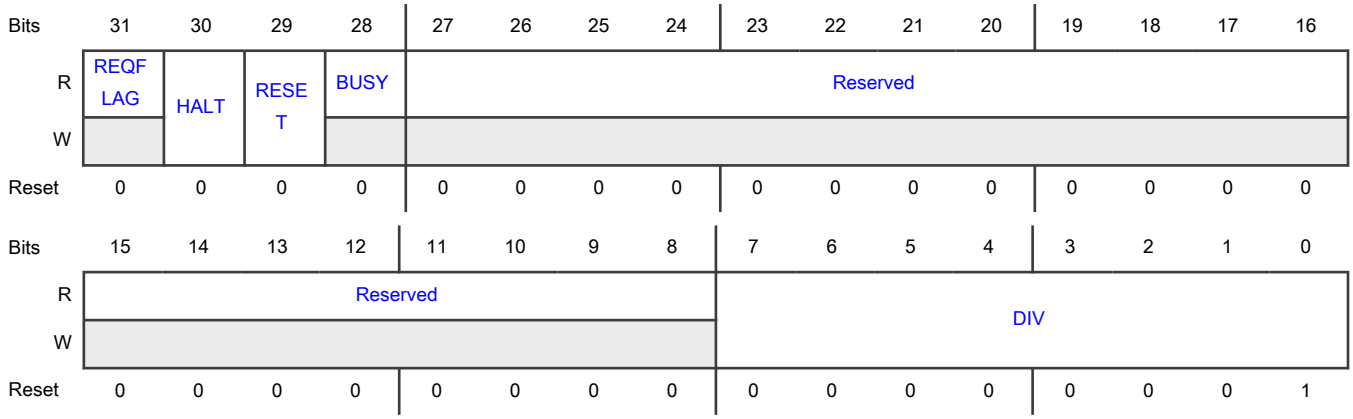
Offset

Register	Offset
FC20FCLKDIV	A64h

Function

Controls the divider for LP_FLEXCOMM20 clock flexcomm20_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished 1b - A change is being made to the divider value
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

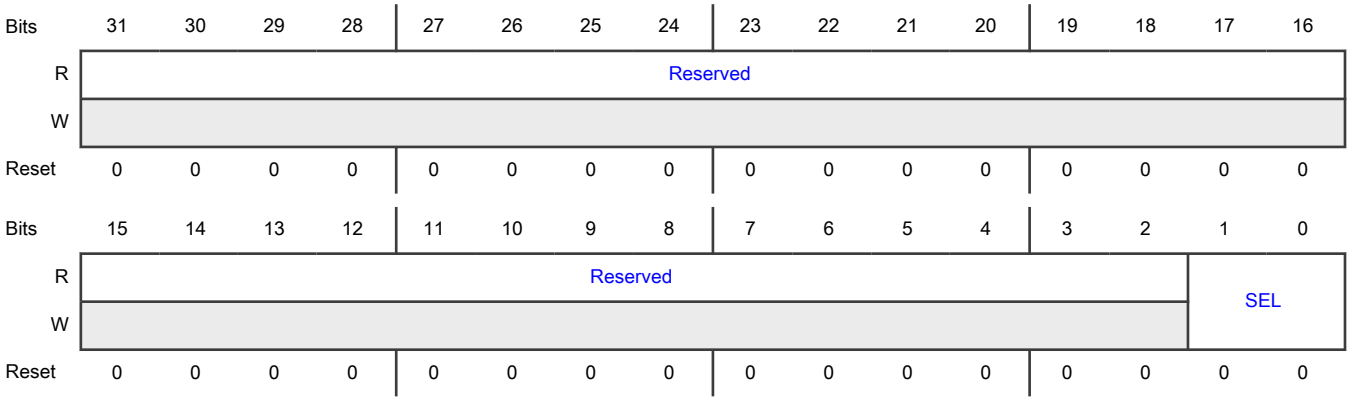
28.7.2.35 VDD1_SENSE Audio Clock Source (AUDIOVDD1CLKSEL)

Offset

Register	Offset
AUDIOVDD1CLKSEL	AA0h

Function
Selects the clock source for audio clock in VDD1_SENSE domain.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Clock Source Select Selects the clock source for audio clock in VDD1_SENSE. 00b - mclk_in 01b - osc_clk 10b - fro2_div8 11b - audio_pll_pfd3

28.7.3 CLKCTL2 register descriptions

This section includes the memory map and register descriptions for Clock Controller 2.

28.7.3.1 CLKCTL2 memory map

CLKCTL2 base address: 4006_5000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDDN_COM Peripheral clock 0 (PSCCTL0)	32	RW	0000_0018h
40h	VDDN_COM Peripheral Clock 0 Set (PSCCTL0_SET)	32	RW	0000_0000h
70h	VDDN_COM Peripheral Clock 0 Clear (PSCCTL0_CLR)	32	RW	0000_0000h
A4h	VDDN_COM Clock Source Select (COMMONVDDNCLKSEL)	32	RW	0000_0000h
ACh	VDDN_COM Clock Divider (COMMONVDDNCLKDIV)	32	RW	4000_0001h
100h	XTAL Oscillator Control 0 (SYSOSCCTL0)	32	RW	0000_0000h
108h	OSC Bypass Clock Source Select (SYSOSCBYPASS)	32	RW	0000_0000h
10Ch	USB0 Clock Source Select (USBCLKSRC24MCLKSEL)	32	RW	0000_0000h
110h	VDDN_COM Base Clock Source Select (COMNBASECLKSEL)	32	RW	0000_0000h
11Ch	USB1 (eUSB) Clock Source Select (EUSBCLKSRC24MCLKSEL)	32	RW	0000_0000h
200h	Main PLL0 Clock Source Select (MAINPLL0CLKSEL)	32	RW	0000_0000h
204h	Main PLL0 Control 0 (MAINPLL0CTL0)	32	RW	0016_0002h
20Ch	Main PLL0 Lock Time Div2 (MAINPLL0LOCKTIMEDIV2)	32	RW	0000_CAFEh
210h	Main PLL0 Numerator (MAINPLL0NUM)	32	RW	04DD_2F15h
214h	Main PLL0 Denominator (MAINPLL0DENOM)	32	RW	1FFF_FFDBh
218h	Main PLL0 PFD (MAINPLL0PFD)	32	RW	8080_8080h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
220h	Main PLL0 PFD Clock Domain Enable (MAINPLL0PFDDOMAINEN)	32	RW	7F7F_7F7Fh
400h	Audio PLL0 Clock Source Select (AUDIOPLL0CLKSEL)	32	RW	0000_0000h
404h	Audio PLL0 Control 0 (AUDIOPLL0CTL0)	32	RW	0016_0002h
40Ch	Audio PLL0 Lock Time Divide-by-2 (AUDIOPLL0LOCKTIMEDIV2)	32	RW	0000_CAFEh
410h	Audio PLL0 Numerator (AUDIOPLL0NUM)	32	RW	04DD_2F15h
414h	Audio PLL0 Denominator (AUDIOPLL0DENOM)	32	RW	1FFF_FFDBh
418h	Audio PLL0 PFD (AUDIOPLL0PFD)	32	RW	8080_8080h
420h	Audio PLL0 PFD Clock Enable (AUDIOPLL0PFDDOMAINEN)	32	RW	7F7F_7F7Fh
424h	Audio PLL0 VCO Clock Enable (AUDIOPLL0VCODOMAINEN)	32	RW	0000_007Fh
600h	CKIL 32kHz Clock Gate (CKIL_32K_GATE)	32	RW	0000_0001h

28.7.3.2 VDDN_COM Peripheral clock 0 (PSCCTL0)

Offset

Register	Offset
PSCCTL0	10h

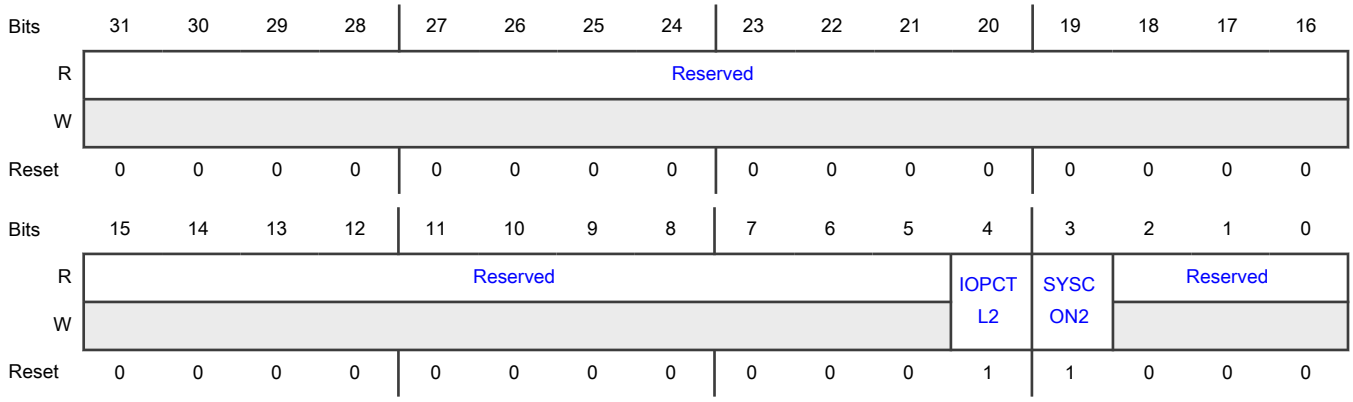
Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing zero disables the clock. If VDDN_COM power rail is OFF, reset after power ON.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 IOPCTL2	IOPCTL2 clock Enables IOPCTL2 clock. 0b - Disable 1b - Enable
3 SYSCON2	SYSCON2 clock Enables SYSCON2 clock. 0b - Disable 1b - Enable
2-0 —	Reserved

28.7.3.3 VDDN_COM Peripheral Clock 0 Set (PSCCTL0_SET)

Offset

Register	Offset
PSCCTL0_SET	40h

Function

Writing an one to a bit position in PSCCTL0_SET sets the corresponding position in PSCCTL0.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											IOPCTL2	SYSCON2	Reserved		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-5 —	Reserved
4 IOPCTL2	IOPCTL2 Clock Enables IOPCTL2 clock. 0b - No effect 1b - Enable
3 SYSCON2	SYSCON2 Clock Enables SYSCON2 clock. 0b - No effect 1b - Enable
2-0 —	Reserved

28.7.3.4 VDDN_COM Peripheral Clock 0 Clear (PSCCTL0_CLR)

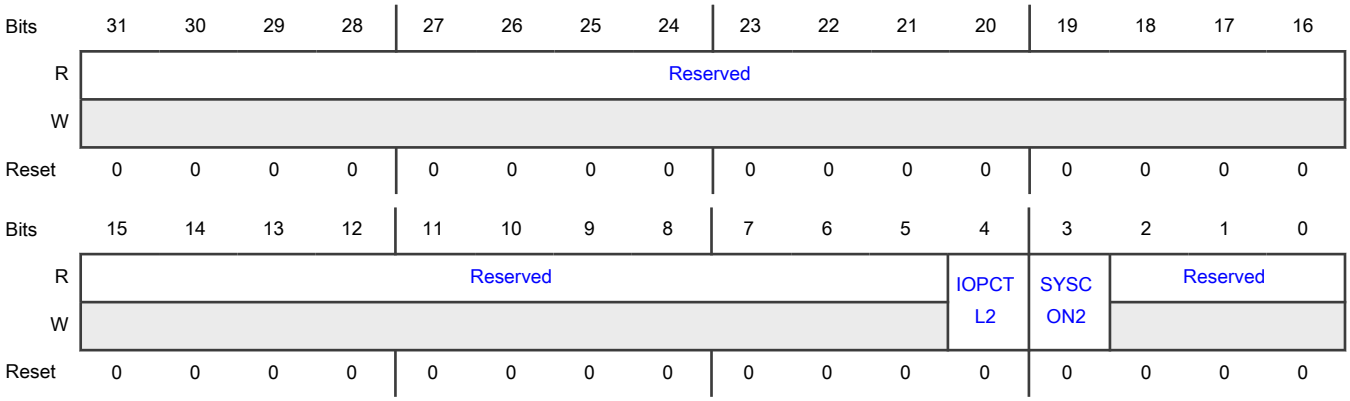
Offset

Register	Offset
PSCCTL0_CLR	70h

Function

Writing an one to a bit position in PSCCTL0_CLR clears the corresponding position in PSCCTL0.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 IOPCTL2	IOPCTL2 Clock Disables IOPCTL2 clock. 0b - No effect 1b - Disable
3 SYSCON2	SYSCON2 Clock Disables SYSCON2 clock. 0b - No effect 1b - Disable
2-0 —	Reserved

28.7.3.5 VDDN_COM Clock Source Select (COMMONVDDNCLKSEL)

Offset

Register	Offset
COMMONVDDNCLKSEL	A4h

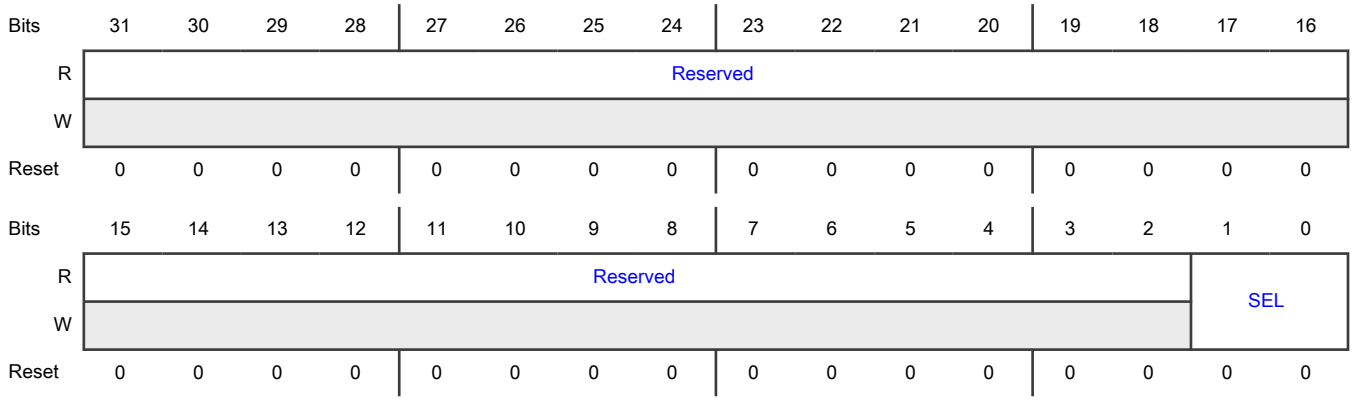
Function

Selects a clock source for VDDN_COM clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDDN_COM Clock Source Select Selects a clock source for VDDN_COM clock. 00b - baseclk_comn 01b - main_pll_pfd3 10b - fro1_max 11b - osc_clk

28.7.3.6 VDDN_COM Clock Divider (COMMONVDDNCLKDIV)

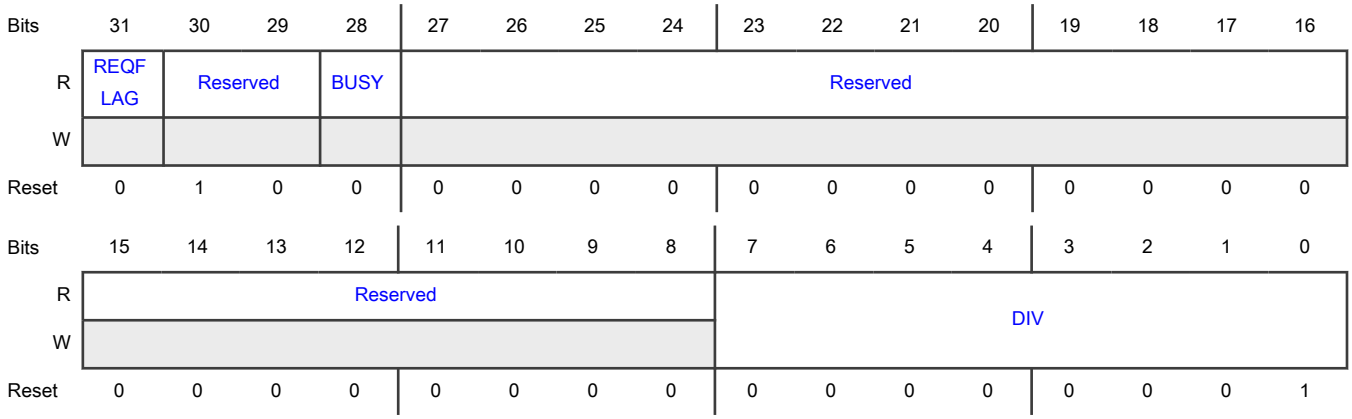
Offset

Register	Offset
COMMONVDDNCLKDIV	ACh

Function

Controls the divider for the VDDN_COM functional clock common_vddn_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The clkout is outputed with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a divider value for VDDN_COM clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.3.7 XTAL Oscillator Control 0 (SYSOSCCTL0)

Offset

Register	Offset
SYSOSCCTL0	100h

Function

Controls whether the XTAL oscillator is bypassed, and sets the oscillator mode.

One aspect of the oscillator high gain mode is that a larger voltage swing is used at the crystal pin. This gives a higher noise immunity within the oscillator and less edge to edge jitter of the internal clock. When high gain mode is not required, power used by the XTAL oscillator can be reduced by using low power mode.

NOTE

High gain mode requires a 1 megohm resistor to be inserted in series with the external crystal. For this reason, high gain mode and low power mode cannot both be used in the same application. The board design must reflect the mode that will be used.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														BYPASS_...	LP_ENAB...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1 BYPASS_ENABLE	Bypass Enable Enables bypass mode. 0b - Enables normal mode. 1b - Enables bypass mode. In this mode a clock can be directly inputted into the XTALIN pin.
0 LP_ENABLE	Low Power Mode Enable Enables low power mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Enables High Gain mode (HP) 1b - Enables Low Power mode (LP)

28.7.3.8 OSC Bypass Clock Source Select (SYSOSCBYPASS)

Offset

Register	Offset
SYSOSCBYPASS	108h

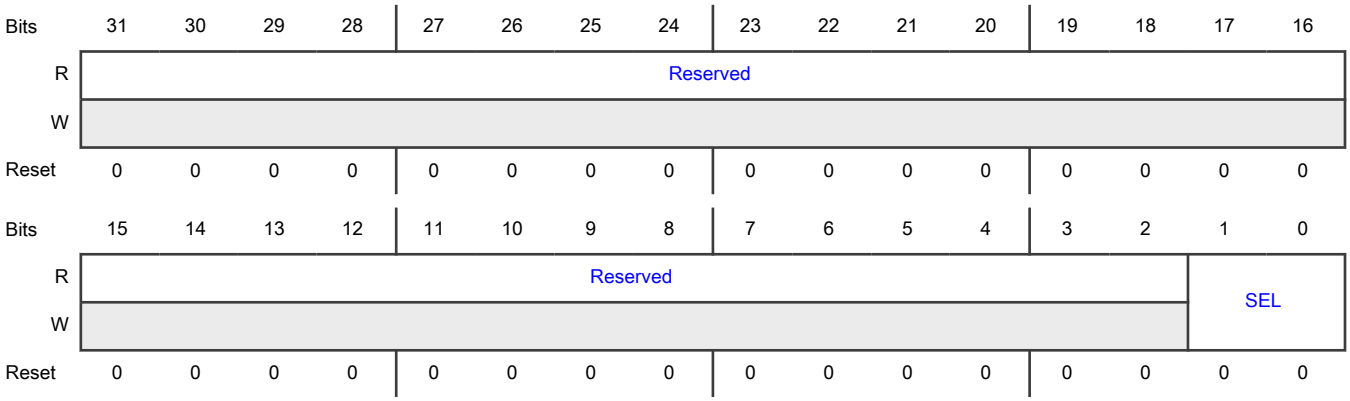
Function

Allows selecting either the Xtal oscillator or the CLKIN function as the source for the internal clk_in signal.

NOTE

If SEL > 0, the oscillator is not being used and can be turned off to save power.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	SYSOSC Bypass Clock Select Source Selects a clock source for SYSOSC bypass. 00b - osc_out 01b - CLKIN

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Reserved 11b - Reserved

28.7.3.9 USB0 Clock Source Select (USBCLKSRC24MCLKSEL)

Offset

Register	Offset
USBCLKSRC24MCLKSEL	10Ch

Function

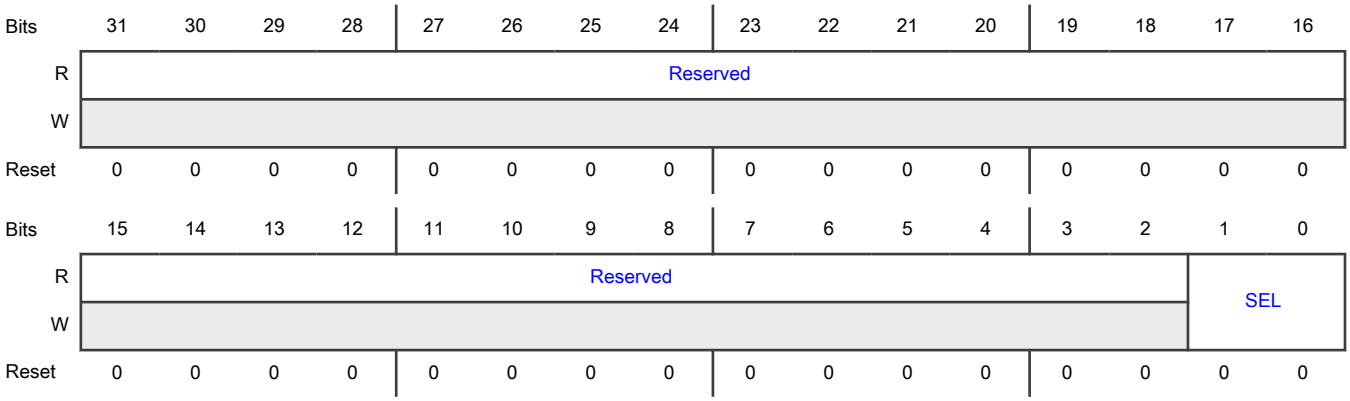
Selects a clock source for 24M USB0 OSC clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

The usage of clock selection is described in USB0 instances feature of USB chapter.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	USB0 24M Input Clock Select Source Selects a clock source for 24M USB input clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE The usage of clock selection is described in USB0 instances feature support" of USB chapter. 00b - osc_clk 01b - fro_div8 10b - Reserved 11b - Reserved

28.7.3.10 VDDN_COM Base Clock Source Select (COMNBASECLKSEL)

Offset

Register	Offset
COMNBASECLKSEL	110h

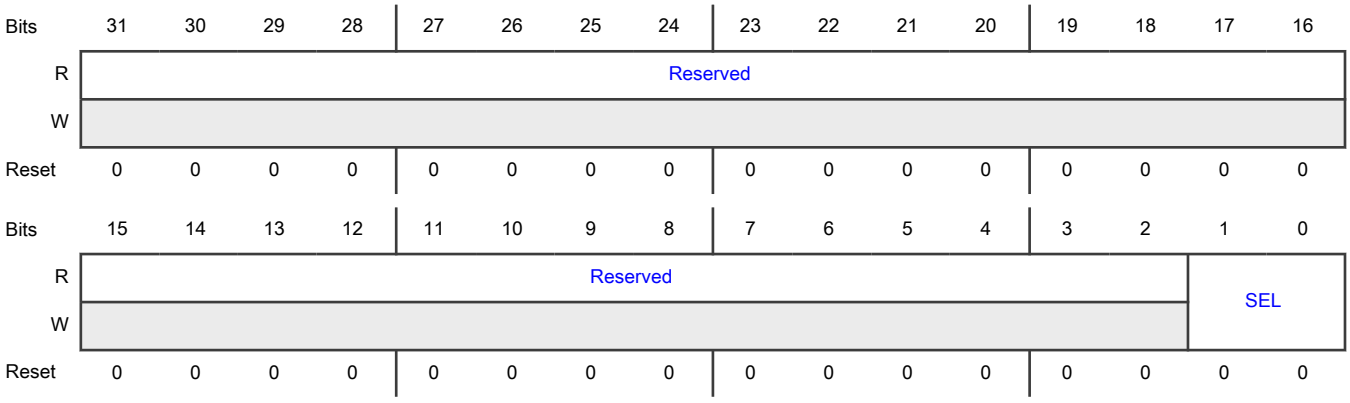
Function

Selects a clock source for VDDN_COM base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
1-0 SEL	Common Base Clock Source Select Selects a source for VDDN_COM base clock. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.3.11 USB1 (eUSB) Clock Source Select (EUSBCLKSRC24MCLKSEL)

Offset

Register	Offset
EUSBCLKSRC24MCLKSEL	11Ch

Function

Selects a clock source for 24M USB1 (eUSB) OSC clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

The usage of clock selection is described in USB1 instances feature of USB chapter.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 SEL	24M USB1 (eUSB) Input Clock Source Selects a source for 24M USB1 (eUSB) input clock. **NOTE** The usage of clock selection is described in USB1 instances feature support" of USB chapter. 00b - osc_clk 01b - fro1_div8 10b - Reserved 11b - Reserved

28.7.3.12 Main PLL0 Clock Source Select (MAINPLL0CLKSEL)

Offset

Register	Offset
MAINPLL0CLKSEL	200h

Function

The MAINPLL0 output frequency is controlled by the programmed values of MULT, NUM and DENOM, and can be expressed as a function of the MAINPLL0 reference input frequency:

$F_{MAINPLL0_OUTPUT} = f_{PLLREFIN} * (MULT + (NUM/DENOM))$

The input reference frequency can be selected using MAINPLL0CLKSEL[SEL].

NOTE

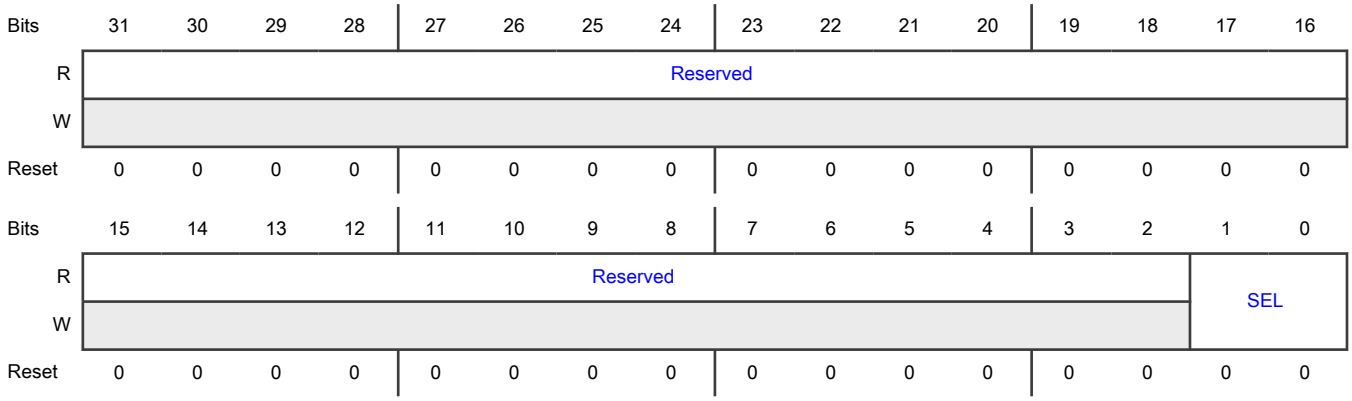
The PLL reference input frequency is limited to the following discrete frequencies: 32, 30, 26, 24, 19.2, 16 MHz.

After choosing an input frequency, choose MULT so that the MAINPLL0 output frequency is 528 MHz. For example, after choosing a MAINPLL0 reference input frequency of 24 MHz, program MULT as 0x16 (decimal 22), because 24 MHz * 22 = 528 MHz. To fine tune the frequency, use NUM and DENOM.

NOTE

This clock selection is internally synchronized. The clocks that are switched from and switched to must run for clock selection to change.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Main PLL0 Clock Select Source Selects a clock source for main PLL0 input clock. 00b - fro1_div8 01b - osc_clk 10b - Reserved 11b - Reserved

28.7.3.13 Main PLL0 Control 0 (MAINPLL0CTL0)

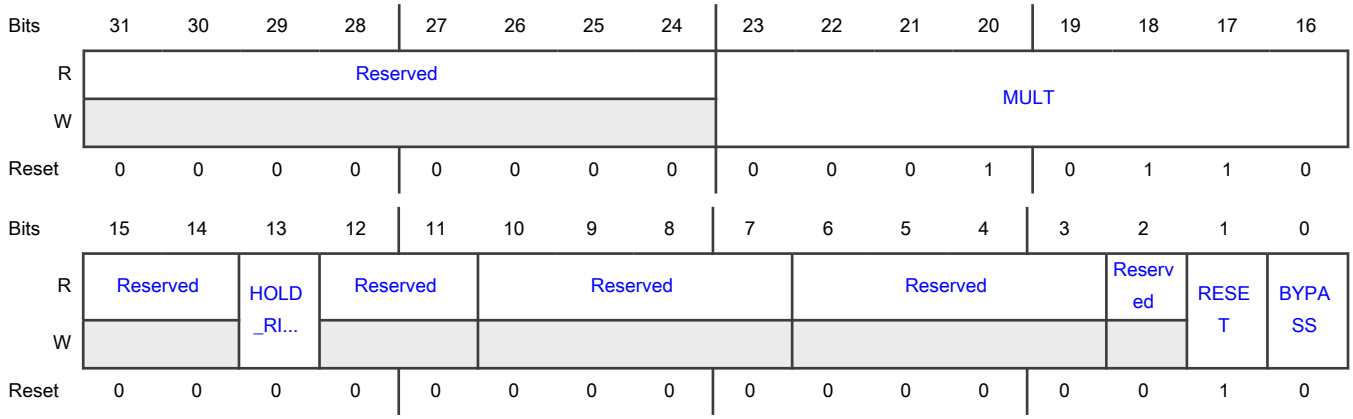
Offset

Register	Offset
MAINPLL0CTL0	204h

Function

Enables control for main PLL0 clock.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 MULT	Multiplication Factor Multiplication factor for $F_{MAINPLL0_OUTPUT}$. The value of MULT is directly decoded. 0001_0000b - Divide by 16 0001_0001b - Divided by 17 0001_0010b - Divided by 18 0001_0011b - Divided by 19 0001_0100b - Divided by 20 0001_0101b - Divided by 21 0001_0110b - Divided by 22
15-14 —	Reserved
13 HOLD_RING_OFF_ENA	Hold Ring Off Control Enables hold ring off to avoid multi-wave problems within the VCO. **NOTE** Asserts HOLD_RING_OFF_ENA bit in the 1st half of lock time. De-asserts HOLD_RING_OFF_ENA bit in the 2nd half of lock time. 0b - Disable 1b - Enable
12-11 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
10-7 —	Reserved
6-3 —	Reserved
2 —	Reserved
1 RESET	MAINPLL0 Reset Enables MAINPLL0 reset. 0b - MAINPLL0 reset is removed. 1b - MAINPLL0 is placed into reset.
0 BYPASS	MAINPLL0 BYPASS Enables bypass for MAINPLL0. 0b - PFD outputs are PFD-programmed clocks. 1b - Bypass mode: PFD outputs are sourced directly from the reference input clock.

28.7.3.14 Main PLL0 Lock Time Div2 (MAINPLL0LOCKTIMEDIV2)

Offset

Register	Offset
MAINPLL0LOCKTIMEDIV2	20Ch

Function

During reset, the BOOTROM will copy the MAINPLL0 worst case characterized lock time value found in OTP Fuses into the respective fields (if the OTP valid signature is present).

To get the information on when half of the lock time has expired and when the full lock has passed, software needs this worst case lock time value to control the Hold Ring Off Control bit (MAINPLL0CTL0[HOLD_RING_OFF]). Any PLL lock sequence requires the HOLD_RING_OFF bit:

- to be enabled (‘1’) during the 1st half of the lock time
- to be disabled (‘0’) during the 2nd half of the lock time

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKTIMEDIV2															
W																
Reset	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	0

Fields

Field	Function
31-16 —	Reserved
15-0 LOCKTIMEDIV2	MAINPLL0 Lock Time Divide-by-2 The programmed lock time is in µs (micro-seconds) and is programmed as half of the actual lock time value.

28.7.3.15 Main PLL0 Numerator (MAINPLL0NUM)

Offset

Register	Offset
MAINPLL0NUM	210h

Function

Provides numerator for the main PLL fractional divider.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUM											
W																
Reset	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NUM															
W																
Reset	0	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1

Fields

Field	Function
31-30 —	Reserved
29-0 NUM	Numerator Sets a numerator value. **NOTE** The value of numerator must always be configured to be less than the value of the denominator. The MAINPLL0NUM register can only be changed when the MAINPLL0 is disabled.

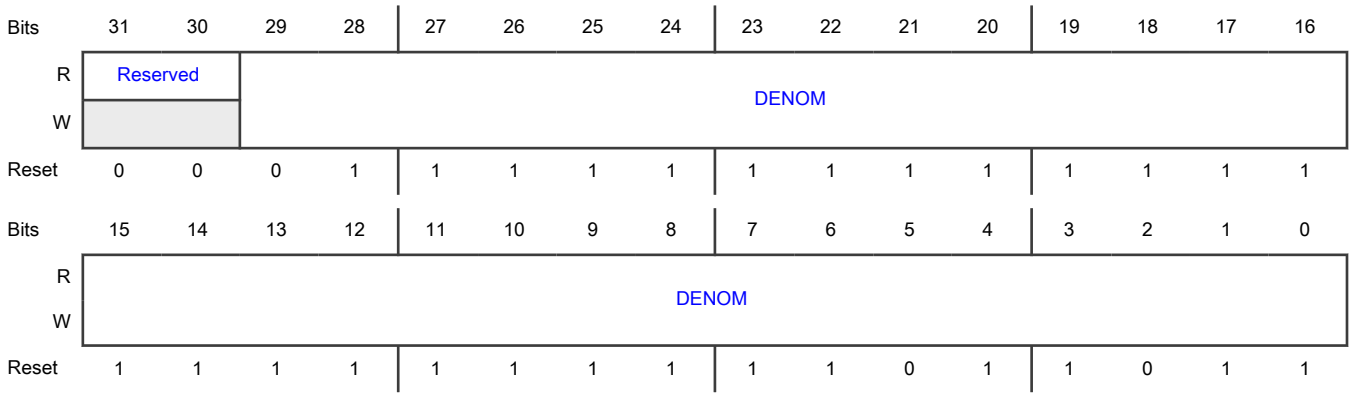
28.7.3.16 Main PLL0 Denominator (MAINPLL0DENOM)

Offset

Register	Offset
MAINPLL0DENOM	214h

Function
Provides the denominator for the main PLL fractional divider.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-0	Denominator

Table continues on the next page...

Table continued from the previous page...

Field	Function
DENOM	Sets a denominator value. **NOTE** The value of numerator must always be configured to be less than the value of the denominator. The MAINPLL0DENOM register can only be changed when the MAINPLL0 is disabled.

28.7.3.17 Main PLL0 PFD (MAINPLL0PFD)

Offset

Register	Offset
MAINPLL0PFD	218h

Function

Defines the control bits for PFD0 to PFD3 clocks derived from MAINPLL0. The PFD output frequencies are governed by:

$F_{PFDx} = F_{MAINPLL0_OUT} * (18 / PFDx)$

When changing PFD values, follow this recommended sequence:

- Ensure that the PFDx clock gets gated by writing 1 to the PFDx_CLKGATE bit.
- Program the new PFD value (which are limited to decimal values 12 to 35).
- Write 0 to the PFDx_CLKGATE register to ungate the PFDx clock.
- Poll the PFDx_CLKRDY status flag until it (the status flag) is set to 1.
- Clear the PFDx_CLKRDY status flag by writing 1 to that specific PFDx_CLKRDY bit.

NOTE

In non-bypass mode, the pfd output should be disabled by the PFDn_CLKGATE bit, until PLL lock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PFD3_CL...	PFD3_CL...	PFD3						PFD2_CL...	PFD2_CL...	PFD2					
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PFD1_CL...	PFD1_CL...	PFD1						PFD0_CL...	PFD0_CL...	PFD0					
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Function
31 PFD3_CLKGATE	PFD3 Clock Gate Enables PFD3 clock gate. 0b - Disable 1b - Enable
30 PFD3_CLKRDY	PFD3 Clock Ready Indicates the PFD3 clock status. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - PFD3 clock is not ready. 1b - PFD3 clock is ready.
29-24 PFD3	PLL Fractional Divider 3 Sets the fractional divider value in the range of 12 to 35 (decimal).
23 PFD2_CLKGATE	PFD2 Clock Gate Enables PFD2 clock gate. 0b - Disable 1b - Enable
22 PFD2_CLKRDY	PFD2 Clock Ready Indicates PFD2 clock status. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - PFD2 clock is not ready. 1b - PFD2 clock is ready.
21-16 PFD2	PLL Fractional Divider 2 Sets the fractional divider value in the range of 12 to 35 (decimal).
15 PFD1_CLKGATE	PFD1 Clock Gate Enables PFD1 clock gate. 0b - Disable 1b - Enable
14 PFD1_CLKRDY	PFD1 Clock Ready Indicates the PFD1 clock status. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - PFD1 clock is not ready.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - PFD1 clock is ready.
13-8 PFD1	PLL Fractional Divider 1 Sets the fractional divider value in the range of 12 to 35 (decimal).
7 PFD0_CLKGATE	PFD0 Clock Gate Enables PDF0 clock gate. 0b - Disable 1b - Enable
6 PFD0_CLKRDY	PFD0 Clock Ready Indicates the PFD0 clock status. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - PFD0 clock is not ready. 1b - PFD0 clock is ready.
5-0 PFD0	PLL Fractional Divider 0 Sets the fractional divider value in the range of 12 to 35 (decimal).

28.7.3.18 Main PLL0 PFD Clock Domain Enable (MAINPLL0PFDDOMAINEN)

Offset

Register	Offset
MAINPLL0PFDDOMAINEN	220h

Function

Enables PFD clock of MAINPLL for various domains. Each PFD clock of MAINPLL can be sent to various domain. To reduce power consumption, turn off the PFD in the domain when this PFD is not configured.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	Reserved	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	Reserved	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Fields

Field	Function
31 —	Reserved
30 PFD3_OF_COMN	PFD3 of VDDN_COM Domain Enables PFD3 clock of VDDN_COM domain. 0b - Disable 1b - Enable
29 PFD3_OF_VDD2_COM	PFD3 of VDD2_COM Domain Enables PFD3 clock of VDD2_COM domain. 0b - Disable 1b - Enable
28 PFD3_OF_MDN	PFD3 of VDDN_MEDIA Domain Enables PFD3 clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
27 PFD3_OF_MD2	PFD3 of VDD2_MEDIA Domain Enables PFD3 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
26 PFD3_OF_VDD2_DSP	PFD3 of VDD2_DSP Domain Enables PFD3 clock of VDD2_DSP domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
25 PFD3_OF_VDD1_SENSE	PFD3 of VDD1_SENSE Domain Enables PFD3 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
24 PFD3_OF_CMPT	PFD3 of VDD2_COMP Domain Enables PFD3 clock of VDD2_COMP domain. 0b - Disable 1b - Enable
23 —	Reserved
22 PFD2_OF_COMN	PFD2 of VDDN_COM Domain Enables PFD2 clock of VDDN_COM domain. 0b - Disable 1b - Enable
21 PFD2_OF_VDD2_COM	PFD2 of VDD2_COM Domain Enables PFD2 clock of VDD2_COM domain. 0b - Disable 1b - Enable
20 PFD2_OF_MDN	PFD2 of VDDN_MEDIA Domain Enables PFD2 clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
19 PFD2_OF_MD2	PFD2 of VDD2_MEDIA Domain Enables PFD2 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
18 PFD2_OF_VDD2_DSP	PFD2 of VDD2_DSP Domain Enables PFD2 clock of VDD2_DSP domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
17 PFD2_OF_VDD1_SENSE	PFD2 of VDD1_SENSE Domain Enables PFD2 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
16 PFD2_OF_CMPT	PFD2 of VDD2_COMP Domain Enables PFD2 clock of VDD2_COMP domain. 0b - Disable 1b - Enable
15 —	Reserved
14 PFD1_OF_COMN	PFD1 of VDDN_COM Domain Enables PFD1 clock of VDDN_COM domain. 0b - Disable 1b - Enable
13 PFD1_OF_VDD2_COM	PFD1 of VDD2_COM Domain Enables PFD1 clock of VDD2_COM domain. 0b - Disable 1b - Enable
12 PFD1_OF_MDN	PFD1 of VDDN_MEDIA Domain Enables PFD1 clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
11 PFD1_OF_MD2	PFD1 of VDD2_MEDIA Domain Enables PFD1 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
10 PFD1_OF_VDD2_DSP	PFD1 of VDD2_DSP Domain Enables PFD1 clock of VDD2_DSP domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
9 PFD1_OF_VDD1_SENSE	PFD1 of VDD1_SENSE Domain Enables PFD1 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
8 PFD1_OF_CMPT	PFD1 of VDD2_COMP Domain Enables PFD1 clock of VDD2_COMP domain. 0b - Disable 1b - Enable
7 —	Reserved
6 PFD0_OF_COMN	PFD0 of VDDN_COM Domain Enables PFD0 clock of VDDN_COM domain. 0b - Disable 1b - Enable
5 PFD0_OF_VDD2_COM	PFD0 of VDD2_COM Domain Enables PFD0 clock of VDD2_COM domain. 0b - Disable 1b - Enable
4 PFD0_OF_MDN	PFD0 of VDDN_MEDIA Domain Enables PFD0 clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
3 PFD0_OF_MD2	PFD0 of VDD2_MEDIA Domain Enables PFD0 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
2 PFD0_OF_VDD2_DSP	PFD0 of VDD2_DSP Domain Enables PFD0 clock of VDD2_DSP domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
1 PFD0_OF_VDD1_SENSE	PFD0 of VDD1_SENSE Domain Enables PFD0 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
0 PFD0_OF_CMPT	PFD0 of VDD2_COMP Domain Enables PFD0 clock of VDD2_COMP domain. 0b - Disable 1b - Enable

28.7.3.19 Audio PLL0 Clock Source Select (AUDIOPLL0CLKSEL)

Offset

Register	Offset
AUDIOPLL0CLKSEL	400h

Function
Selects a clock source for audio PLL0 clock.

The Audio PLL0 output frequency is controlled by the programmed values of MULT, NUM and DENOM and can be expressed as a function of the AUDIOPLL0 reference input frequency. Use this AUDIOPLL0CLKSEL register to select the AUDIOPLL0 reference input frequency.

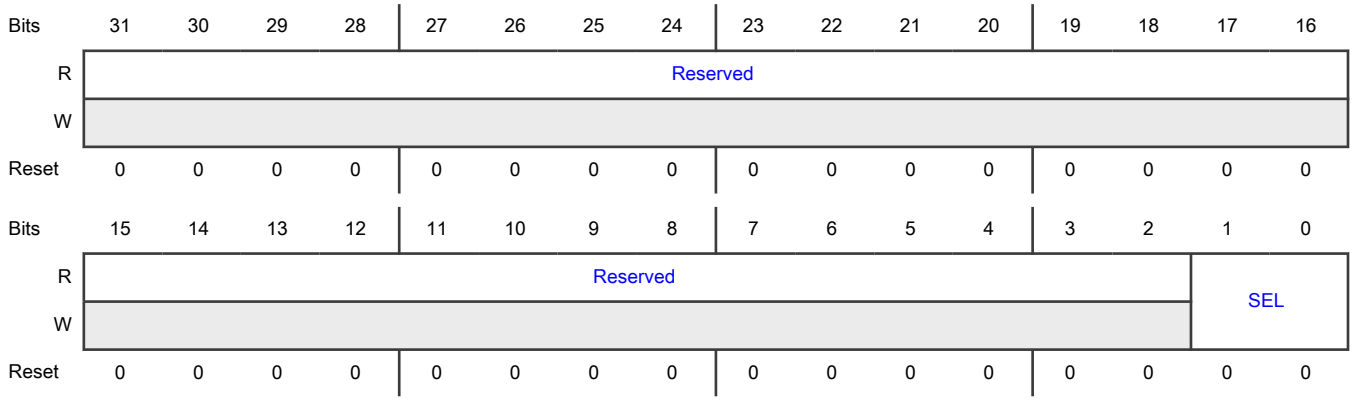
$F_{MAINPLL0} = F_{PLLREFIN} * (MULT + NUM/DENOM)$

After an input frequency has been chosen, MULT must be chosen so that the AUDIOPLL0 output frequency is 528 MHz. Therefore, if choosing an AUDIOPLL0 reference input frequency of 24 MHz, then program MULT as (0x16) or decimal 22, because 24 MHz * 22 = 528 MHz. Fine-tuning of the frequency can be done using NUM and DENOM.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Audio PLL0 Clock Source Select Select source for audio PLL0 reference input clock. **NOTE** The PLL reference input frequency is limited to the following discrete frequencies: 32, 30, 26, 24, 19.2, 16 MHz. 00b - fro1_div8 01b - osc_clk (User-Selectable) 10b - Reserved 11b - Reserved

28.7.3.20 Audio PLL0 Control 0 (AUDIOPLL0CTL0)

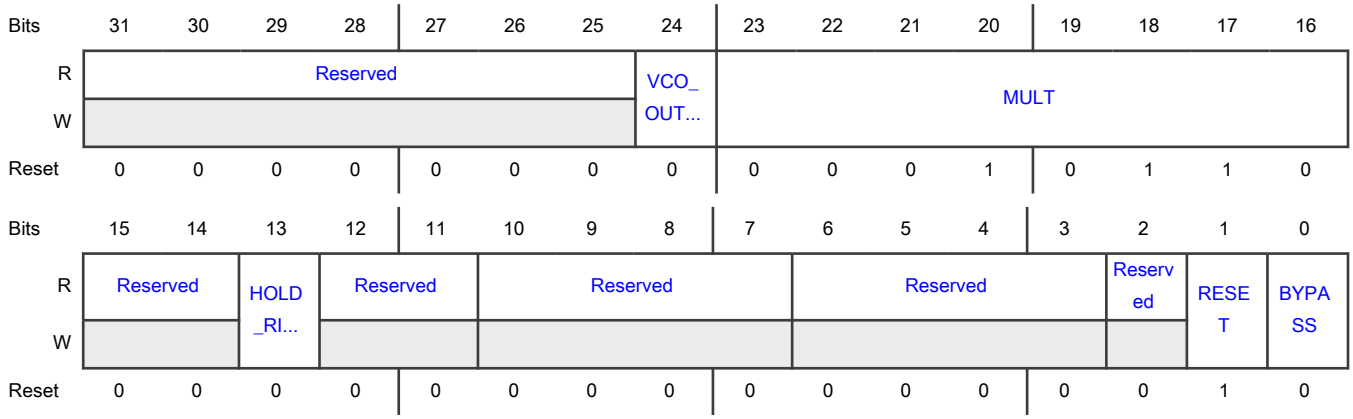
Offset

Register	Offset
AUDIOPLL0CTL0	404h

Function

Enables control for Audio PLL0.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 VCO_OUT_ENABLE	VCO Output Enable Enables VCO output. Each time after PLL power up, software needs to wait the whole PLL lock time the to enable VCO output . 0b - Disable 1b - Enable
23-16 MULT	Multiplication Factor Sets the multiplication factor for $F_{AUDIOPLL0_OUTPUT}$ to one of the following values. The value of MULT is directly decoded. 0001_0000b - Divided by 16 0001_0001b - Divided by 17 0001_0010b - Divided by 18 0001_0011b - Divided by 19 0001_0100b - Divided by 20 0001_0101b - Divided by 21 0001_0110b - Divided by 22
15-14 —	Reserved
13 HOLD_RING_OFF_ENA	Hold Ring Off Control Enables hold ring off to avoid multi-wave problems within VCO.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Assert HOLD_RING_OFF_ENA bit in the 1st half of lock time, and de-assert HOLD_RING_OFF_ENA bit in the 2nd half of lock time. 0b - Disable 1b - Enable
12-11 —	Reserved
10-7 —	Reserved
6-3 —	Reserved
2 —	Reserved
1 RESET	AUDIOPLL0 Reset 0b - Audio PLL0 reset is removed. 1b - Audio PLL0 is placed into reset.
0 BYPASS	Bypass Mode Enables bypass mode. 0b - PFD outputs are PFD-programmed clocks. 1b - Bypass mode. PFD outputs are sourced directly from the reference input clock.

28.7.3.21 Audio PLL0 Lock Time Divide-by-2 (AUDIOPLL0LOCKTIMEDIV2)

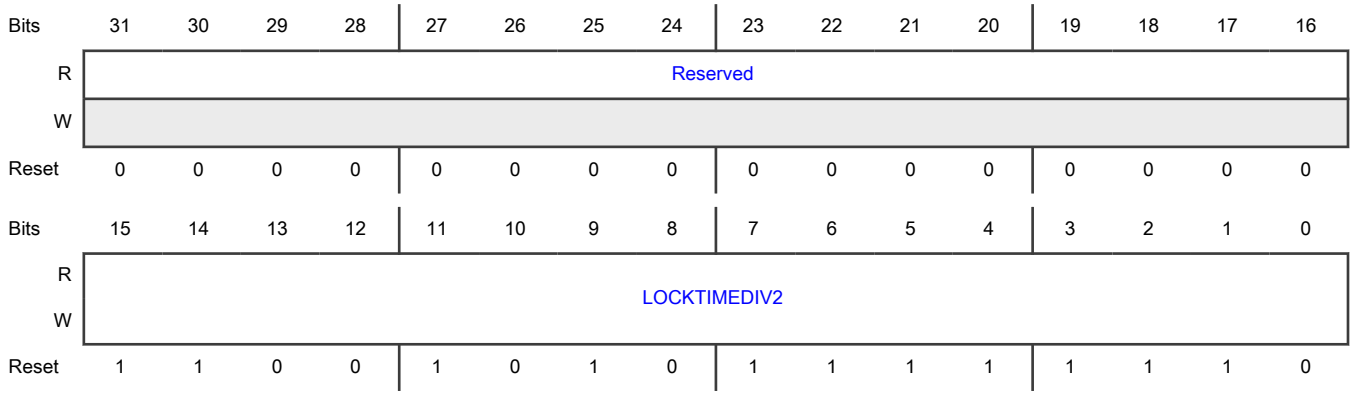
Offset

Register	Offset
AUDIOPLL0LOCKTIMEDIV2	40Ch

Function

When reset, the BOOTROM copies the AUDIOPLL0 worst case characterized lock time value (found in the OTP fuses) into the respective fields (if the OTP valid signature is present) for software to control the HOLD_RING_OFF bit in the AUDIOPLL0CTL0 register, and get the information on when half the lock time has expired and when the full lock has passed. Any PLL lock sequence requires HOLD_RING_OFF to be enabled (') during the first half of the lock time, and HOLD_RING_OFF to be disabled (0' during the second half of the lock time.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 LOCKTIMEDIV2	AUDIOPLL0 Lock Time Divide-by-2 The programmed lock time is in us (microseconds) and is programmed as half the actual lock time value.

28.7.3.22 Audio PLL0 Numerator (AUDIOPLL0NUM)

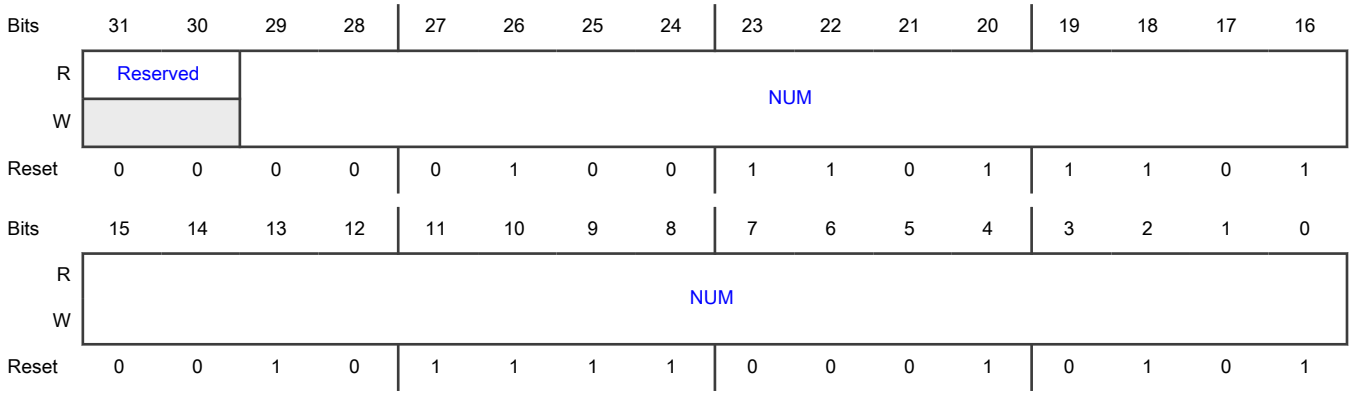
Offset

Register	Offset
AUDIOPLL0NUM	410h

Function

Provides the numerator for the audio PLL fractional divider. If the PLL numerator and denominator are set up and subsequently altered while the PLL is running, the amount of change on a single step of the NUM should be limited to 100 ppm of the current setting.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-0 NUM	Numerator Sets the numerator of the Audio PLL0 fractional loop divider. • The value of numerator must always be configured to be less than the value of the denominator. • The AUDIOPLL0NUM register can only be changed when the Audio PLL0 is disabled.

28.7.3.23 Audio PLL0 Denominator (AUDIOPLL0DENOM)

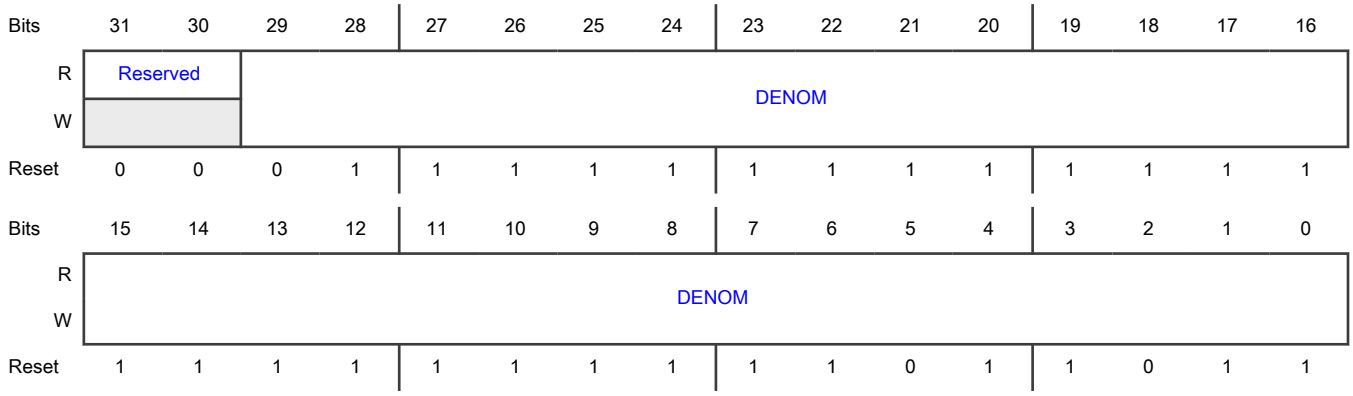
Offset

Register	Offset
AUDIOPLL0DENOM	414h

Function

Provides the denominator for the audio PLL fractional divider.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-0 DENOM	Denominator Sets the denominator of the Audio PLL0 fractional loop divider. • The value of denominator must always be configured to be greater than the value of the numerator. • The AUDIOPLL0DENOM register can only be changed when the Audio PLL0 is disabled.

28.7.3.24 Audio PLL0 PFD (AUDIOPLL0PFD)

Offset

Register	Offset
AUDIOPLL0PFD	418h

Function

Defines the control bits for the PFD0 to PFD3 clocks derived from the Audio PLL0.

The PFD output frequencies are governed by the following expression:

$F_{PFDx} = F_{AUDIOPLL0_OUT} * (18 / PFDx)$

To change PFD values, follow these steps:

1. Ensure that the PFDx clock gets gated by writing 1 to the PFDx_CLKGATE bit.
2. Program the new PFD value limited to 12 to 35 (in decimal).
3. Write 0 to the PFDx_CLKGATE register to ungate the PFDx clock.
4. Poll the PFDx_CLKRDY status flag until it (the status flag) is set to 1.
5. Clear the PFDx_CLKRDY status flag by writing 1 to that specific PFDx_CLKRDY bit.

NOTE

In non-bypass mode, the pfd output should be disabled by the PFDn_CLKGATE bit, until PLL lock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PFD3_CL...	PFD3_CL...	PFD3						PFD2_CL...	PFD2_CL...	PFD2					
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PFD1_CL...	PFD1_CL...	PFD1						PFD0_CL...	PFD0_CL...	PFD0					
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Function
31 PFD3_CLKGATE	PFD3 Clock Gate Enables PFD3 clock gate. 0b - Disable 1b - Enable
30 PFD3_CLKRDY	PFD3 Clock Ready Status Flag Indicates the ready status of PFD3 clock. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - Not ready 1b - Ready
29-24 PFD3	PLL Fractional Divider 3 Sets the PLL fractional divider 3 value in the range of 12 to 35 (in decimal).
23 PFD2_CLKGATE	PFD2 Clock Gate Enables PFD2 clock gate. 0b - Disable 1b - Enable
22 PFD2_CLKRDY	PFD2 Clock Ready Status Flag Indicates the ready status of PFD2 clock. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not ready 1b - Ready
21-16 PFD2	PLL Fractional Divider 2 Sets the PLL fractional divider 2 value in the range of 12 to 35 (in decimal).
15 PFD1_CLKGATE	PFD1 Clock Gate Enables PFD1 clock gate. 0b - Disable 1b - Enable
14 PFD1_CLKRDY	PFD1 Clock Ready Status Flag Indicates the ready status of PFD1 clock. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - Not ready 1b - Ready
13-8 PFD1	PLL Fractional Divider 1 Sets the PLL fractional divider 1 value in the range of 12 to 35 (in decimal).
7 PFD0_CLKGATE	PFD0 Clock Gate Enables PFD0 clock gate. 0b - Disable 1b - Enable
6 PFD0_CLKRDY	PFD0 Clock Ready Status Flag Indicates the ready status of PFD0 clock. Writing both 1 or 0 to this bit will clear this status. The recommend sequence here is write this bit, then program the new PFD value. After this bit is set to 1, the phase divider clock output is stable. 0b - Not ready 1b - Ready
5-0 PFD0	PLL Fractional Divider 0 Sets the PLL fractional divider value in the range of 12 to 35 (in decimal).

28.7.3.25 Audio PLL0 PFD Clock Enable (AUDIOPLL0PFDDOMAINEN)

Offset

Register	Offset
AUDIOPLL0PFDDOMAINEN	420h

Function

Each PFD clock of Audio PLL0 can be sent to various domains to reduce power consumption when this PFD is not set to be used in such domain. The PFD can be disabled for such domain.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	PFD3_OF...	Reserved	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...	PFD2_OF...
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	PFD1_OF...	Reserved	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...	PFD0_OF...
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Fields

Field	Function
31 —	Reserved
30 PFD3_OF_COMN	PFD3 of VDDN_COM Domain Enables PFD3 clock of VDDN_COM domain. 0b - Disable 1b - Enable
29 PFD3_OF_VDD2_COM	PFD3 of VDD2_COM Domain Enables PFD3 clock of VDD2_COM domain. 0b - Disable 1b - Enable
28	PFD3 of VDDN_MEDIA Domain Enables PFD3 clock of VDDN_MEDIA domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
PFD3_OF_MDN	0b - Disable 1b - Enable
27 PFD3_OF_MD2	PFD3 of VDD2_MEDIA Domain Enables PFD3 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
26 PFD3_OF_VDD2_DSP	PFD3 of VDD2_DSP Domain Enables PFD3 clock of VDD2_DSP domain. 0b - Disable 1b - Enable
25 PFD3_OF_VDD1_SENSE	PFD3 of VDD1_SENSE Domain Enables PFD3 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
24 PFD3_OF_CMPT	PFD3 of VDD2_COMP Domain Enables PFD3 clock of VDD2_COMP domain. 0b - Disable 1b - Enable
23 —	Reserved
22 PFD2_OF_COMN	PFD2 of VDDN_COM Domain Enables PFD2 clock of VDDN_COM domain. 0b - Disable 1b - Enable
21 PFD2_OF_VDD2_COM	PFD2 of VDD2_COM Domain Enables PFD2 clock of VDD2_COM domain. 0b - Disable 1b - Enable
20 PFD2_OF_MDN	PFD2 of VDDN_MEDIA Domain Enables PFD2 clock of VDDN_MEDIA domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
19 PFD2_OF_MD2	PFD2 of VDD2_MEDIA Domain Enables PFD2 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
18 PFD2_OF_VDD2_DSP	PFD2 of VDD2_DSP Domain Enables PFD2 clock of VDD2_DSP domain. 0b - Disable 1b - Enable
17 PFD2_OF_VDD1_SENSE	PFD2 of VDD1_SENSE Domain Enables PFD2 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
16 PFD2_OF_CMPT	PFD2 of VDD2_COMP Domain Enables PFD2 clock of VDD2_COMP domain. 0b - Disable 1b - Enable
15 —	Reserved
14 PFD1_OF_COMN	PFD1 of VDDN_COM Domain Enables PFD1 clock of VDDN_COM domain. 0b - Disable 1b - Enable
13 PFD1_OF_VDD2_COM	PFD1 of VDD2_COM Domain Enables PFD1 clock of VDD2_COM domain. 0b - Disable 1b - Enable
12 PFD1_OF_MDN	PFD1 of VDDN_MEDIA Domain Enables PFD1 clock of VDDN_MEDIA domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
11 PFD1_OF_MD2	PFD1 of VDD2_MEDIA Domain Enables PFD1 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
10 PFD1_OF_VDD2_DSP	PFD1 of VDD2_DSP Domain Enables PFD1 clock of VDD2_DSP domain 0b - Disable 1b - Enable
9 PFD1_OF_VDD1_SENSE	PFD1 of VDD1_SENSE Domain Enables PFD0 clock of VDD1_SENSE domain 0b - Disable 1b - Enable
8 PFD1_OF_CMPT	PFD1 of VDD2_COMP Domain Enables PFD1 clock of VDD2_COMP domain 0b - Disable 1b - Enable
7 —	Reserved
6 PFD0_OF_COMN	PFD0 of VDDN_COM Domain Enables PFD0 clock of VDDN_COM domain. 0b - Disable 1b - Enable
5 PFD0_OF_VDD2_COM	PFD0 of VDD2_COM Domain Enables PFD0 clock of VDD2_COM domain. 0b - Disable 1b - Enable
4 PFD0_OF_MDN	PFD0 of VDDN_MEDIA Domain Enables PFD0 clock of VDDN_MEDIA domain.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
3 PFD0_OF_MD2	PFD0 of VDD2_MEDIA Domain Enables PFD0 clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
2 PFD0_OF_VDD2_DSP	PFD0 of VDD2_DSP Domain Enables PFD0 clock of VDD2_DSP domain. 0b - Disable 1b - Enable
1 PFD0_OF_VDD1_SENSE	PFD0 of VDD1_SENSE Domain Enables PFD0 clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
0 PFD0_OF_CMPT	PFD0 of VDD2_COMP Domain Enables PFD0 clock of VDD2_COMP domain. 0b - Disable 1b - Enable

28.7.3.26 Audio PLL0 VCO Clock Enable (AUDIOPLL0VCODOMAINEN)

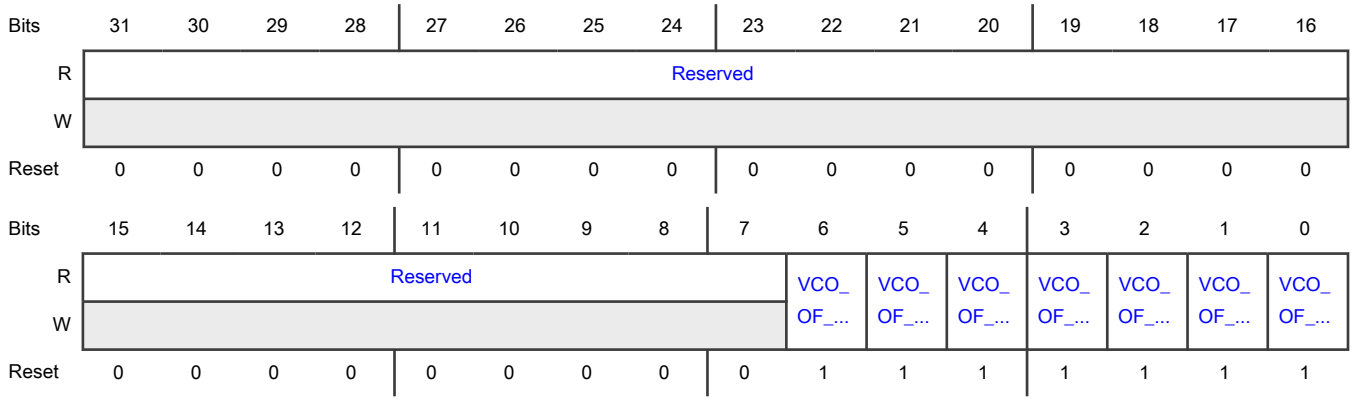
Offset

Register	Offset
AUDIOPLL0VCODOMAINEN	424h

Function

VCO clock of Audio PLL0 can be sent to various domains to reduce power consumption when VCO is not set to use in such domain. This VCO can be disabled for such domain.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 VCO_OF_COMN	VCO of VDDN_COM Domain Enables VCO clock of VDDN_COM domain. 0b - Disable 1b - Enable
5 VCO_OF_VDD2_COM	VCO of VDD2_COM Domain Enables VCO clock of VDD2_COM domain. 0b - Disable 1b - Enable
4 VCO_OF_MDN	VCO of VDDN_MEDIA Domain Enables VCO clock of VDDN_MEDIA domain. 0b - Disable 1b - Enable
3 VCO_OF_MD2	VCO of VDD2_MEDIA Domain Enables VCO clock of VDD2_MEDIA domain. 0b - Disable 1b - Enable
2 VCO_OF_VDD2_DSP	VCO of VDD2_DSP Domain Enables VCO clock of VDD2_DSP domain. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 VCO_OF_VDD1_SENSE	VCO of VDD1_SENSE Domain Enables VCO clock of VDD1_SENSE domain. 0b - Disable 1b - Enable
0 VCO_OF_CMPT	VCO of VDD2_COMP Domain Enables VCO clock of VDD2_COMP domain. 0b - Disable 1b - Enable

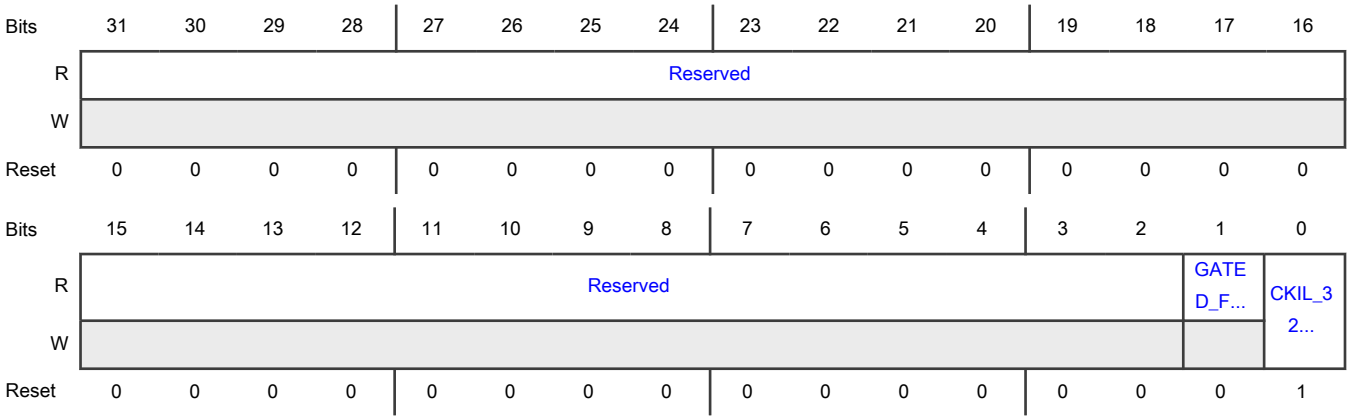
28.7.3.27 CKIL 32kHz Clock Gate (CKIL_32K_GATE)

Offset

Register	Offset
CKIL_32K_GATE	600h

Function
Gates the CKIL 32kHz clock.

Diagram



Fields

Field	Function
31-2 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 GATED_FLAG	Gated Flag Indicates the gating status of CKIL 32kHz clock. 0b - The CKIL 32 kHz clock is not gated. 1b - The CKIL 32 kHz clock is gated.
0 CKIL_32K_EN	CKIL 32 kHz Clock Enable Enables CKIL 32 kHz clock. 0b - Gates CKIL 32 kHz clock. You need to read GATED_FLAG field with the value of 1b to confirm the clock is already gated before going ahead. 1b - Enables CKIL 32 kHz clock.

28.7.4 CLKCTL3 register descriptions

This section includes the memory map and register descriptions for Clock Controller 3.

The peripherals in CLKCTL3 are shared for both CPU0 and CPU1. There is a common scenario where CPU0 wants to turn off one clock, but CPU1 is still using this clock. To handle such kind of scenario, two sets of registers are configured in CLKCTL3 named as PSCCTLn_COMP (these registers can only be written by CPU0, but they can be read by both CPU0 and CPU1.) and PSCCTLn_SENS (these registers can only be written by CPU1, but they can be read by both CPU1 and CPU0.). In this way, if CPU0 does not need such peripheral work, you can disable it by writing PSCCTLn_COMP. It also can read PSCCTLn_SENS back to check whether this clock is really disabled.

28.7.4.1 CLKCTL3 memory map

CLKCTL3 base address: 4006_1000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0_COMP)	32	RW	00F0_2000h
40h	VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_COMP_SET)	32	RW	0000_0000h
70h	VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_COMP_CLR)	32	RW	0000_0000h
90h	One Source Clock Slice Enable for VDD2_COMP CPU (ONE_SRC_CLKSLICE_ENABLE_COMP)	32	RW	0000_0000h
210h	Low Power Oscillator Control 0 (LPOSCCTL0)	32	R	807B_C4D4h
214h	VDD1_SENSE Clock Source (SENSEBASECLKSEL)	32	R	0000_0000h
290h	FRO_TUNER2 Clock Status (FRO2CLKSTATUS)	32	R	0000_0000h
298h	FRO2MAX Clock Domain Enable (FRO2MAXDOMAINEN)	32	RW	0000_007Fh
400h	VDD1_SENSE Main Clock Divider (SENSEMAINCLKDIV)	32	RW	0000_0001h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
434h	VDD1_SENSE Main Clock Source Select (MAINCLKSEL)	32	RW	0000_0000h
450h	VDD1_SENSE RAM Clock Source Select (SENSERAMCLKSEL)	32	RW	0000_0000h
45Ch	VDD1_SENSE RAM Clock Divider (SENSERAMCLKDIV)	32	RW	0000_0001h
480h	OSTIMER Functional Clock Source Select (OSTIMERFCLKSEL)	32	RW	0000_0000h
484h	OSTIMER Functional Clock Divider (OSTIMERFCLKDIV)	32	RW	4000_0001h
600h	SDADC Functional Clock Source Select (SDADCFCLKSEL)	32	RW	0000_0000h
604h	SDADC Functional Clock Divider (SDADCFCLKDIV)	32	RW	4000_0001h
620h	ADC0 (SARADC) Functional Clock Source Select (SARADCFCLKSEL)	32	RW	0000_0000h
624h	ADC0 (SARADC) Functional Clock Divider (SARADCFCLKDIV)	32	RW	4000_0001h
750h	Wake 32 kHZ Clock Source Select (WAKE32KCLKSEL)	32	RW	0000_0001h
780h	MICFIL Functional Clock Source Select (MICFILFCLKSEL)	32	RW	0000_0004h
784h	MICFIL Functional Clock Divider (MICFILFCLKDIV)	32	RW	4000_0001h
788h	LPI2C15 Functional Clock Source Select (LPI2C15FCLKSEL)	32	RW	0000_0000h
78Ch	LPI2C15 Functional Clock Divider (LPI2C15FCLKDIV)	32	RW	4000_0001h
800h	CLKOUT_VDD1 Clock Source Select (CLKOUTCLKSEL)	32	RW	0000_0000h
804h	CLKOUT_VDD1 Clock Divider (CLKOUTCLKDIV)	32	RW	4000_0001h
810h	VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0_SENS)	32	RW	00F0_2000h
840h	VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_SENS_SET)	32	RW	0000_0000h
870h	VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_SENS_CLR)	32	RW	0000_0000h
890h	One Source Clock Slice Enable for VDD1_SENSE CPU (ONE_SRC_CLKSLICE_ENABLE_SENSE)	32	RW	0000_0000h

28.7.4.2 VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0_COMP)

Offset

Register	Offset
PSCCTL0_COMP	10h

Function

Allows software to enable specific peripherals. Writing a zero to any assigned bit in this register gates the clock. Writing an one enables the clock and allows the specified peripheral to operate. If VDD1_SENSE power domain is power OFF, reset after power ON.

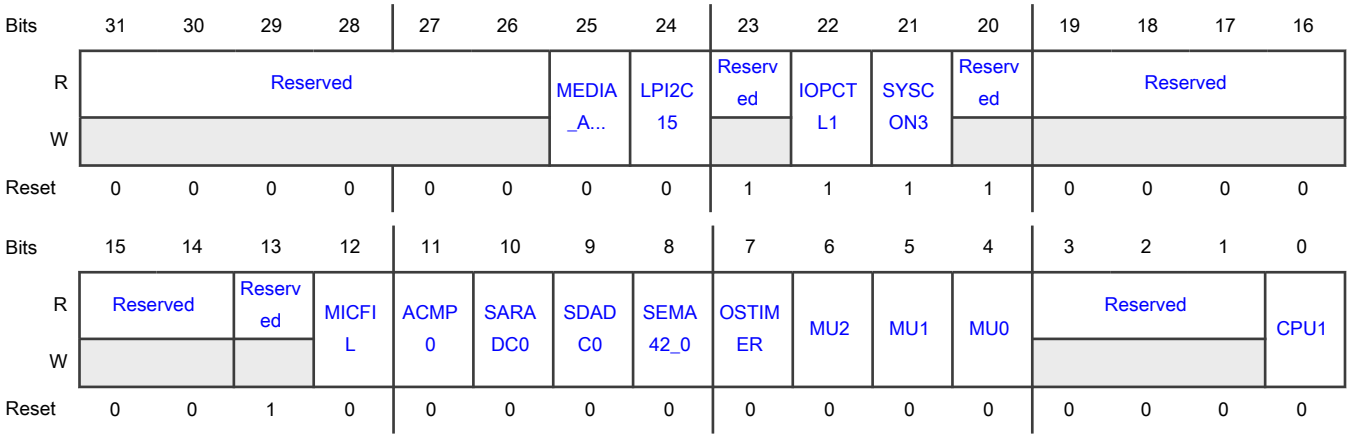
NOTE

This register can only be written by CPU0. CPU1 can not write this register. However, CPU1 can read it to query the really status of device's clock. CPU0 can write zero to PSCCTL0_COMP to disable the clock. But whether this clock is really disabled, software need to read back the related bit from PSCCTL0_SENS, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACCESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Enables media access RAM arbiter1 clock. 0b - Disable 1b - Enable
24 LPI2C15	LPI2C15 Clock Enables LPI2C15 clock. 0b - Disable 1b - Enable
23 —	Reserved
22 IOPCTL1	IOPCTL1 Clock Enables IOPCTL1 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
21 SYSCON3	SYSCON3 Clock Enables SYSCON3 clock. 0b - Disable 1b - Enable
20 —	Reserved
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Enables MICFIL clock. 0b - Disable 1b - Enable
11 ACMP0	ACMP0 Clock Enables ACMP0 clock. 0b - Disable 1b - Enable
10 SARADC0	ADC0 Clock Enables ADC0 clock. 0b - Disable 1b - Enable
9 SDADC0	SDADC0 Clock Enables SDADC0 clock. 0b - Disable 1b - Enable
8 SEMA42_0	SEMA42_0 Clock Enables SEMA42_0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
7 OSTIMER	OSTIMER Clock Enables OSTIMER clock. 0b - Disable 1b - Enable
6 MU2	MU2 Clock Enables MU2 clock. 0b - Disable 1b - Enable
5 MU1	MU1 Clock Enables MU1 clock. 0b - Disable 1b - Enable
4 MU0	MU0 Clock Enables MU0 clock. 0b - Disable 1b - Enable
3-1 —	Reserved
0 CPU1	CPU1 Clock Enables CPU1 clock. 0b - Disable 1b - Enable

28.7.4.3 VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_COMP_SET)

Offset

Register	Offset
PSCCTL0_COMP_SET	40h

Function

Writing an one to a bit position in PSCCTL0_SET sets the corresponding position in PSCCTL0.

NOTE

This register can only be written by CPU0, CPU1 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						MEDIA_A...	LPI2C15	Reserved	IOPCTL1	SYSCON3	Reserved	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Reserved	MICFIL	ACMP0	SARADC0	SDADC0	SEMA42_0	OSTIMER	MU2	MU1	MU0	Reserved			CPU1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACCESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Enables media access RAM arbiter1 clock. 0b - No effect 1b - Enable
24 LPI2C15	LPI2C15 Clock Enables LPI2C15 clock. 0b - No effect 1b - Enable
23 —	Reserved
22 IOPCTL1	IOPCTL1 Clock Enables IOPCTL1 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
21 SYSCON3	SYSCON3 Clock Enables SYSCON3 clock. 0b - No effect 1b - Enable
20 —	Reserved
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Enables MICFIL clock. 0b - No effect 1b - Enable
11 ACMP0	ACMP0 Clock Enables ACMP0 clock. 0b - No effect 1b - Enable
10 SARADC0	ADC0 Clock Enables ADC0 clock. 0b - No effect 1b - Enable
9 SDADC0	SDADC0 Clock Enables SDADC0 clock. 0b - No effect 1b - Enable
8 SEMA42_0	SEMA42_0 Clock Enables SEMA42_0 clock. 0b - No effect 1b - Enable
7	OSTIMER Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
OSTIMER	Enables OSTIMER clock. 0b - No effect 1b - Enable
6 MU2	MU2 Clock Enables MU2 clock. 0b - No effect 1b - Enable
5 MU1	MU1 Clock Enables MU1 clock. 0b - No effect 1b - Enable
4 MU0	MU0 Clock Enables MU0 clock. 0b - No effect 1b - Enable
3-1 —	Reserved
0 CPU1	CPU1 Clock Enables CPU1 clock. 0b - No effect 1b - Enable

28.7.4.4 VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_COMP_CLR)

Offset

Register	Offset
PSCCTL0_COMP_CLR	70h

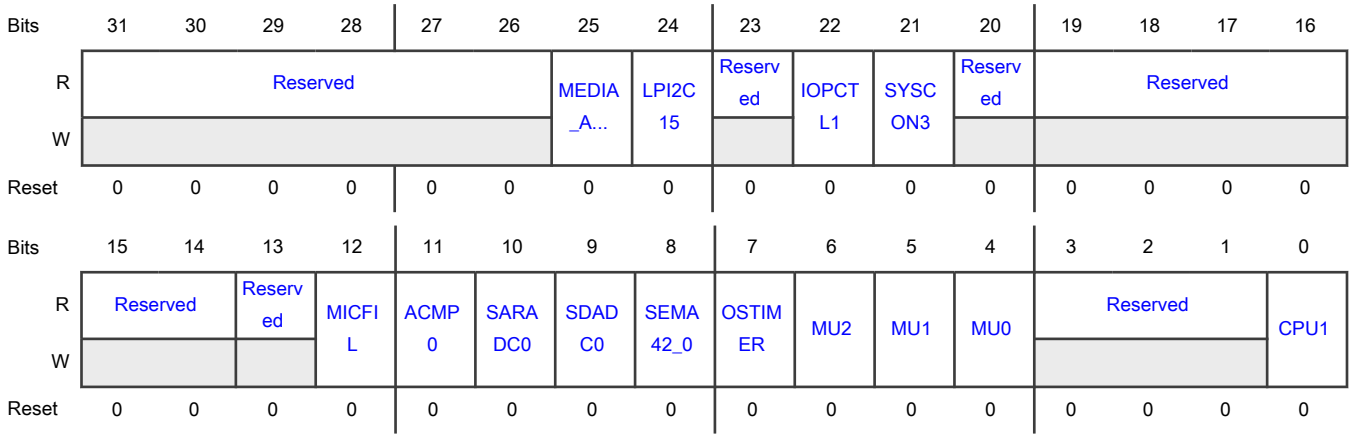
Function

Writing an one to a bit position in PSCCTL0_CLR clears the corresponding position in PSCCTL0.

NOTE

This register can only be written by CPU0. CPU1 can not write this register. CPU0 can write 1 to PSCCTL0_COMP_CLR to disable the clock. But whether this clock is really disabled, software need to read back the related bit from PSCCTL0_SENS, since if one core enables the clock, it cannot be disabled by another core.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACCESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Disables media access RAM arbiter1 clock. 0b - No effect 1b - Disable
24 LPI2C15	LPI2C15 Clock Disables LPI2C15 clock. 0b - No effect 1b - Disable
23 —	Reserved
22 IOPCTL1	IOPCTL1 Clock Disables IOPCTL1 clock. 0b - No effect 1b - Disable
21 SYSCON3	SYSCON3 Clock Disables SYSCON3 clock. 0b - No effect 1b - Disable
20	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Disables MICFIL clock. 0b - No effect 1b - Disable
11 ACMP0	ACMP0 Clock Disables ACMP0 clock. 0b - No effect 1b - Disable
10 SARADC0	ADC0 Clock DisablesADC0 clock. 0b - No effect 1b - Disable
9 SDADC0	SDADC0 Clock Disables SDADC0 clock. 0b - No effect 1b - Disable
8 SEMA42_0	SEMA42_0 Clock Disables SEMA42_0 clock. 0b - No effect 1b - Disable
7 OSTIMER	OSTIMER Clock Disables OSTIMER clock 0b - No effect 1b - Disable
6	MU2 Clock Disables MU2 clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
MU2	0b - No effect 1b - Disable
5 MU1	MU1 Clock Disables MU1 clock. 0b - No effect 1b - Disable
4 MU0	MU0 Clock Disables MU0 clock. 0b - No effect 1b - Disable
3-1 —	Reserved
0 CPU1	CPU1 Clock Disables CPU1 clock. 0b - No effect 1b - Disable

28.7.4.5 One Source Clock Slice Enable for VDD2_COMP CPU (ONE_SRC_CLKSLICE_ENABLE_COMP)

Offset

Register	Offset
ONE_SRC_CLKSLICE_ENABLE_COMP	90h

Function

Controls the gating of one source clock slice. Multi source clock slices gating is controlled by enabling the SEL_EN bit field of a series of clock select source registers, for example, XSPI*n*CLKSEL.

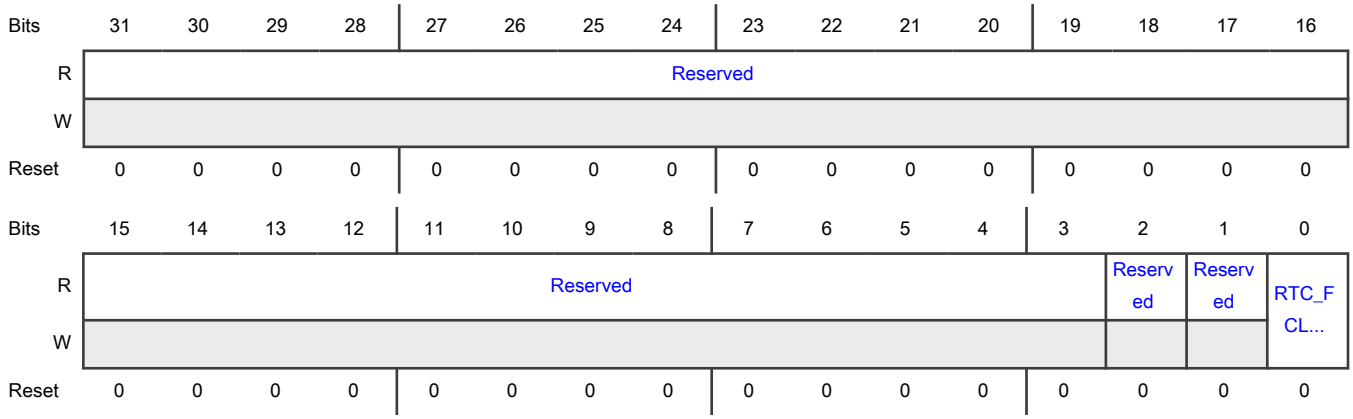
NOTE

Not all the clock select source registers have this SEL_EN bit for some clocks cannot be gated, such as base clock.

NOTE

This register is compute core access only.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 —	Reserved
1 —	Reserved
0 RTC_FCLK_EN	RTC Functional Clock Gating Enable Enables RTC functional clock. This clock's source is lposc_1m. Make sure lposc_1m is active before enabling the RTC functional clock. 0b - Gates RTC functional clock. 1b - Enables RTC functional clock.

28.7.4.6 Low Power Oscillator Control 0 (LPOSCCTL0)

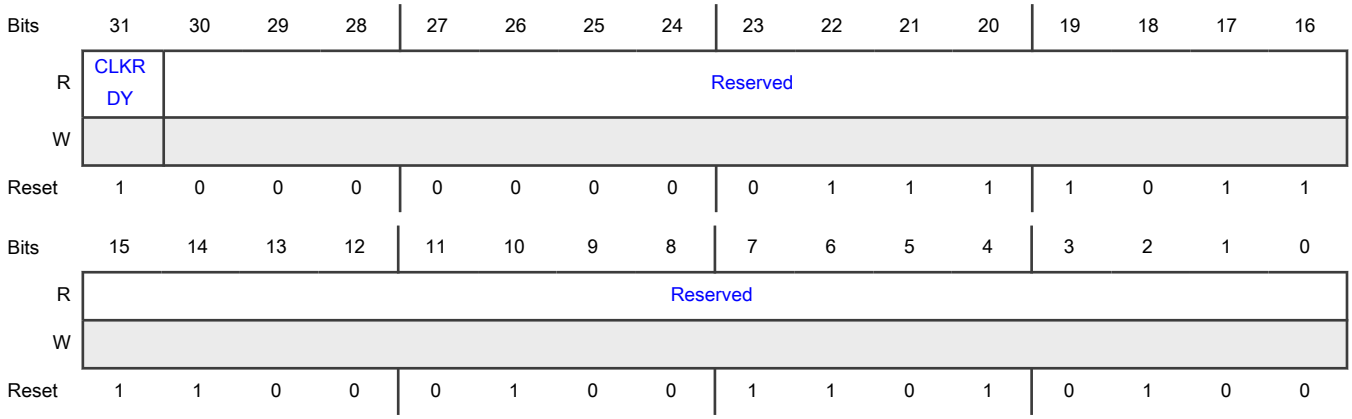
Offset

Register	Offset
LPOSCCTL0	210h

Function

Indicates the status of the 1 MHz low power oscillator (1m_lposc).

Diagram



Fields

Field	Function
31 CLKRDY	LPOSC Clock Ready Indicates the status of LPOSC clock. 64 microseconds are needed before the clock is ready. 0b - LPOSC clock is not ready. 1b - LPOSC clock is ready.
30-0 —	Reserved

28.7.4.7 VDD1_SENSE Clock Source (SENSEBASECLKSEL)

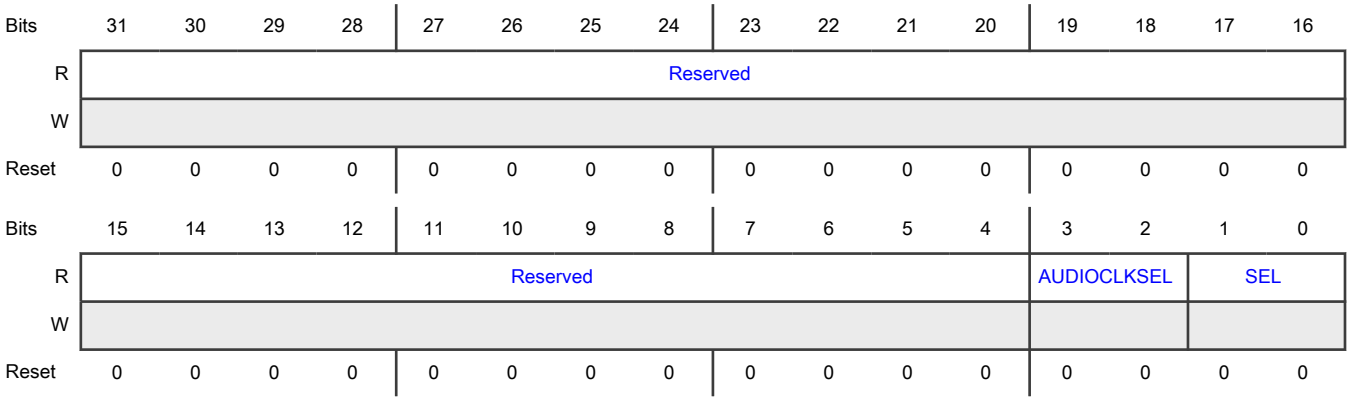
Offset

Register	Offset
SENSEBASECLKSEL	214h

Function

Indicates VDD1_SENSE base clock source and audio clock source. CPU0 can query this register.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-2 AUDIOCLKSEL	VDD1_SENSE Audio Clock Source Select Indicates the clock source of VDD1_SENSE audio clock. 00b - mclk_in 01b - osc_clk 10b - fro2_div8 11b - audio_pll_pfd3
1-0 SEL	VDD1_SENSE Base Clock Source Selection Indicates clock source of main VDD1_SENSE base clock. 00b - fro1_div3 01b - fro1_max 10b - fro2_div3 11b - 1m_lposc

28.7.4.8 FRO_TUNER2 Clock Status (FRO2CLKSTATUS)

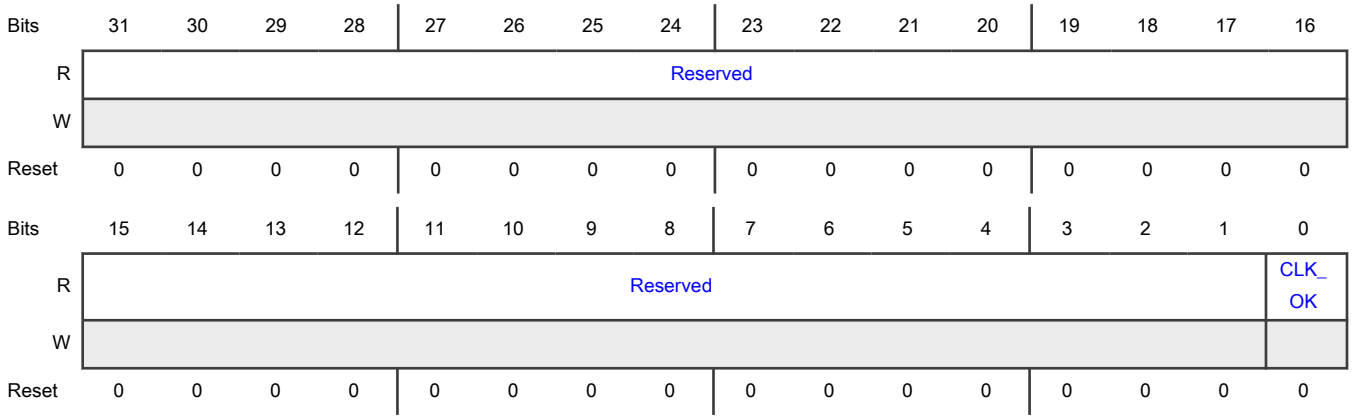
Offset

Register	Offset
FRO2CLKSTATUS	290h

Function

Indicates whether the FRO_TUNER2 clock has reached its final frequency.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CLK_OK	FRO_TUNER2 Clock OK Indicates the status when the Free-Running Oscillator (FRO_TUNER2) reaches its final frequency. 0b - FRO_TUNER2 clock has not yet reached its final frequency. 1b - FRO_TUNER2 clock has reached its final frequency.

28.7.4.9 FRO2MAX Clock Domain Enable (FRO2MAXDOMAINEN)

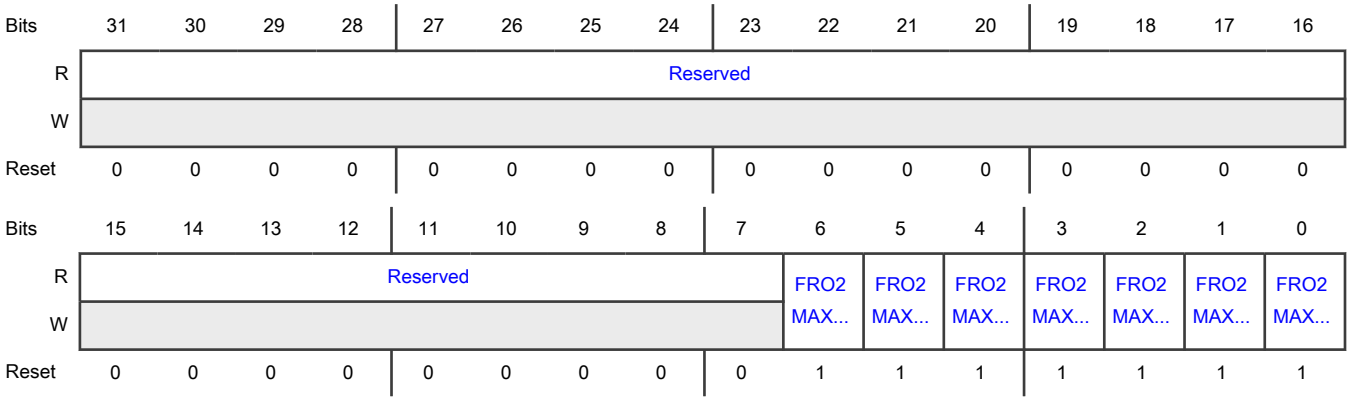
Offset

Register	Offset
FRO2MAXDOMAINEN	298h

Function

Sends fro2_max clock to various domains for application. To reduce power consumption, when software configuration does not use this clock in such domain, turn off this clock in such domain.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 FRO2MAX_OF_COMN	Enable fro2_max Clock Control of VDDN_COM Domain 0b - Disable 1b - Enable
5 FRO2MAX_OF_VDD2_COM	Enable fro2_max Clock Control of VDD2_COM Domain 0b - Disable 1b - Enable
4 FRO2MAX_OF_MDN	Enable fro2_max Clock Control of VDDN_MEDIA Domain 0b - Disable 1b - Enable
3 FRO2MAX_OF_MD2	Enable fro2_max Clock Control of VDD2_MEDIA Domain 0b - Disable 1b - Enable
2 FRO2MAX_OF_VDD2_DSP	Enable fro2_max Clock Control of VDD2_DSP Domain 0b - Disable 1b - Enable
1 FRO2MAX_OF_SENSE	Enable fro2_max Clock Control of VDD1_SENSE Domain 0b - Disable 1b - Enable
0 FRO2MAX_OF_CMPT	Enable fro2_max Clock Control of VDD2_COMP Domain 0b - Disable 1b - Enable

28.7.4.10 VDD1_SENSE Main Clock Divider (SENSEMAINCLKDIV)

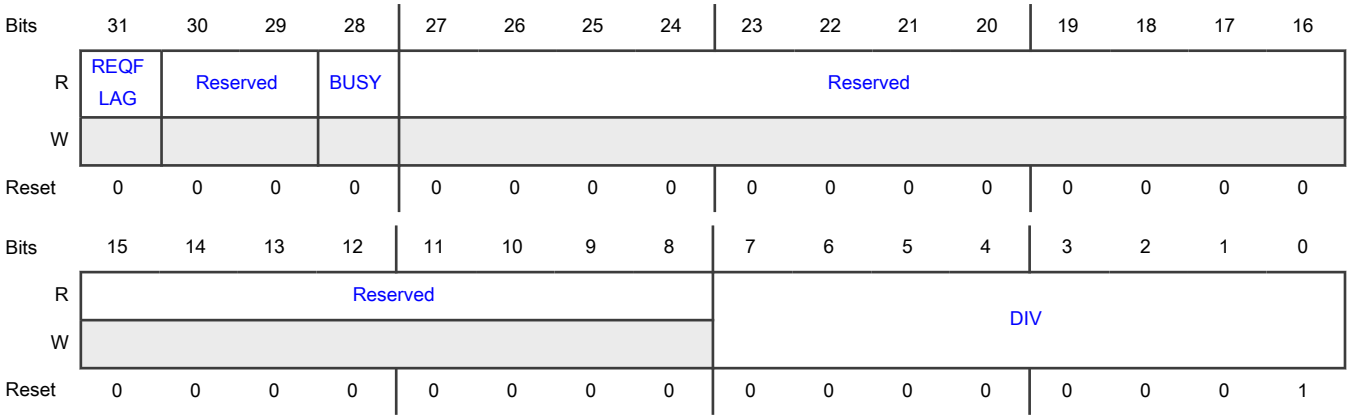
Offset

Register	Offset
SENSEMAINCLKDIV	400h

Function

Controls how the clock sources (sense_main_clk) of the main clock is divided to provide the system clock to the AHB bus, CPU, and memories.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.4.11 VDD1_SENSE Main Clock Source Select (MAINCLKSEL)

Offset

Register	Offset
MAINCLKSEL	434h

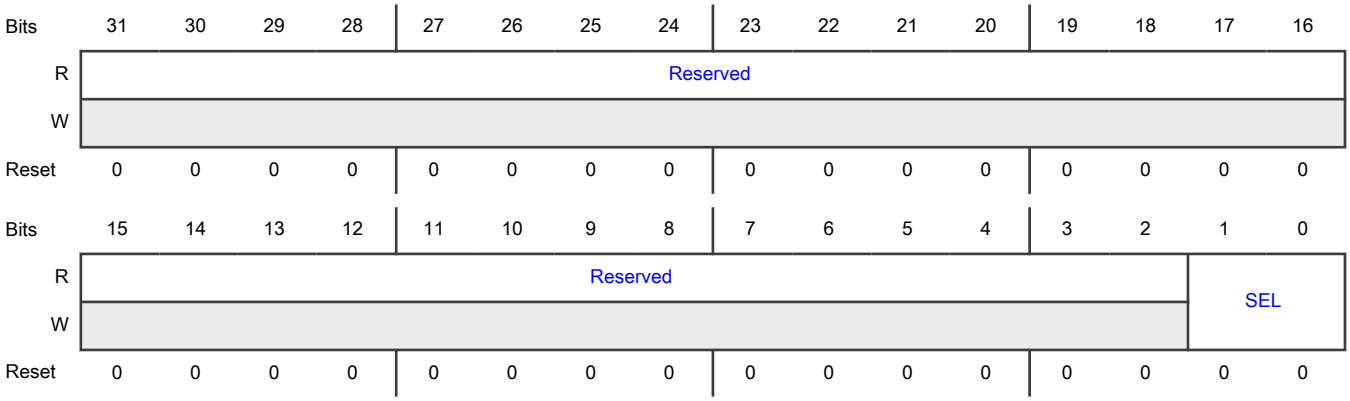
Function

Selects the clock source for VDD1_SENSE main clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
1-0 SEL	VDD1_SENSE Main Clock Source Select Selects the clock source for VDD1_SENSE main clock. 00b - baseclk_sense 01b - fro2_max 10b - audio_pll_pfd3 11b - fro1_max

28.7.4.12 VDD1_SENSE RAM Clock Source Select (SENSERAMCLKSEL)

Offset

Register	Offset
SENSERAMCLKSEL	450h

Function

Selects the clock source for RAM in VDD1_SENSE. This clock is used for async access.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
1-0 SEL	VDD1_SENSE RAM Clock Source Select Selects the clock source for VDD1_SENSE RAM clock. 00b - baseclk_sense 01b - fro2_max 10b - audio_pll_pfd2 11b - fro1_max

28.7.4.13 VDD1_SENSE RAM Clock Divider (SENSERAMCLKDIV)

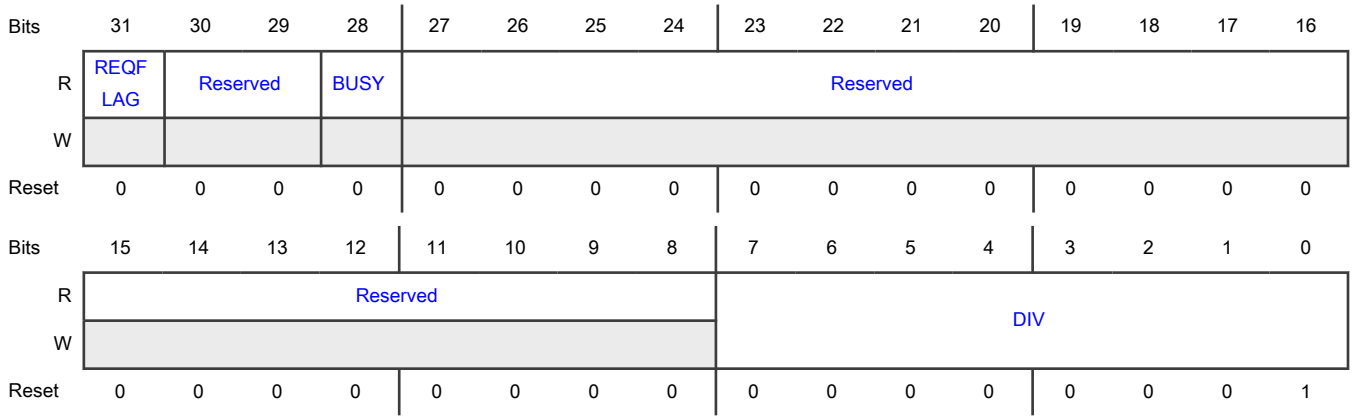
Offset

Register	Offset
SENSERAMCLKDIV	45Ch

Function

Controls the divider for the VDD1_SENSE RAM functional clock sense_ram_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

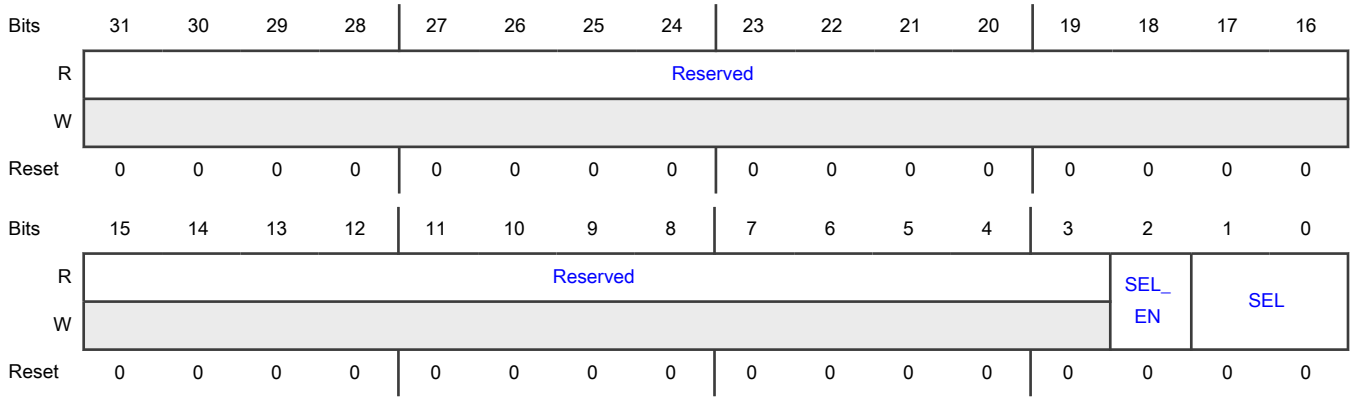
28.7.4.14 OSTIMER Functional Clock Source Select (OSTIMERFCLKSEL)

Offset

Register	Offset
OSTIMERFCLKSEL	480h

Function
Selects the clock source for the OSTIMER.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables the clock MUX output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects a clock source for OSTIMER. 00b - baseclk_sense 01b - wake32k_clk 10b - 1m_lposc 11b - osc_clk

28.7.4.15 OSTIMER Functional Clock Divider (OSTIMERFCLKDIV)

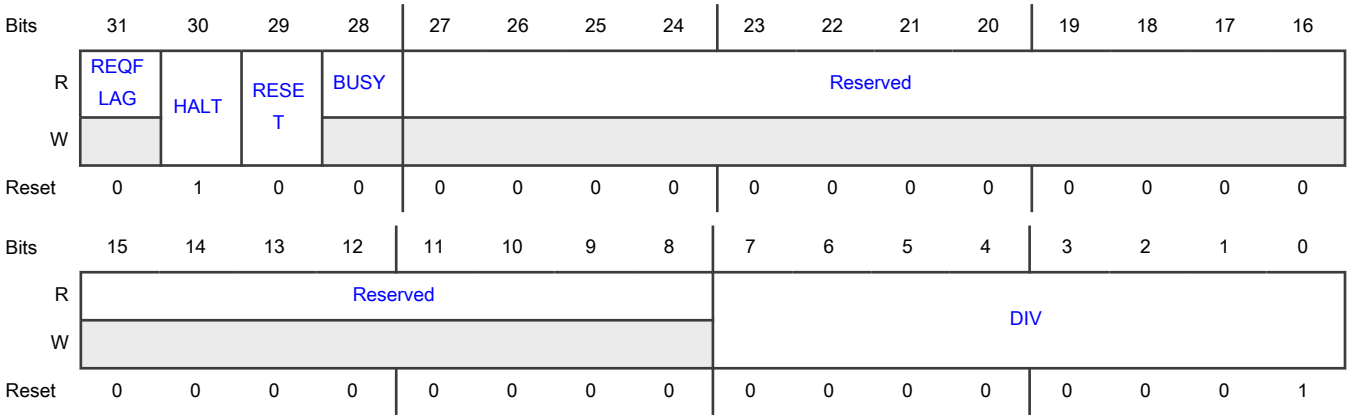
Offset

Register	Offset
OSTIMERFCLKDIV	484h

Function

Controls the divider for the OSTIMER functional clock osevent_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change. 0b - The divider change has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. The intent is to allow the divider clock source to be changed without the risk of a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
7-0 DIV	Clock Divider Value Select Selects a divider: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

28.7.4.16 SDADC Functional Clock Source Select (SDADCFCLKSEL)

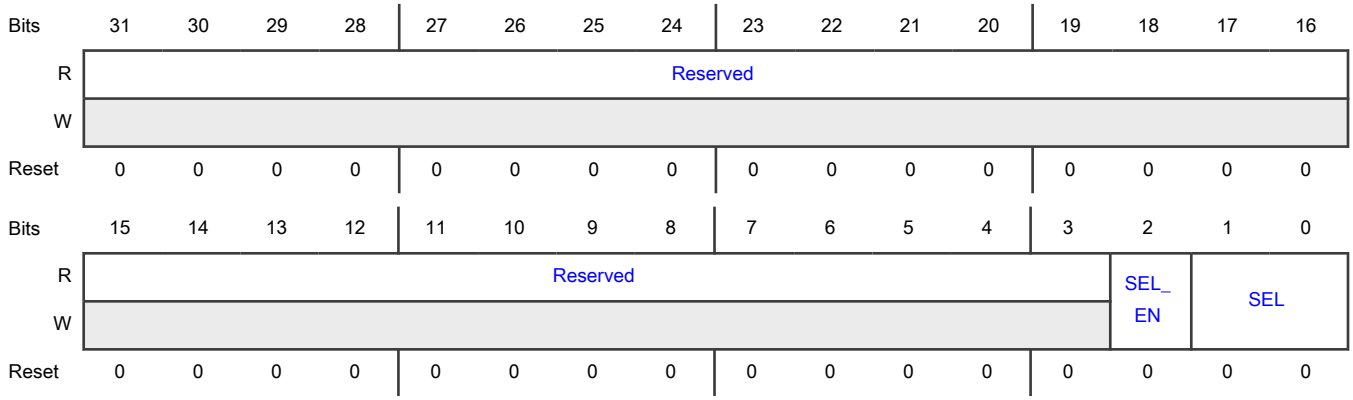
Offset

Register	Offset
SDADCFCLKSEL	600h

Function

Selects the clock source for SDADC functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2	Clock MUX Output Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEL_EN	Enables the clock mux output for SDADC functional clock to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	SDADC Functional Clock Source Select Selects the clock source for SARADC functional clock. 00b - baseclk_sense 01b - fro2_div8 10b - audio_pll_vco 11b - audio_clk

28.7.4.17 SDADC Functional Clock Divider (SDADCFCLKDIV)

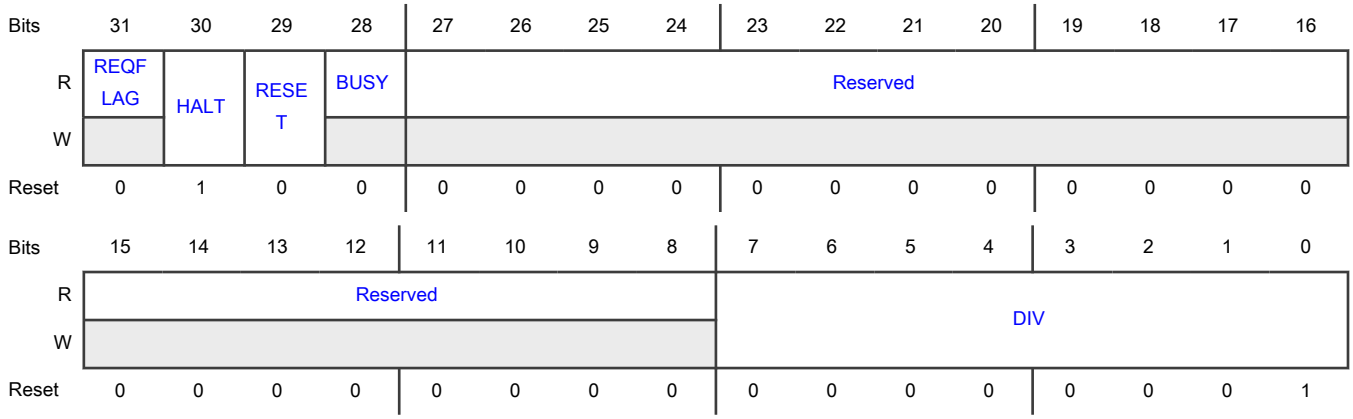
Offset

Register	Offset
SDADCFCLKDIV	604h

Function

Controls the divider for the SDADC function clock sdadc_fclk.

Diagram



Fields

Field	Function
31	Request Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
REQFLAG	Indicates the divider status. A clock being divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter for SDADC functional clock to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

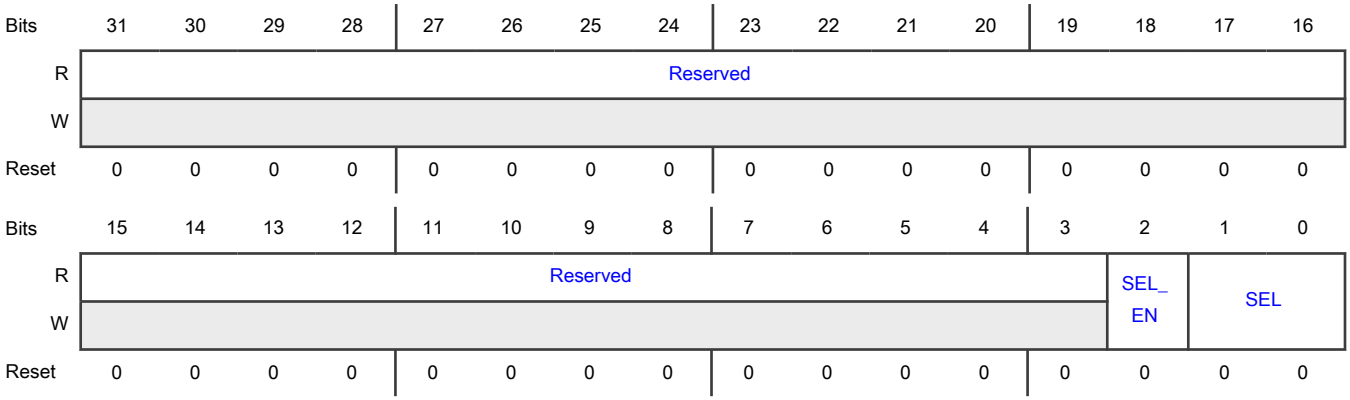
28.7.4.18 ADC0 (SARADC) Functional Clock Source Select (SARADCFCLKSEL)

Offset

Register	Offset
SARADCFCLKSEL	620h

Function
Selects the clock source for ADC0 functional clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables the mux output for ADC0 clock to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects a clock source for ADC0 (SARADC) clock. 00b - baseclk_sense 01b - main_pll_pfd1 10b - fro2_max 11b - osc_clk

28.7.4.19 ADC0 (SARADC) Functional Clock Divider (SARADCFCLKDIV)

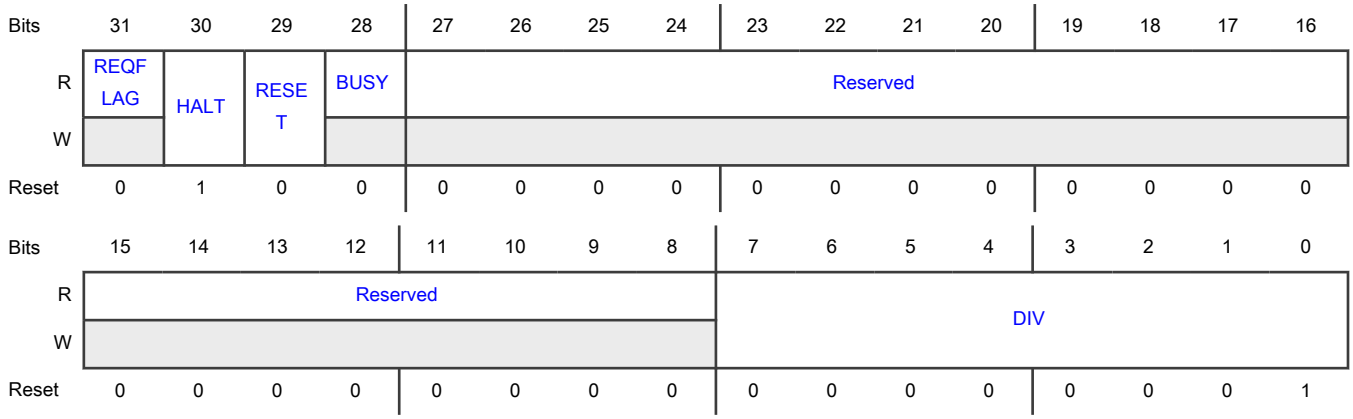
Offset

Register	Offset
SARADCFCLKDIV	624h

Function

Controls the divider for the ADC0 (SARADC) function clock saradc_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter for ADC0 (SARADC) clock to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down)

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selection Selects a divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.4.20 Wake 32 kHZ Clock Source Select (WAKE32KCLKSEL)

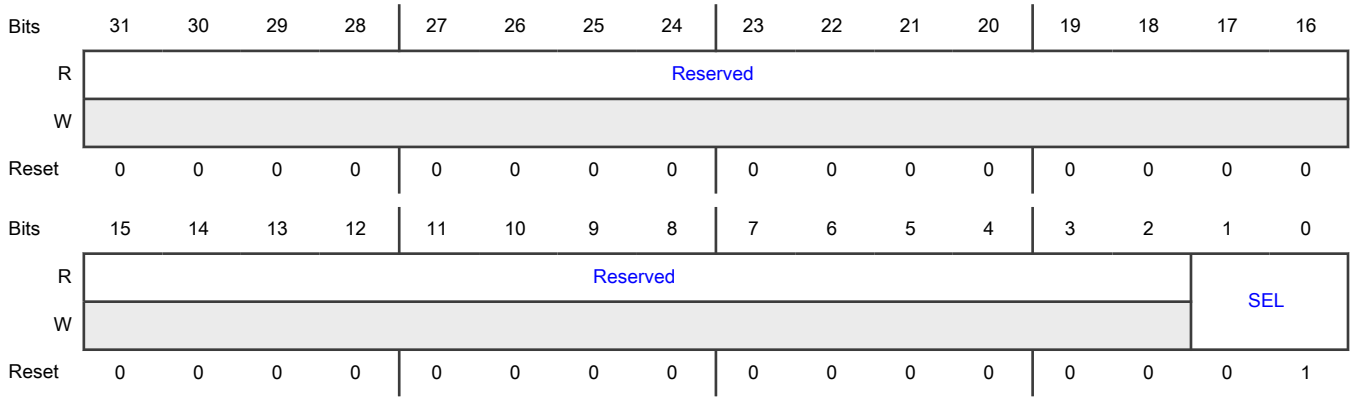
Offset

Register	Offset
WAKE32KCLKSEL	750h

Function

Selects clock source for wake 32 kHz clock.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Clock Source Select Selects clock source for wake 32 kHz clock. 00b - osc_32k 01b - lp_32k (1m_lposc clock divided by 32) 10b - Reserved 11b - Tied to logic 0. This may be selected to reduce power when no output is needed.

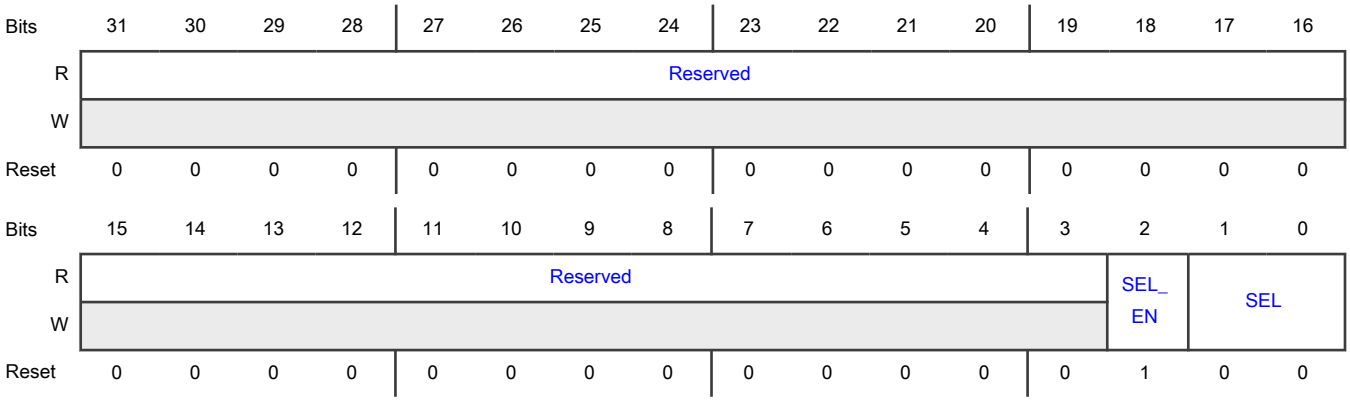
28.7.4.21 MICFIL Functional Clock Source Select (MICFILFCLKSEL)

Offset

Register	Offset
MICFILFCLKSEL	780h

Function
Selects the source for MICFIL clock.

Diagram



Fields

Field	Function
31-3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 SEL_EN	Clock MUX Output Enable Enables the clock mux output to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select Selects a source for MICFIL clock. 00b - baseclk_sense 01b - audio_pll_pfd3 10b - fro2_max 11b - audio_clk

28.7.4.22 MICFIL Functional Clock Divider (MICFILFCLKDIV)

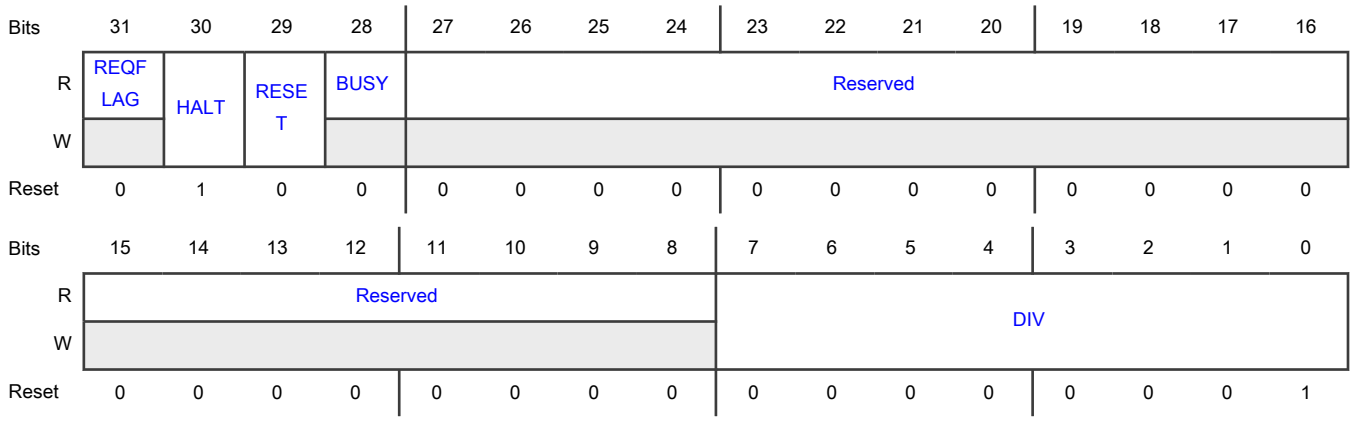
Offset

Register	Offset
MICFILFCLKDIV	784h

Function

Controls the divider for the MICFIL function clock micfil_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change. 0b - The divider change has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. The intent is to allow the divider clock source to be changed without the risk of a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter for MICFIL clock to use a new divider value immediately, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Selects a divider: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

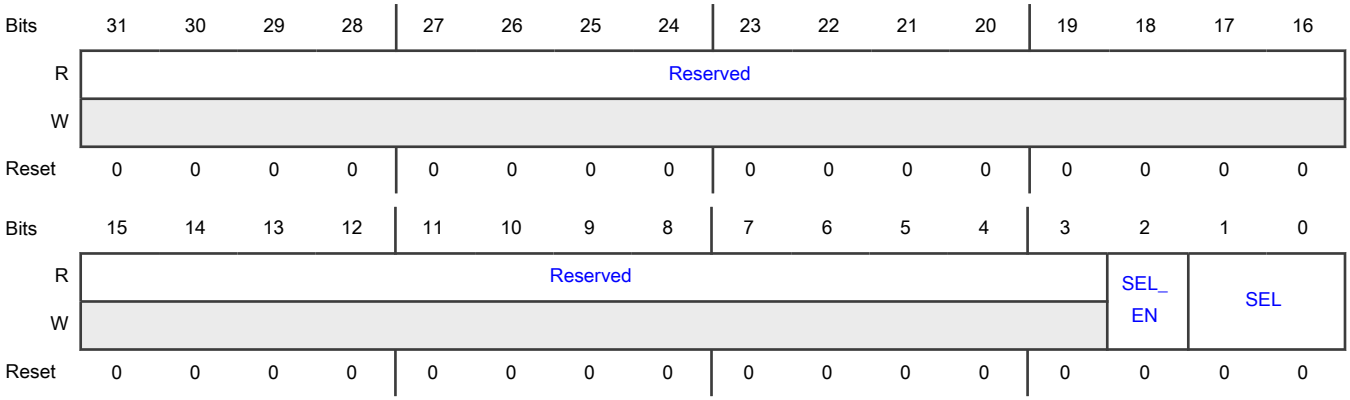
28.7.4.23 LPI2C15 Functional Clock Source Select (LPI2C15FCLKSEL)

Offset

Register	Offset
LPI2C15FCLKSEL	788h

Function
Selects the clock source for the LPI2C15 clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock MUX Output Enable Enables output of clock mux for LPI2C15 to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	LPI2C15 Clock Source Select Selects the clock source for LPI2C15 functional clock. 00b - baseclk_sense 01b - fro1_max 10b - fro1_div2 11b - fro2_max

28.7.4.24 LPI2C15 Functional Clock Divider (LPI2C15FCLKDIV)

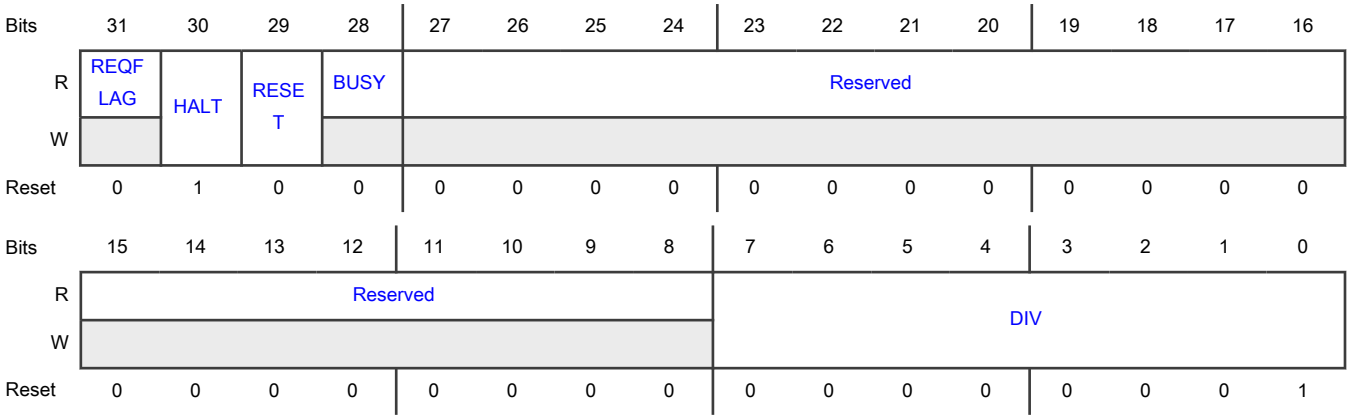
Offset

Register	Offset
LPI2C15FCLKDIV	78Ch

Function

Controls the divider for the LPI2C15 functional clock lpi2c_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. A clock being divided must be running for this status to change. 0b - The Divider change has been finished (clock being divided must be running for this status to change). 1b - The Divider value is being changed.
30 HALT	Divider Counter Halt Halts the divider counter. The intent is to allow the divider's clock source to be changed, without causing a glitch at the output. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	LPI2C15 Clock Divider Value Selects the divider value: • 0: Divide by 1 • 1: Divide by 2 • . . . • 256: Divide by 256

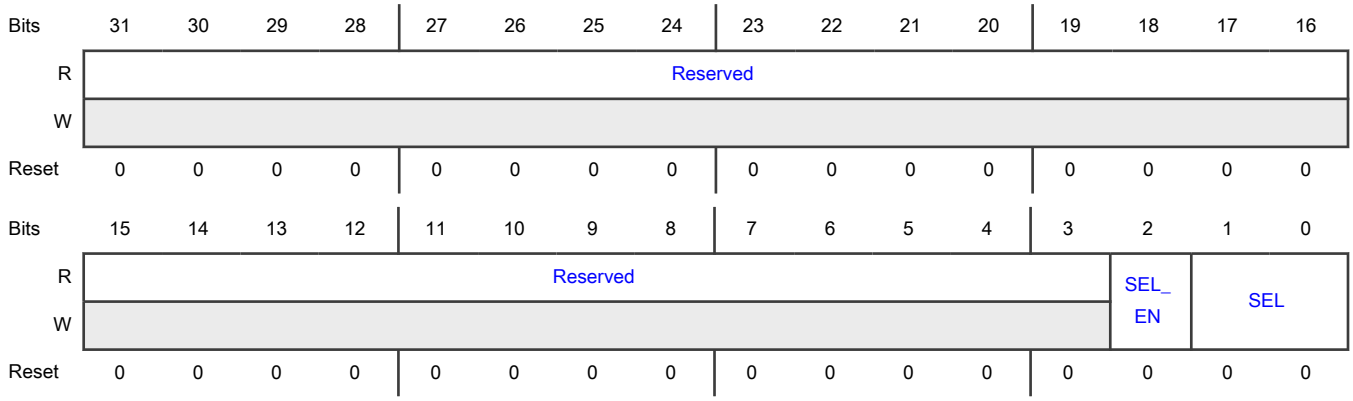
28.7.4.25 CLKOUT_VDD1 Clock Source Select (CLKOUTCLKSEL)

Offset

Register	Offset
CLKOUTCLKSEL	800h

Function
Selects the CLKOUT_VDD1 clock source of VDD1_SENSE domain.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables output of clock mux to reduce power. 0b - Gates output of clock mux. 1b - Enables output of clock mux.
1-0 SEL	Clock Source Select 00b - baseclk_sense 01b - audio_pll_pfd3 10b - fro2_max 11b - fro1_max

28.7.4.26 CLKOUT_VDD1 Clock Divider (CLKOUTCLKDIV)

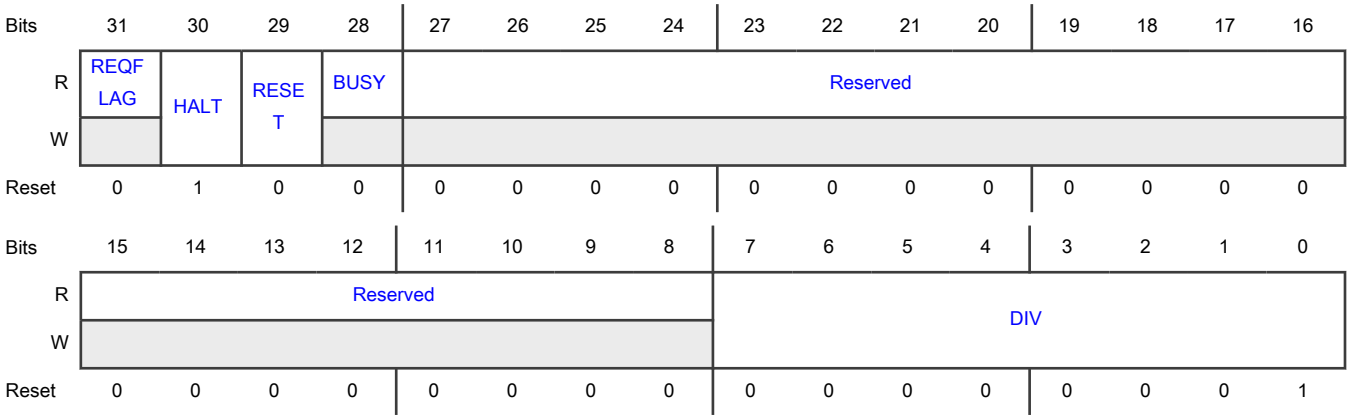
Offset

Register	Offset
CLKOUTCLKDIV	804h

Function

Controls the divider for the CLKOUT_VDD1 clock sense_clkout.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Shows the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value immediately, instead of completing the previous count. 0b - No effect 1b - Divider Counter Reset
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT_VDD1 is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	CLKOUT_VDD1 Clock Divider Selects a CLKOUT_VDD1 clock divider:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.4.27 VDD1_SENSE Peripheral Clock Control 0 (PSCCTL0_SENS)

Offset

Register	Offset
PSCCTL0_SENS	810h

Function

Allows software to enable specific peripherals. Writing a zero to any assigned bit in this register gates the clock. Writing an one enables the clock and allows the specified peripheral to operate. If VDD1_SENSE power domain is power OFF, reset after power ON.

NOTE

This register can only be written by CPU1. CPU0 can not write this register. However, CPU0 can read it to query the really status of device's clock. CPU1 can write zero to PSCCTL0_SENS to disable the clock. But whether this clock is really disabled, software need to read back the related bit from PSCCTL0_COMP, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						MEDIA_A...	LPI2C15	GLIKEY1	IOPCTL1	SYSCON3	Reserved	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Reserved	MICFIL	ACMP0	SARADC0	SDADC0	SEMA42_0	OSTIMER	MU2	MU1	MU0	Reserved			Reserved
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACCESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Enables media access RAM arbiter1 clock. 0b - Disable 1b - Enable
24 LPI2C15	LPI2C15 Clock Enables LPI2C15 clock. 0b - Disable 1b - Enable
23 GLIKEY1	GLIKEY1 Clock Enables GLIKEY1 clock. 0b - Disable 1b - Enable
22 IOPCTL1	IOPCTL1 Clock Enables IOPCTL1 clock. 0b - Disable 1b - Enable
21 SYSCON3	SYSCON3 Clock Enables SYSCON3 clock. 0b - Disable 1b - Enable
20 —	Reserved
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Enables MICFIL clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
11 ACMP0	ACMP0 Clock Enables ACMP0 clock. 0b - Disable 1b - Enable
10 SARADC0	ADC0 Clock Enables ADC0 clock. 0b - Disable 1b - Enable
9 SDADC0	SDADC0 Clock Enables SDADC0 clock. 0b - Disable 1b - Enable
8 SEMA42_0	SEMA42_0 Clock Enables SEMA42_0 clock. 0b - Disable 1b - Enable
7 OSTIMER	OSTIMER Clock Enables OSTIMER clock. 0b - Disable 1b - Enable
6 MU2	MU2 Clock Enables MU2 clock. 0b - Disable 1b - Enable
5 MU1	MU1 Clock Enables MU1 clock. 0b - Disable 1b - Enable
4	MU0 Clock

Table continues on the next page...

Table continued from the previous page...

Field	Function
MU0	Enables MU0 clock. 0b - Disable 1b - Enable
3-1 —	Reserved
0 —	Reserved

28.7.4.28 VDD1_SENSE Peripheral Clock Control 0 Set (PSCCTL0_SENS_SET)

Offset

Register	Offset
PSCCTL0_SENS_SET	840h

Function

Writing an one to a bit position in PSCCTL0_SET sets the corresponding position in PSCCTL0.

NOTE

This register can only be written by CPU1. CPU0 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						MEDIA_A...	LPI2C15	Reserved	IOPCTL1	SYSCON3	Reserved	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Reserved	MICFIL	ACMP0	SARADC0	SDADC0	SEMA42_0	OSTIMER	MU2	MU1	MU0	Reserved			CPU1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Enables Media Access_RAM arbiter1 clock. 0b - No effect 1b - Enable
24 LPI2C15	LPI2C15 Clock Enables LPI2C15 clock. 0b - No effect 1b - Enable
23 —	Reserved
22 IOPCTL1	IOPCTL1 Clock Enables IOPCTL1 clock. 0b - No effect 1b - Enable
21 SYSCON3	SYSCON3 Clock Enables SYSCON3 clock. 0b - No effect 1b - Enable
20 —	Reserved
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Enables MICFIL clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 ACMP0	ACMP0 Clock Enables ACMP0 clock. 0b - No effect 1b - Enable
10 SARADC0	ADC0 Clock Enables ADC0 clock. 0b - No effect 1b - Enable
9 SDADC0	SDADC0 Clock Enables SDADC0 clock. 0b - No effect 1b - Enable
8 SEMA42_0	SEMA42_0 Clock Enables SEMA42_0 clock. 0b - No effect 1b - Enable
7 OSTIMER	OSTIMER Clock Enables OSTIMER clock. 0b - No effect 1b - Enable
6 MU2	MU2 Clock Enables MU2 clock. 0b - No effect 1b - Enable
5 MU1	MU1 Clock Enables MU1 clock. 0b - No effect 1b - Enable
4 MU0	MU0 Clock Enables MU0 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
3-1 —	Reserved
0 CPU1	CPU1 Clock Enables CPU1 clock. 0b - No effect 1b - Enable

28.7.4.29 VDD1_SENSE Peripheral Clock Control 0 Clear (PSCCTL0_SENS_CLR)

Offset

Register	Offset
PSCCTL0_SENS_CLR	870h

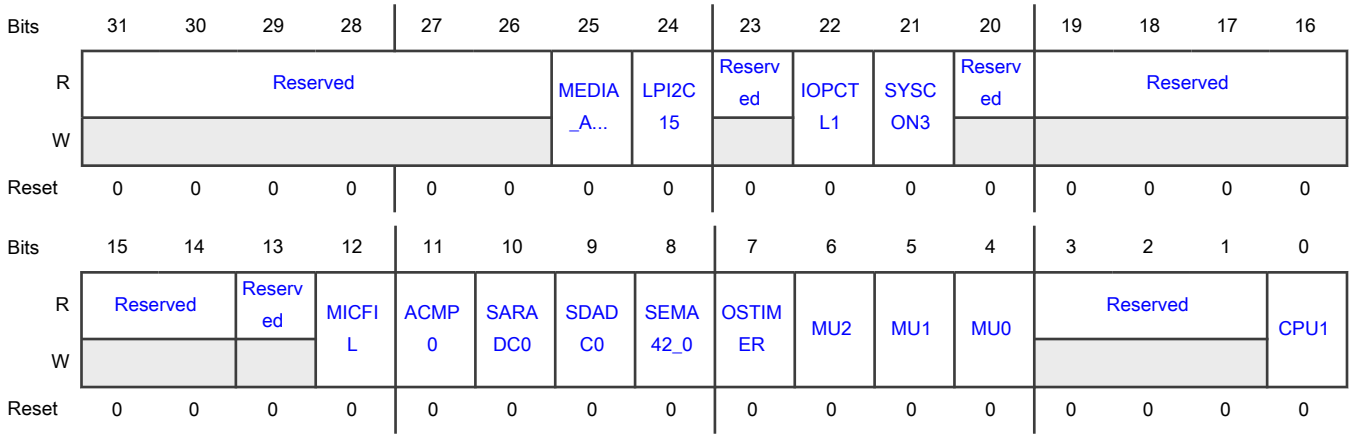
Function

Writing an one to a bit position in PSCCTL0_CLR clears the corresponding position in PSCCTL0.

NOTE

This register can only be written by CPU1. CPU0 can not write this register. CPU1 can write 1 to PSCCTL0_SENS_CLR to disable the clock. But whether this clock is really disabled, software need to read back the related bit from PSCCTL0_COMP, since if one core enables the clock, it cannot be disabled by another core.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 MEDIA_ACCESS_RAM_ARBITER1	Media Access RAM Arbiter1 Clock Disables media access RAM arbiter1 clock. 0b - No effect 1b - Disable
24 LPI2C15	LPI2C15 Clock Disables LPI2C15 clock. 0b - No effect 1b - Disable
23 —	Reserved
22 IOPCTL1	IOPCTL1 Clock Disables IOPCTL1 clock. 0b - No effect 1b - Disable
21 SYSCON3	SYSCON3 Clock Disables SYSCON3 clock. 0b - No effect 1b - Disable
20 —	Reserved
19-14 —	Reserved
13 —	Reserved
12 MICFIL	MICFIL Clock Disables MICFIL clock. 0b - No effect 1b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 ACMP0	ACMP0 Clock Disables ACMP0 clock. 0b - No effect 1b - Disable
10 SARADC0	ADC0 Clock Disables ADC0 clock. 0b - No effect 1b - Disable
9 SDADC0	SDADC0 Clock Disables SDADC0 clock. 0b - No effect 1b - Disable
8 SEMA42_0	SEMA42_0 Clock Disables SEMA42_0 clock. 0b - No effect 1b - Disable
7 OSTIMER	OSTIMER Clock Disables OSTIMER clock 0b - No effect 1b - Disable
6 MU2	MU2 Clock Disables MU2 clock 0b - No effect 1b - Disable
5 MU1	MU1 Clock Disables MU1 clock 0b - No effect 1b - Disable
4 MU0	MU0 Clock Disables MU0 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable
3-1 —	Reserved
0 CPU1	CPU1 Clock Disables CPU1 clock. 0b - No effect 1b - Disable

28.7.4.30 One Source Clock Slice Enable for VDD1_SENSE CPU (ONE_SRC_CLKSLICE_ENABLE_SENSE)

Offset

Register	Offset
ONE_SRC_CLKSLICE_ENABLE_SENSE	890h

Function

Controls the gating of one source clock slice. Multi source clock slices gating is controlled by enabling the SEL_EN bit field of a series of clock select source registers, for example, XSPI*n*CLKSEL.

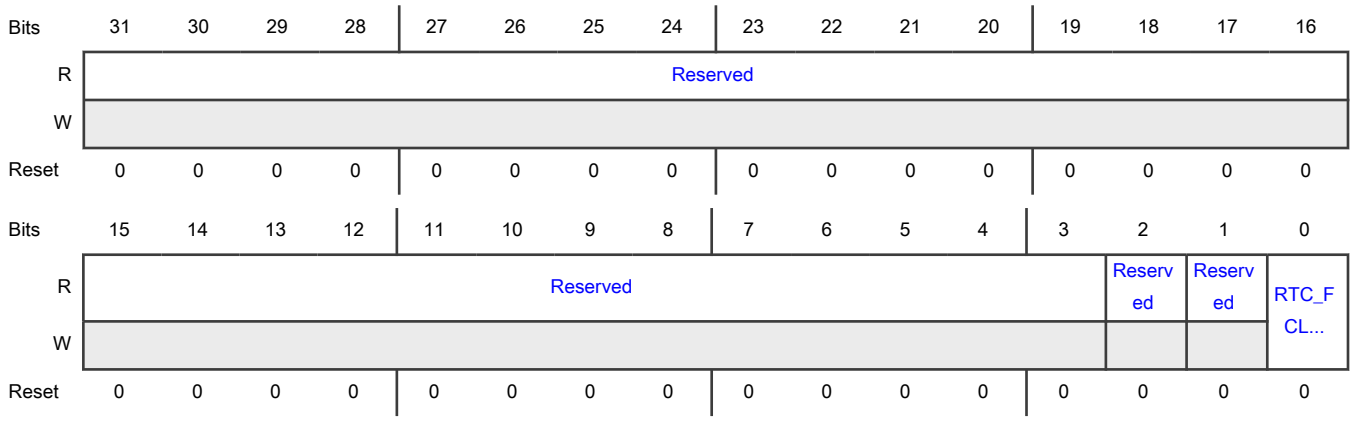
NOTE

Not all the clock select source registers have this SEL_EN bit for some clocks cannot be gated, such as base clock.

NOTE

This register is VDD1_SENSE core access only

Diagram



Fields

Field	Function
31-3 —	Reserved
2 —	Reserved
1 —	Reserved
0 RTC_FCLK_EN	RTC Functional Clock Gating Enable Enables RTC functional clock. This clock's source is lposc_1m. Make sure lposc_1m is active before enabling the RTC functional clock. 0b - Gates RTC functional clock. 1b - Enables RTC functional clock.

28.7.5 CLKCTL4 register descriptions

This section includes the memory map and register descriptions for Clock Controller 4.

The peripherals in CLKCTL4 are shared for both CPU0 and CPU1. There is a common scenario where CPU0 wants to turn off one clock, but CPU1 is still using this clock. To handle such kind of scenario, two sets of registers are configured in CLKCTL4: PSCCTLn_COMP (these registers can only be written by CPU0, but they can be read by both CPU0 and CPU1.) and PSCCTLn_SENS (these registers can only be written by CPU1, but they can be read by both CPU1 and CPU0.). In this way, if CPU0 does not need such peripheral work, the peripheral can be disabled by writing PSCCTLn_COMP. CPU0 can read PSCCTLn_SENS back to check whether this clock is really disabled.

28.7.5.1 CLKCTL4 memory map

CLKCTL4 base address: 400A_1000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDD2_COMP Media Peripheral Clock Control 0 (PSCCTL0_MEDIA)	32	RW	2080_2000h
14h	VDD2_COMP Media Peripheral Clock Control 1 (PSCCTL1_MEDIA)	32	RW	0000_0000h
40h	VDD2_COMP Media Peripheral Clock Control 0 Set (PSCCTL0_MEDIA_SET)	32	RW	0000_0000h
44h	VDD2_COMP Media Peripheral Clock Control 1 Set (PSCCTL1_MEDIA_SET)	32	RW	0000_0000h
70h	VDD2_COMP Media Peripheral Clock Control 0 Clear (PSCCTL0_MEDIA_CLR)	32	RW	0000_0000h
74h	VDD2_COMP Media Peripheral Clock Control 1 Clear (PSCCTL1_MEDIA_CLR)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
90h	One Source Clock Slice Enable for VDD2_COMP Core (ONE_SRC_CLKSLICE_ENABLE)	32	RW	0000_0000h
A4h	VDDN_MEDIA Clock Source Select (MEDIAVDDNCLKSEL)	32	RW	0000_0000h
ACh	VDDN_MEDIA Clock Divider (MEDIAVDDNCLKDIV)	32	RW	4000_0001h
104h	Media Main Clock Source Select (MEDIAMAINCLKSEL)	32	RW	0000_0000h
10Ch	Media Main Clock Divider (MEDIAMAINCLKDIV)	32	RW	4000_0001h
110h	VDDN_MEDIA Base Clock Source Select (MDNBASECLKSEL)	32	RW	0000_0000h
114h	VDD2_MEDIA Base Clock Source Select (MD2BASECLKSEL)	32	RW	0000_0000h
200h	XSPI2 Functional Clock Source Select (XSPI2FCLKSEL)	32	RW	0000_0000h
204h	XSPI2 Functional Clock Divider (XSPI2FCLKDIV)	32	RW	4000_0001h
220h	USB0 Functional Clock Source Select (USBFCLKSEL)	32	RW	0000_0000h
240h	USB1 (eUSB) Functional Clock Source Select (eUSBFCLKSEL)	32	RW	0000_0000h
260h	SDIO0 Functional Clock Source Select (SDIO0FCLKSEL)	32	RW	0000_0000h
264h	SDIO0 Functional Clock Divider (SDIO0FCLKDIV)	32	RW	4000_0001h
280h	SDIO1 Functional Clock Source Select (SDIO1FCLKSEL)	32	RW	0000_0000h
284h	SDIO1 Functional Clock Divider (SDIO1FCLKDIV)	32	RW	4000_0001h
300h	MIPI_DSI_Host PHY Clock Source Select (DPHYCLKSEL)	32	RW	0000_0000h
304h	MIPI_DSI_Host PHY Clock Divider (DPHYCLKDIV)	32	RW	4000_0001h
308h	MIPI_DSI_Host DPHY Escape Mode Clock Source Select (DPHYESCCLKSEL)	32	RW	0000_0000h
30Ch	MIPI_DSI_Host DPHY Escape Mode Receive Clock Divider (DPHYESCRXCLKDIV)	32	RW	4000_0010h
310h	MIPI_DSI_Host DPHY Escape Mode Transmit Clock Divider (DPHYESCTXCLKDIV)	32	RW	4000_0011h
320h	VGPU Clock Source Select (VGPUCLKSEL)	32	RW	0000_0000h
324h	VGPU Clock Divider (VGPUCLKDIV)	32	RW	4000_0001h
328h	LPSPI14 Clock Source Select (LPSPI14CLKSEL)	32	RW	0000_0000h
32Ch	LPSPI14 Clock Divider (LPSPI14CLKDIV)	32	RW	4000_0001h
330h	LPSPI16 Clock Source Select (LPSPI16CLKSEL)	32	RW	0000_0000h
334h	LPSPI16 Clock Divider (LPSPI16CLKDIV)	32	RW	4000_0001h
338h	FLEXIO Clock Source Select (FLEXIOCLKSEL)	32	RW	0000_0000h
33Ch	FLEXIO Clock Divider (FLEXIOCLKDIV)	32	RW	4000_0001h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
340h	LCDIF Pixel Clock Source Select (LCDIFPIXELCLKSEL)	32	RW	0000_0000h
344h	LCDIF Pixel Clock Divider (LCDIFPIXELCLKDIV)	32	RW	4000_0001h
700h	Low frequency Clock Divider (LOWFREQCLKDIV)	32	RW	4000_0001h
810h	VDD1_SENSE Media Peripheral Clock Control 0 (PSCCTL0_SENS)	32	RW	2080_2000h
814h	VDD1_SENSE Media Peripheral Clock Control 1 (PSCCTL1_SENS)	32	RW	0000_0000h
840h	VDD1_SENSE Media Peripheral Clock Control 0 Set (PSCCTL0_SENS_SET)	32	RW	0000_0000h
844h	VDD1_SENSE Media Peripheral Clock Control 1 Set (PSCCTL1_SENS_SET)	32	RW	0000_0000h
870h	VDD1_SENSE Media Peripheral Clock Control 0 Clear (PSCCTL0_SENS_CLR)	32	RW	0000_0000h
874h	VDD1_SENSE Media Peripheral Clock Control 1 Clear (PSCCTL1_SENS_CLR)	32	RW	0000_0000h
890h	One Source Clock Slice Enable for VDD1_SENSE Core (ONE_SRC_CLKSLICE_ENABLE_SENSE)	32	RW	0000_0000h

28.7.5.2 VDD2_COMP Media Peripheral Clock Control 0 (PSCCTL0_MEDIA)

Offset

Register	Offset
PSCCTL0_MEDIA	10h

Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing zero disables the clock. If VDD2_MEDIA power rail is OFF, reset after power ON.

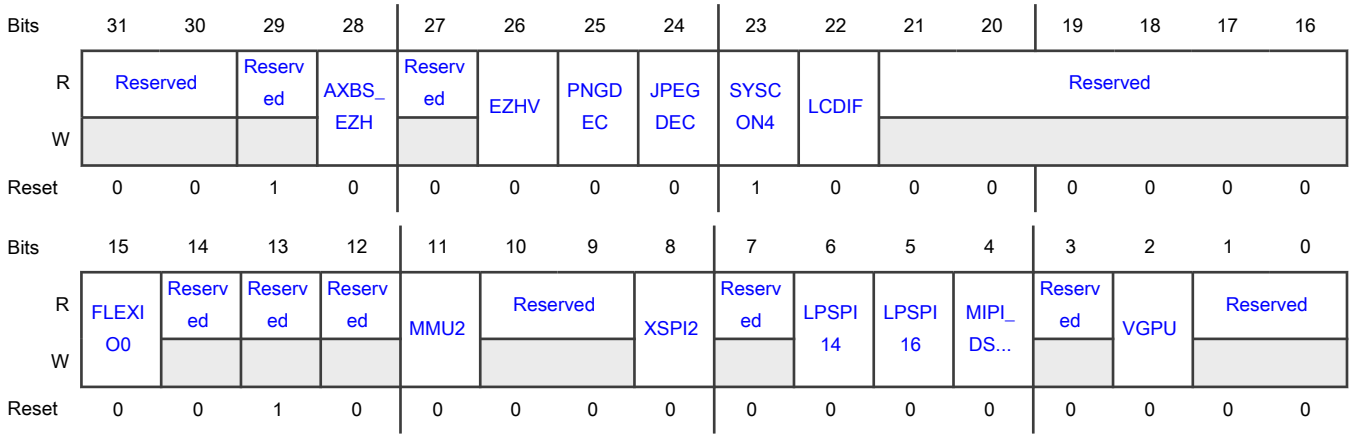
NOTE

This register can only be written by CPU0. CPU1 can not write this register but CPU1 can read it to query the real status of the device's clock. CPU0 can write zero to PSCCTL0_MEDIA to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL0_SENS, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS for EZHV Clock Enables AXBS for EZHV clock. 0b - Disable 1b - Enable
27 —	Reserved
26 EZHV	EZHV Clock Enables EZHV clock. 0b - Disable 1b - Enable
25 PNGDEC	PNGDEC Clock Enables PNGDEC clock. 0b - Disable 1b - Enable
24 JPEGDEC	JPEGDEC Clock Enables JPEGDEC clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
23 SYSCON4	SYSCON4 Clock Enables SYSCON4 clock. 0b - Disable 1b - Enable
22 LCDIF	LCDIF Clock Enables LCDIF clock. 0b - Disable 1b - Enable
21-16 —	Reserved
15 FLEXIO0	FLEXIO0 Clock Enables FLEXIO0 clock. 0b - Disable 1b - Enable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Enables MMU2 clock. 0b - Disable 1b - Enable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Enables XSPI2 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Enables LPSPI14 clock. 0b - Disable 1b - Enable
5 LPSPI16	LPSPT16 Clock Enables LPSPI16 clock. 0b - Disable 1b - Enable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Enables MIPI_DSI_HOST clock. 0b - Disable 1b - Enable
3 —	Reserved
2 VGPU	VGPU Clock Enables VGPU clock. 0b - Disable 1b - Enable
1-0 —	Reserved

28.7.5.3 VDD2_COMP Media Peripheral Clock Control 1 (PSCCTL1_MEDIA)

Offset

Register	Offset
PSCCTL1_MEDIA	14h

Function

Allows software to control specific peripheral's clock. Writing 1 to any assigned bit in this register enables the clock and allows the specified peripheral to operate. Writing 0 disables the clock. If VDD2_MEDIA power rail is OFF, reset after power ON.

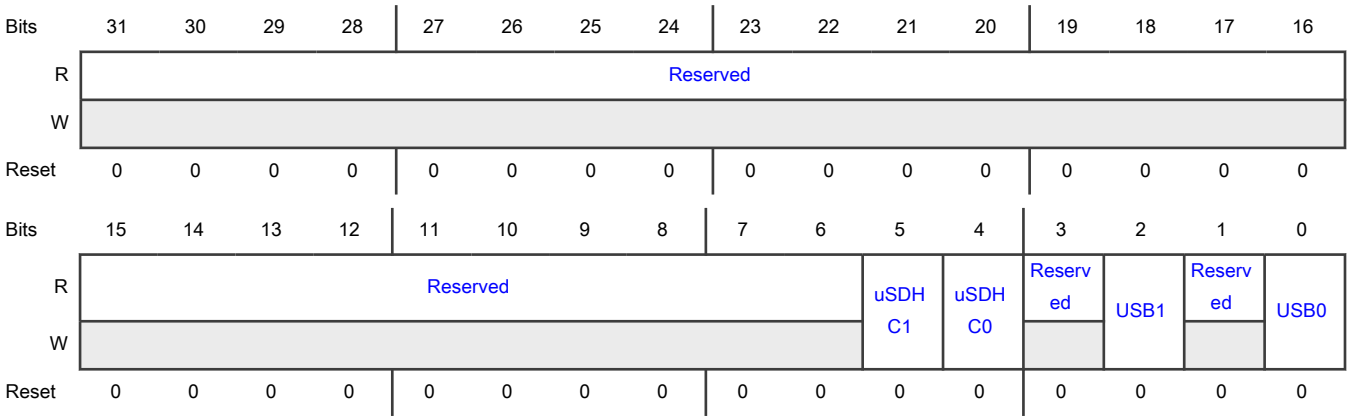
NOTE

This register can only be written by CPU0. CPU1 can not write this register but CPU1 can read it to query the really status of device's clock. CPU0 can write zero to PSCCTL1_MEDIA to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL1_SENS, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Clock Enables uSDHC1 clock. 0b - Disable 1b - Enable
4 uSDHC0	uSDHC0 Clock Enables uSDHC0 clock. 0b - Disable 1b - Enable
3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 USB1	USB1 and eUSBPHY Clock Enables USB1 (eUSB) and eUSBPHY clock. 0b - Disable 1b - Enable
1 —	Reserved
0 USB0	USB0 and USBPHY0 Clock Enables USB0 and USBPHY0 clock. 0b - Disable 1b - Enable

28.7.5.4 VDD2_COMP Media Peripheral Clock Control 0 Set (PSCCTL0_MEDIA_SET)

Offset

Register	Offset
PSCCTL0_MEDIA_SET	40h

Function

Writing an one to a bit position in PSCCTL0_MEDIA_SET sets the corresponding position in PSCCTL0_MEDIA.

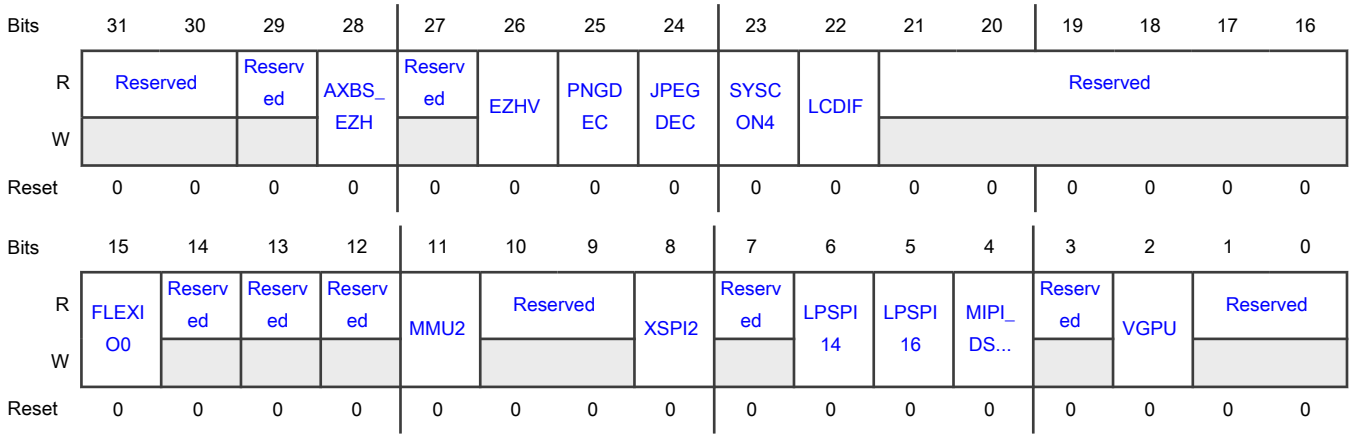
NOTE

This register can only be written by CPU0. CPU1 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS for EZHV Clock Enables AXBS for EZHV clock. 0b - No effect 1b - Enable
27 —	Reserved
26 EZHV	EZHV Clock Enables EZHV clock. 0b - No effect 1b - Enable
25 PNGDEC	PNGDEC Clock Enables PNGDEC clock. 0b - No effect 1b - Enable
24 JPEGDEC	JPEGDEC Clock Enables JPEGDEC clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Enable
23 SYSCON4	SYSCON4 Clock Enables SYSCON4 clock. 0b - No effect 1b - Enable
22 LCDIF	LCDIF Clock Enables LCDIF clock. 0b - No effect 1b - Enable
21-16 —	Reserved
15 FLEXIO0	FLEXIO0 Clock Enables FLEXIO0 clock. 0b - No effect 1b - Enable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Enables MMU2 clock. 0b - No effect 1b - Enable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Enables XSPI2 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Enable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Enables LPSPI14 clock. 0b - No effect 1b - Enable
5 LPSPI16	LPSPI16 Clock Enables LPSPI16 clock. 0b - No effect 1b - Enable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Enables MIPI_DSI_HOST clock. 0b - No effect 1b - Enable
3 —	Reserved
2 VGPU	VGPU Clock Enables VGPU clock. 0b - No effect 1b - Enable
1-0 —	Reserved

28.7.5.5 VDD2_COMP Media Peripheral Clock Control 1 Set (PSCCTL1_MEDIA_SET)

Offset

Register	Offset
PSCCTL1_MEDIA_SET	44h

Function

Writing an one to a bit position in PSCCTL1_MEDIA_SET sets the corresponding position in PSCCTL1_MEDIA.

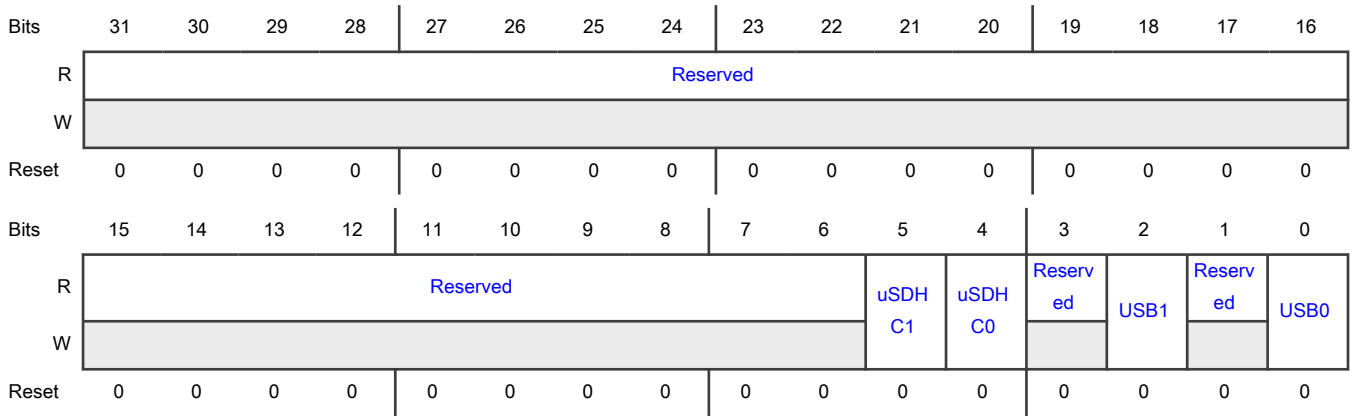
NOTE

This register can only be written by CPU0. CPU1 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved Slot reserved
5 uSDHC1	uSDHC1 Clock Enables uSDHC1 clock. 0b - No effect 1b - Enable
4 uSDHC0	uSDHC0 Clock Enables uSDHC0 clock. 0b - No effect 1b - Enable
3 —	Reserved
2 USB1	USB1 and eUSBPHY Clock Enables USB1 (eUSB) and eUSBPHY clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 —	Reserved
0 USB0	USB0 and USBPHY0 Clock Enables USB0 and USBPHY0 clock. 0b - No effect 1b - Enable

28.7.5.6 VDD2_COMP Media Peripheral Clock Control 0 Clear (PSCCTL0_MEDIA_CLR)

Offset

Register	Offset
PSCCTL0_MEDIA_CLR	70h

Function

Writing an one to a bit position in PSCCTL0_MEDIA_CLR clears the corresponding position in PSCCTL0_MEDIA.

NOTE

This register can only be written by CPU0, CPU1 can not write this register. CPU0 can write 1 to PSCCTL0_MEDIA_CLR to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL0_SENS, since if one core enables the clock, it cannot be disabled by another core.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		Reserved	AXBS_EZH	Reserved	EZHV	PNGDEC	JPEGDEC	SYSCON4	LCDIF	Reserved					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FLEXIO0	Reserved	Reserved	Reserved	MMU2	Reserved		XSPI2	Reserved	LPSPI14	LPSPI16	MIPI_DS...	Reserved	VGPU	Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS_EZHV Clock Disables EZH clock. 0b - No effect 1b - Disable
27 —	Reserved
26 EZHV	EZHV Clock Disables EZHV clock. 0b - No effect 1b - Disable
25 PNGDEC	PNG Decoder Clock Disables PNG decoder clock. 0b - No effect 1b - Disable
24 JPEGDEC	JPEGDEC Clock Disables JPEGDEC clock. 0b - No effect 1b - Disable
23 SYSCON4	SYSCON4 Clock Disables SYSCON4 clock. 0b - No effect 1b - Disable
22 LCDIF	LCDIF Clock Disables LCDIF clock. 0b - No effect 1b - Disable
21-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
15 FLEXIO0	FLEXIO0 Clock Disables FLEXIO0 clock control 0b - No effect 1b - Disable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Disables MMU2 clock. 0b - No effect 1b - Disable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Disables XSPI2 clock. 0b - No effect 1b - Disable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Disables LPSPI14 clock. 0b - No effect 1b - Disable
5 LPSPI16	LPSPI16 Clock Disables LPSPI16 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Disables MIPI_DSI_HOST clock. 0b - No effect 1b - Disable
3 —	Reserved
2 VGPU	VGPU Clock Disables VGPU clock. 0b - No effect 1b - Disable
1-0 —	Reserved

28.7.5.7 VDD2_COMP Media Peripheral Clock Control 1 Clear (PSCCTL1_MEDIA_CLR)

Offset

Register	Offset
PSCCTL1_MEDIA_CLR	74h

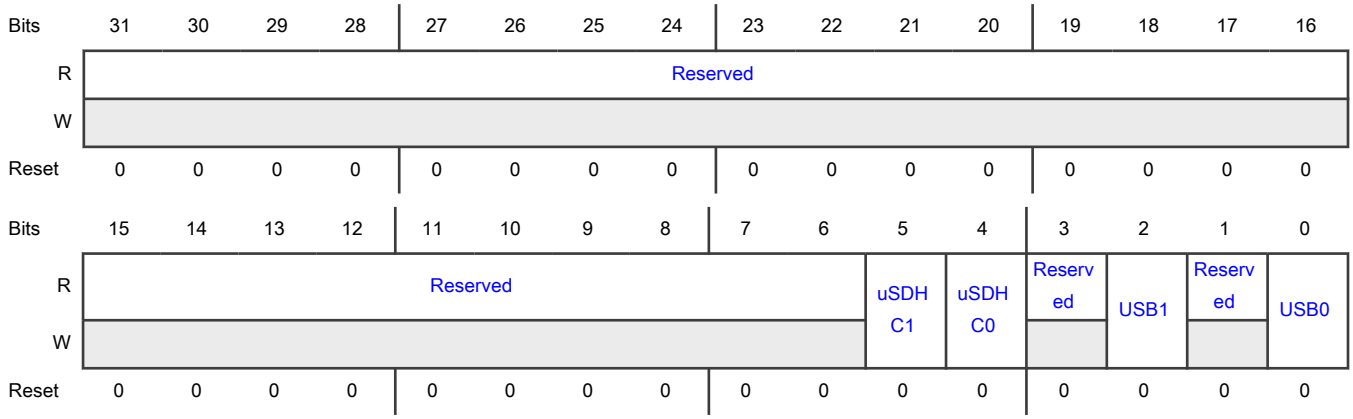
Function

Writing an one to a bit position in PSCCTL1_MEDIA_CLR clears the corresponding position in PSCCTL1_MEDIA.

NOTE

This register can only be written by CPU0, CPU1 can not write this register. CPU0 can write 1 to PSCCTL1_MEDIA_CLR to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL1_SENS, since if one core enables the clock, it cannot be disabled by another core.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Clock Disables uSDHC1 clock. 0b - No effect 1b - Disable
4 uSDHC0	uSDHC0 Clock Disables uSDHC0 clock. 0b - No effect 1b - Disable
3 —	Reserved
2 USB1	USB1 and eUSBPHY Clock Disables USB1 (eUSB) and eUSBPHY clock. 0b - No effect 1b - Disable
1 —	Reserved
0 USB0	USB0 and USBPHY0 Clock Disables USB0 and USBPHY0 clock. 0b - No effect 1b - Disable

28.7.5.8 One Source Clock Slice Enable for VDD2_COMP Core (ONE_SRC_CLKSLICE_ENABLE)

Offset

Register	Offset
ONE_SRC_CLKSLICE_ENABLE	90h

Function

Controls the gating of one source clock slice. Multi source clock slices gating is controlled by enabling the SEL_EN bit field of a series of clock select source registers, for example, XSPI*n*CLKSEL.

NOTE

Not all the clock select source registers have this SEL_EN bit for some clocks cannot be gated, such as base clock.

NOTE

This register is VDD2_COMP core access only.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															USBPHY_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 USBPHY_REFCLK_EN	USBPHY0 Reference Clock Enables USBPHY0 reference clock. This clock's source is osc_clk, please make sure the source clock is active before enabling the USBPHY0 reference clock. 0b - Gates USBPHY0 reference clock. 1b - Enables USBPHY0 reference clock.

28.7.5.9 VDDN_MEDIA Clock Source Select (MEDIAVDDNCLKSEL)

Offset

Register	Offset
MEDIAVDDNCLKSEL	A4h

Function

Selects a clock source for VDDN_MEDIA clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDDN Clock Source Select Selects clock source for VDDN_MEDIA clock. 00b - baseclk_mdn 01b - main_pll_pfd0 10b - fro0_max 11b - main_pll_pfd2

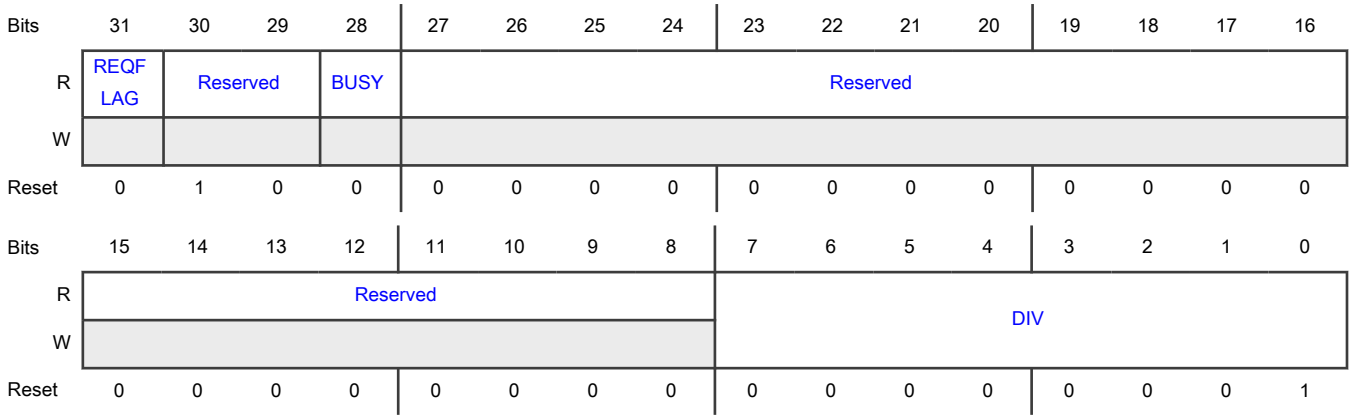
28.7.5.10 VDDN_MEDIA Clock Divider (MEDIAVDDNCLKDIV)

Offset

Register	Offset
MEDIAVDDNCLKDIV	ACh

Function
Controls the divider for the VDDN_MEDIA main clock media_vddn_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0	Clock Divider Value Select

Table continues on the next page...

Table continued from the previous page...

Field	Function
DIV	Selects a divider value for VDDN_MEDIA clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.11 Media Main Clock Source Select (MEDIAMAINCLKSEL)

Offset

Register	Offset
MEDIAMAINCLKSEL	104h

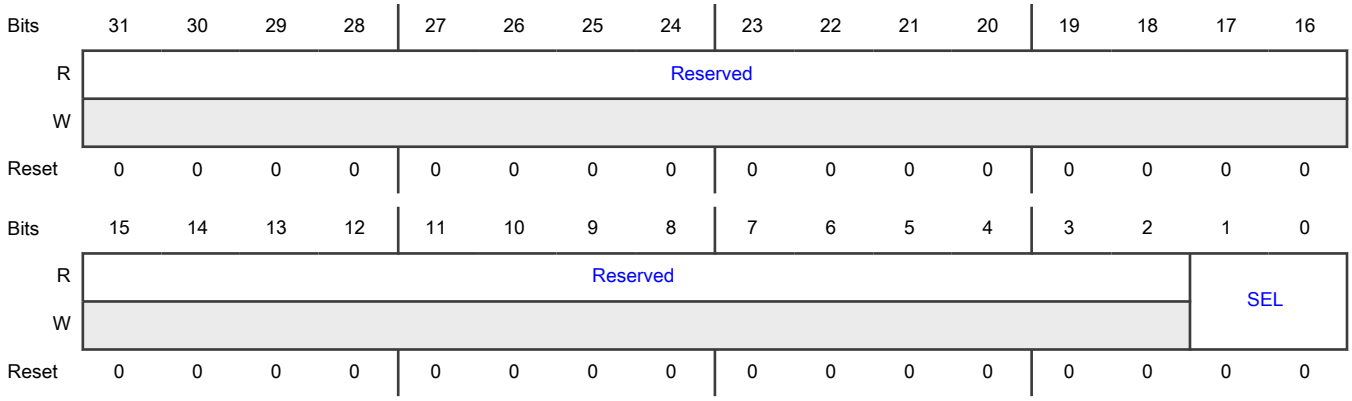
Function

Selects a clock source for VDD2_MEDIA main clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
1-0 SEL	Media Main Clock Source Select Selects the source for VDD2_MEDIA main clock. 00b - baseclk_md2 01b - main_pll_pfd0 10b - fro0_max 11b - main_pll_pfd2

28.7.5.12 Media Main Clock Divider (MEDIAMAINCLKDIV)

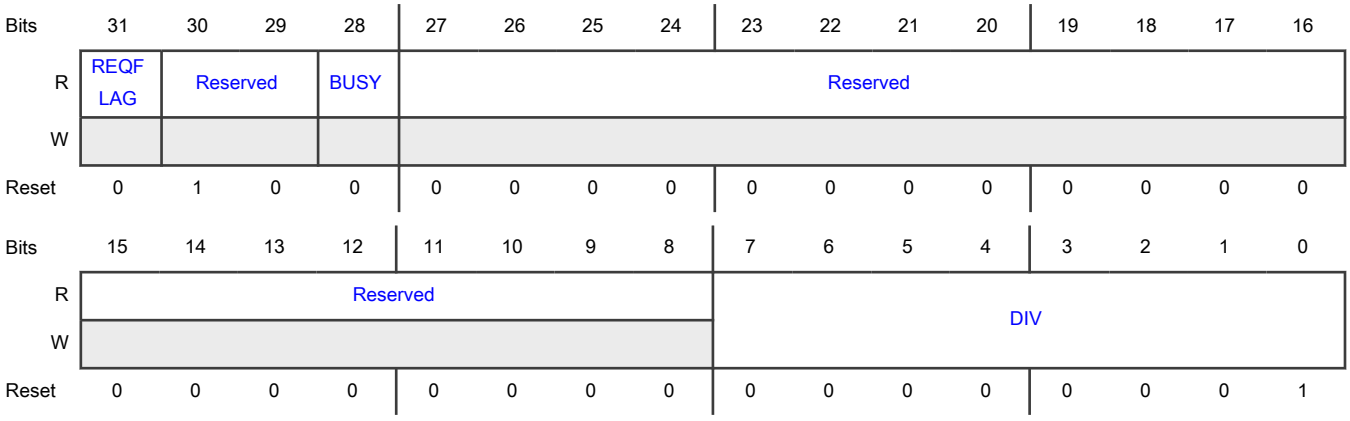
Offset

Register	Offset
MEDIAMAINCLKDIV	10Ch

Function

Controls the divider for the media function clock media_main_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for media main clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.13 VDDN_MEDIA Base Clock Source Select (MDNBASECLKSEL)

Offset

Register	Offset
MDNBASECLKSEL	110h

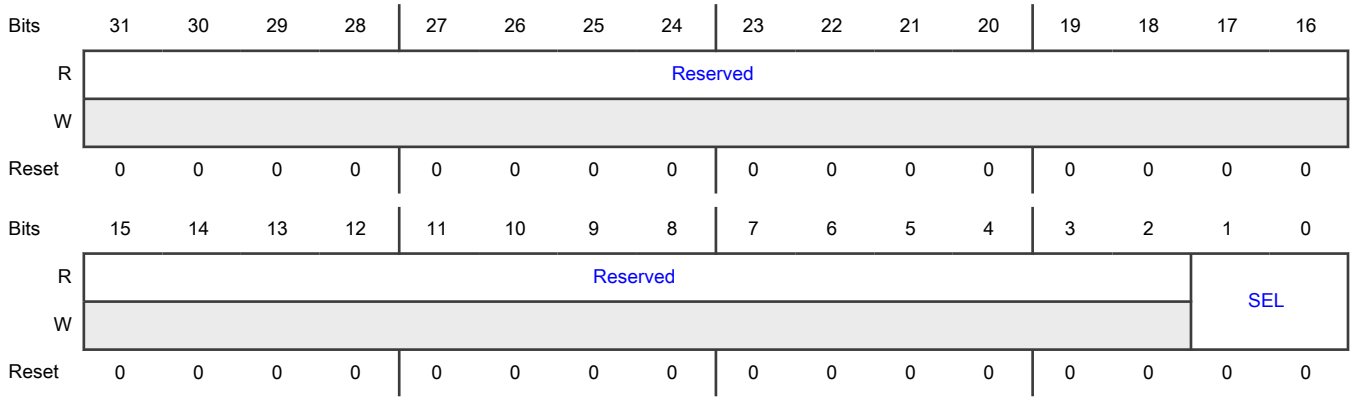
Function

Selects a clock source for VDDN_MEDIA base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	VDDN_MEDIA Base Clock Source Select Selects the source for the VDDN_MEDIA base clock. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.5.14 VDD2_MEDIA Base Clock Source Select (MD2BASECLKSEL)

Offset

Register	Offset
MD2BASECLKSEL	114h

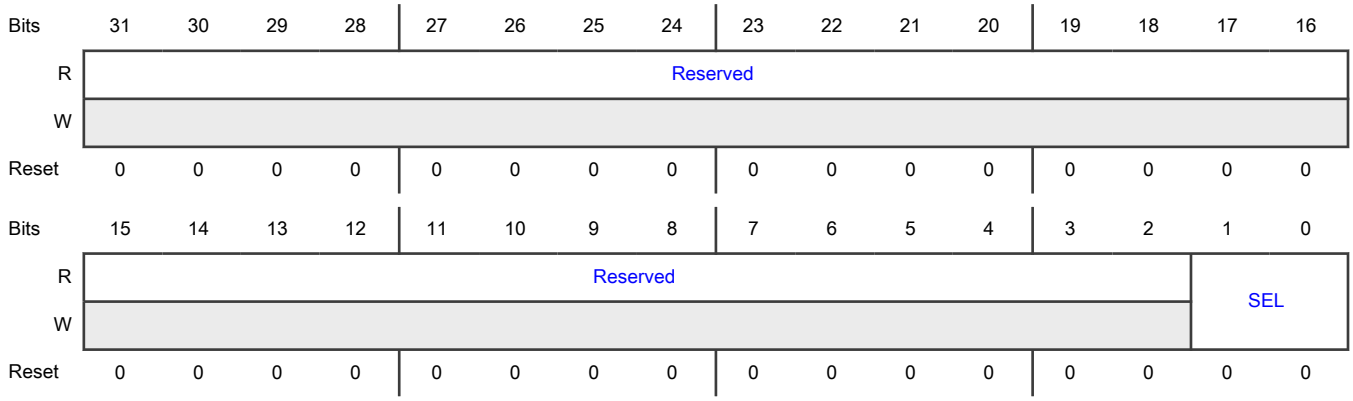
Function

Selects a clock source for VDD2_MEDIA base clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 SEL	Media VDD2 Base Clock Source Select Selects a clock source for the VDD2_MEDIA base clock. 00b - fro1_div3 01b - fro1_max 10b - fro0_div3 11b - 1m_lposc

28.7.5.15 XSPI2 Functional Clock Source Select (XSPI2FCLKSEL)

Offset

Register	Offset
XSPI2FCLKSEL	200h

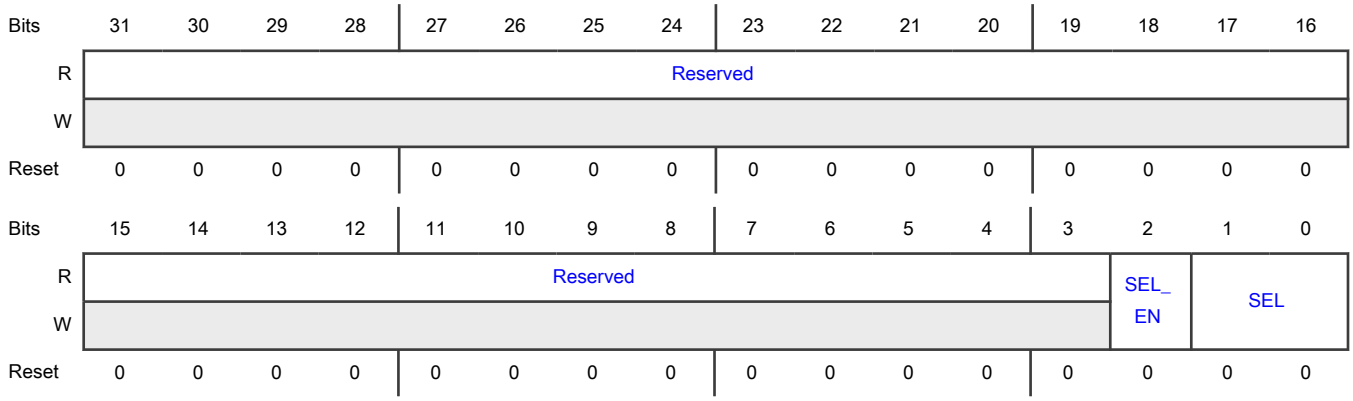
Function

Selects clock source for XSPI2 clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables the mux output for XSPI2 clock to reduce power. 0b - Gates mux output for XSPI2 clock. 1b - Enables mux output for XSPI2 clock.
1-0 SEL	XSPI2 Clock Source Select Selects source for XSPI2 clock. 00b - baseclk_comn 01b - audio_pll_pfd1 10b - fro0_max 11b - main_pll_pfd3

28.7.5.16 XSPI2 Functional Clock Divider (XSPI2FCLKDIV)

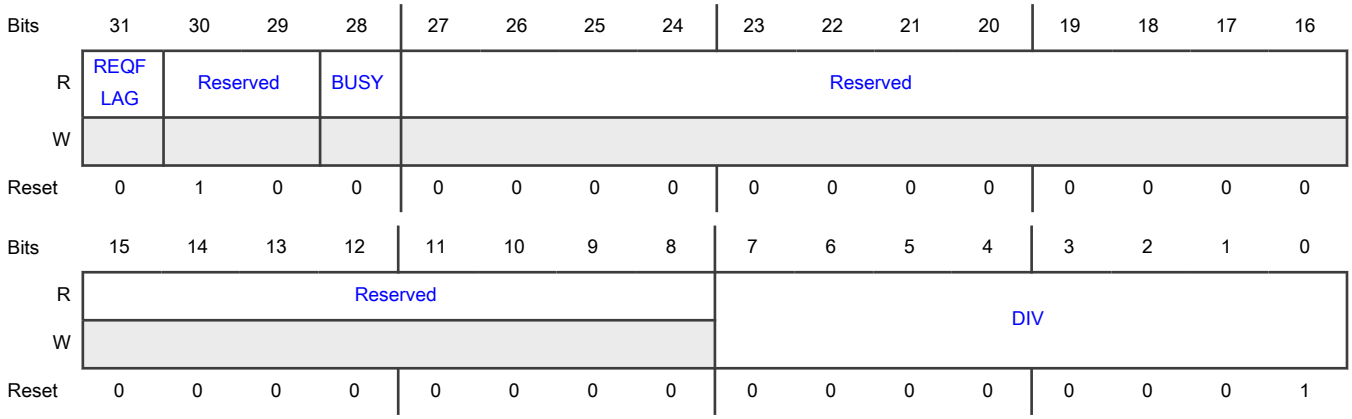
Offset

Register	Offset
XSPI2FCLKDIV	204h

Function

Controls the divider for XSPI2 clock xspi2_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30-29 —	Reserved
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for XSPI2 clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.17 USB0 Functional Clock Source Select (USBFCLKSEL)

Offset

Register	Offset
USBFCLKSEL	220h

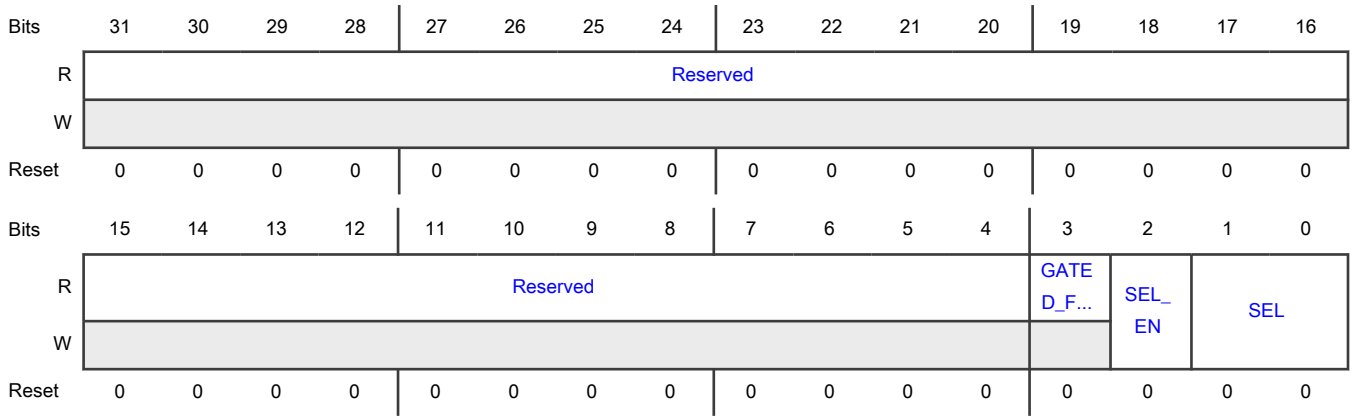
Function

Selects clock source for USB0 functional clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 GATED_FLAG	Gate Flag Indicates the gating status of USB0 clock. 0b - The USB0 fclk is not gated. 1b - The USB0 fclk is gated.
2 SEL_EN	Clock Mux Output Enable Enables mux output for USB0 clock to reduce power. 0b - Gates USB0 clock. It is necessary to read back bit3 with the value of 1 to confirm the clock is already gated then go ahead. 1b - Enables USB0 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
1-0 SEL	Clock Source Select Selects clock source for USB0 clock. 00b - wake32k_clk 01b - 1m_lposc 10b - osc_clk_usb 11b - Reserved

28.7.5.18 USB1 (eUSB) Functional Clock Source Select (eUSBFCLKSEL)

Offset

Register	Offset
eUSBFCLKSEL	240h

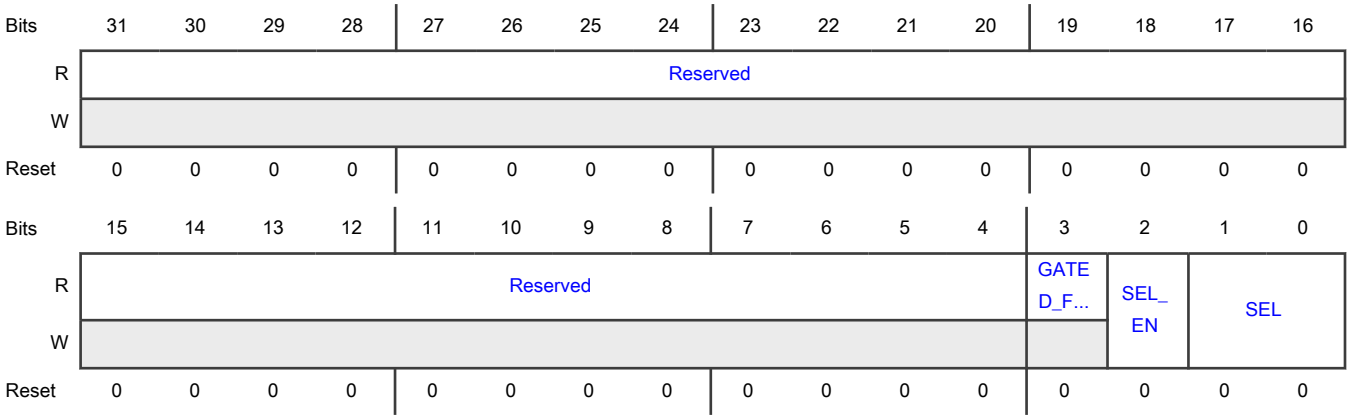
Function

Selects a clock source for USB1 (eUSB) functional clock.

NOTE

The clock selection is internally synchronized. The clocks that are switched from and switched to must run for changing selection.

Diagram



Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
3 GATED_FLAG	Gate Flag Indicates the gating status of USB1 (eUSB) clock. 0b - The USB1 functional clock is not gated. 1b - The USB1 functional clock is gated.
2 SEL_EN	Clock Mux Output Enable Enables the mux output for USB1 (eUSB) clock to reduce power. 0b - Gates mux output for USB1 functional clock. It is necessary to read back bit3 with the value of 1 to confirm the clock is already gated then go ahead. 1b - Enables mux output for USB1 functional clock.
1-0 SEL	USB1 Clock Source Select Selects a clock source for USB1 (eUSB) clock. 00b - wake32k_clk 01b - 1m_lposc 10b - osc_clk_eusb 11b - Reserved

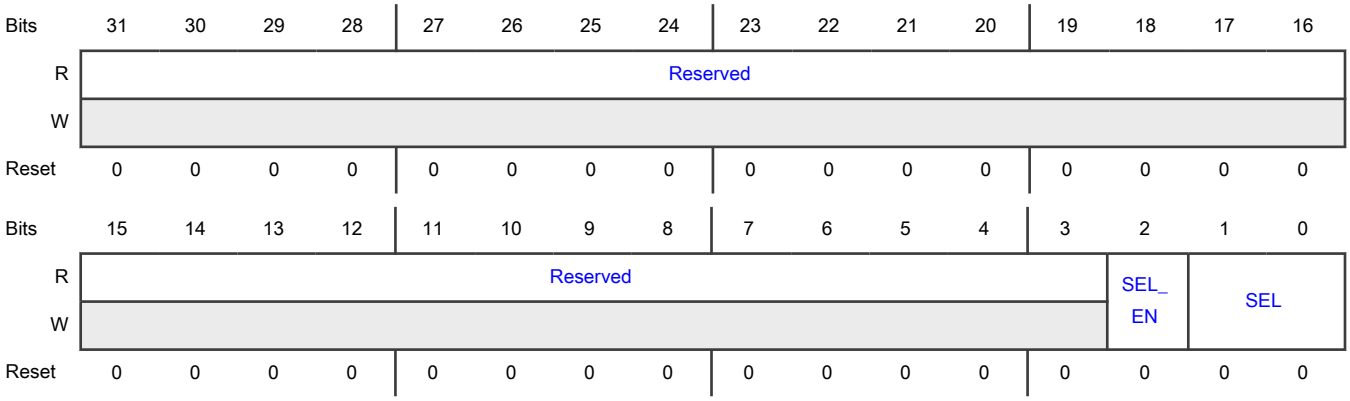
28.7.5.19 SDIO0 Functional Clock Source Select (SDIO0FCLKSEL)

Offset

Register	Offset
SDIO0FCLKSEL	260h

Function
Selects a clock source for the SD/MMC/SDIO interface 0 (uSDHC0).

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for SD/MMC/SDIO interface 0 to reduce power. 0b - Gates mux output for SD/MMC/SDIO interface 0 clock. 1b - Enables mux output for SD/MMC/SDIO interface 0 clock.
1-0 SEL	SDIO0 Functional Clock Source Select Selects a source for SD/MMC/SDIO interface 0. 00b - baseclk_mdn 01b - audio_pll_pfd0 10b - fro0_max 11b - main_pll_pfd2

28.7.5.20 SDIO0 Functional Clock Divider (SDIO0FCLKDIV)

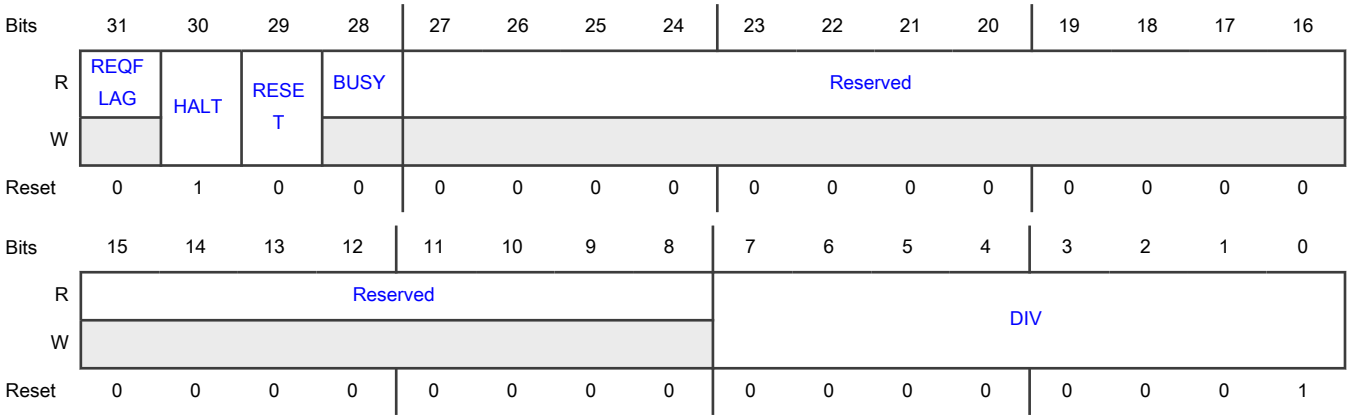
Offset

Register	Offset
SDIO0FCLKDIV	264h

Function

Controls the divider for the SD/MMC/SDIO interface 0 (uSDHC0) functional clock sdio0_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a divider value for SD/MMC/SDIO interface 0: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.21 SDIO1 Functional Clock Source Select (SDIO1FCLKSEL)

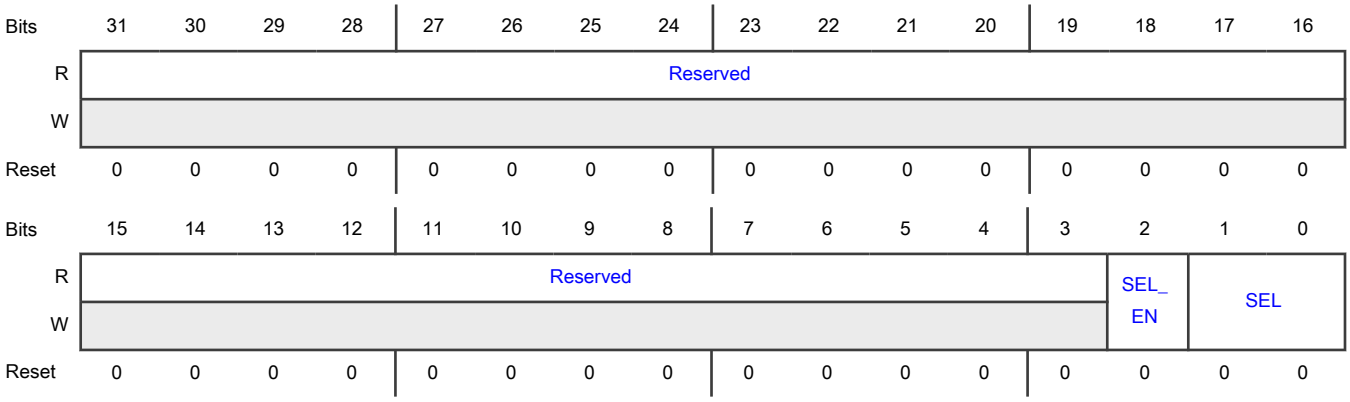
Offset

Register	Offset
SDIO1FCLKSEL	280h

Function

Selects a clock source for the SD/MMC/SDIO interface 1 (uSDHC1).

Diagram



Fields

Field	Function
31-3 —	Reserved
2	Clock Mux Output Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEL_EN	Enables mux output for SD/MMC/SDIO interface 1 to reduce power. 0b - Gates mux output for SD/MMC/SDIO interface 1 clock. 1b - Enables mux output for SD/MMC/SDIO interface 1 clock.
1-0 SEL	SDIO1 Clock Source Select Selects source for SD/MMC/SDIO interface 1. 00b - VDDN_MEDIA base clock 01b - Audio PLL PFD0 clock 10b - fro0_max 11b - Main PLL PFD1 clock

28.7.5.22 SDIO1 Functional Clock Divider (SDIO1FCLKDIV)

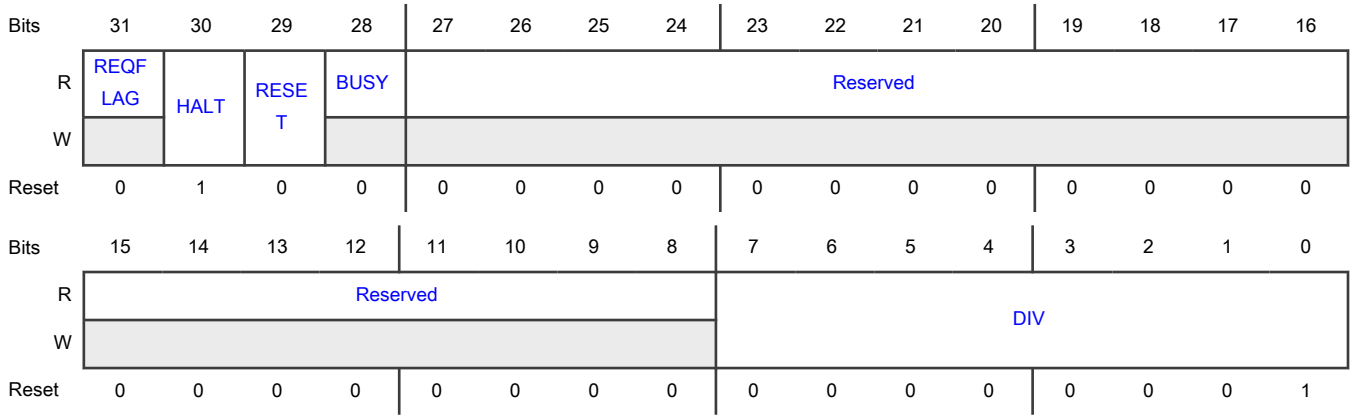
Offset

Register	Offset
SDIO1FCLKDIV	284h

Function

Controls the divider for the SD/MMC/SDIO interface 1 (uSDHC1) functional clock sdio1_fclk.

Diagram



Fields

Field	Function
31	Request Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
REQFLAG	Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for SD/MMC/SDIO interface 1: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.23 MIPI_DSI_Host PHY Clock Source Select (DPHYCLKSEL)

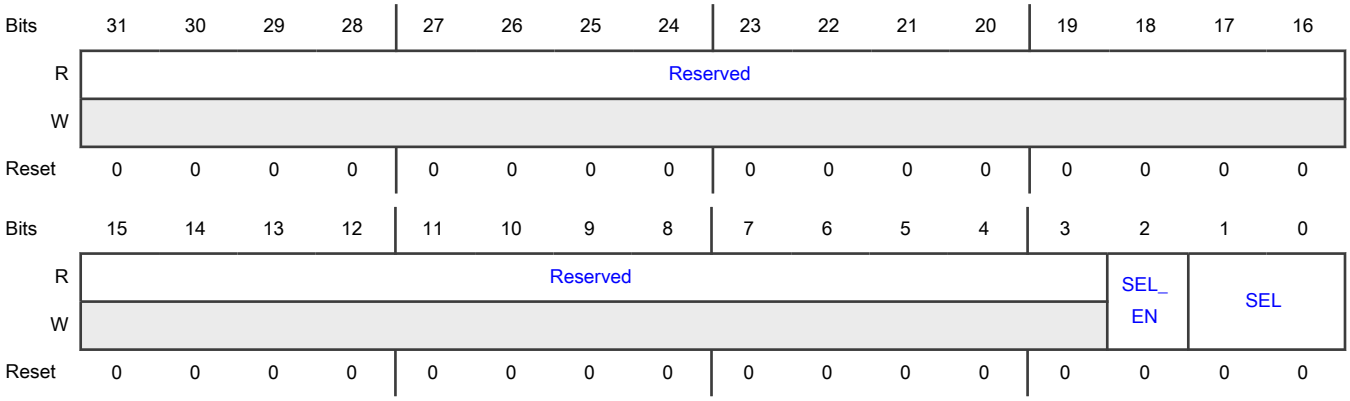
Offset

Register	Offset
DPHYCLKSEL	300h

Function

Selects a clock source for the MIPI_DSI_Host PHY clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for MIPI_DSI_Host PHY clock to reduce power. 0b - Gates mux output for MIPI_DSI_Host PHY clock. 1b - Enables mux output for MIPI_DSI_Host PHY clock.
1-0 SEL	MIPI_DSI_Host PHY Clock Source Select Selects clock source for MIPI_DSI_Host PHY clock. 00b - baseclk_md2 01b - Reserved 10b - fro0_max 11b - audio_pll_pfd2

28.7.5.24 MIPI_DSI_Host PHY Clock Divider (DPHYCLKDIV)

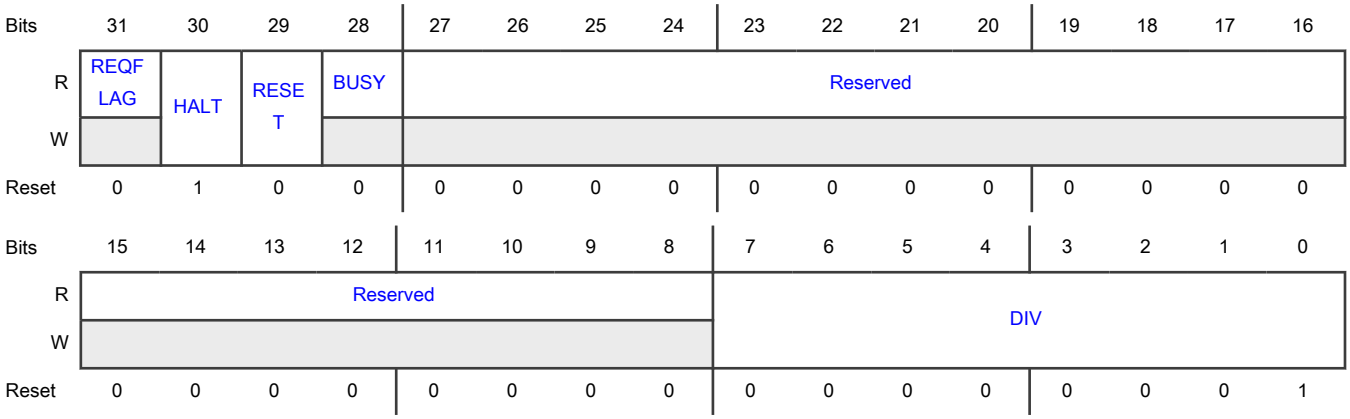
Offset

Register	Offset
DPHYCLKDIV	304h

Function

Controls the divider for the MIPI_DSI_Host PHY clock dphy_bit_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.25 MIPI_DSI_Host DPHY Escape Mode Clock Source Select (DPHYESCCLKSEL)

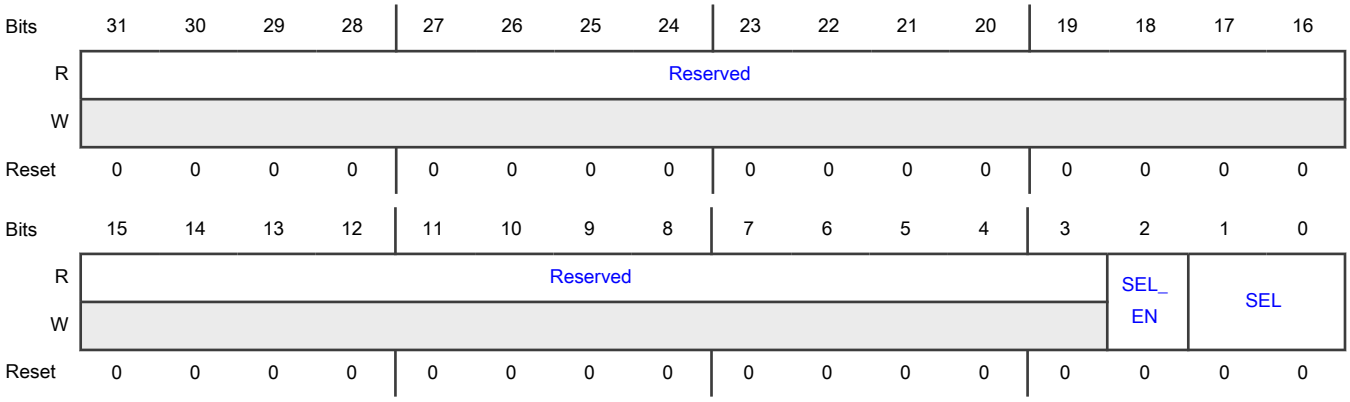
Offset

Register	Offset
DPHYESCCLKSEL	308h

Function

Selects a clock source for the MIPI_DSI_Host DPHY clock in escape mode.

Diagram



Fields

Field	Function
31-3 —	Reserved
2	Clock Mux Output Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEL_EN	Enables mux output for the MIPI_DSI_Host DPHY clock in the escape mode to reduce power. 0b - Gates mux output for MIPI_DSI_Host DPHY clock. 1b - Enables mux output for MIPI_DSI_Host DPHY clock.
1-0 SEL	Clock Source Select Selects a clock source for the MIPI_DSI_Host DPHY clock in the escape mode. 00b - baseclk_md2 01b - main_pll_pfd1 10b - fro0_max 11b - audio_pll_pfd2

28.7.5.26 MIPI_DSI_Host DPHY Escape Mode Receive Clock Divider (DPHYESCRXCLKDIV)

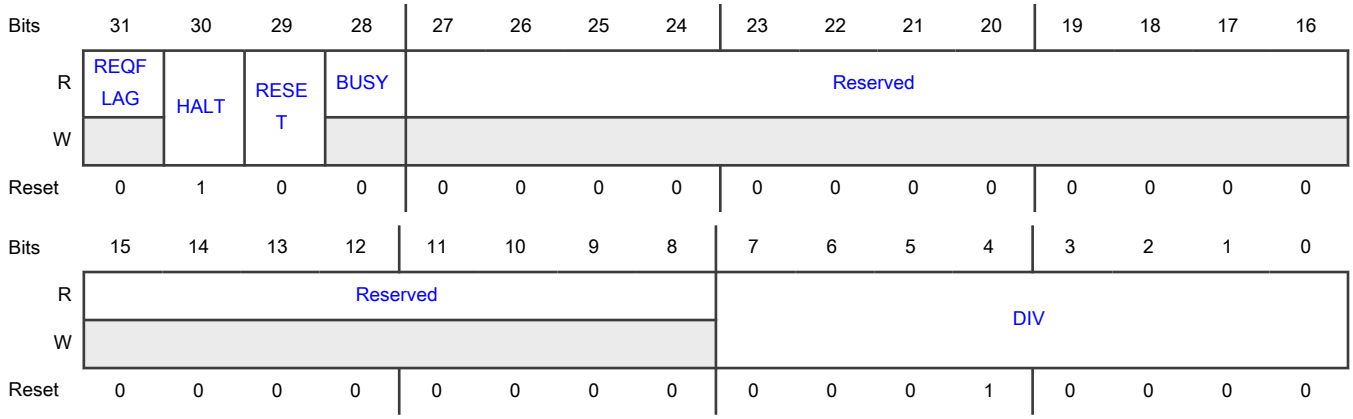
Offset

Register	Offset
DPHYESCRXCLKDIV	30Ch

Function

Controls the divider for the MIPI_DSI_Host DPHY receive clock dphy_rxtx_clk in escape mode.

Diagram



Fields

Field	Function
31	Request Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
REQFLAG	Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.27 MIPI_DSI_Host DPHY Escape Mode Transmit Clock Divider (DPHYESCTXCLKDIV)

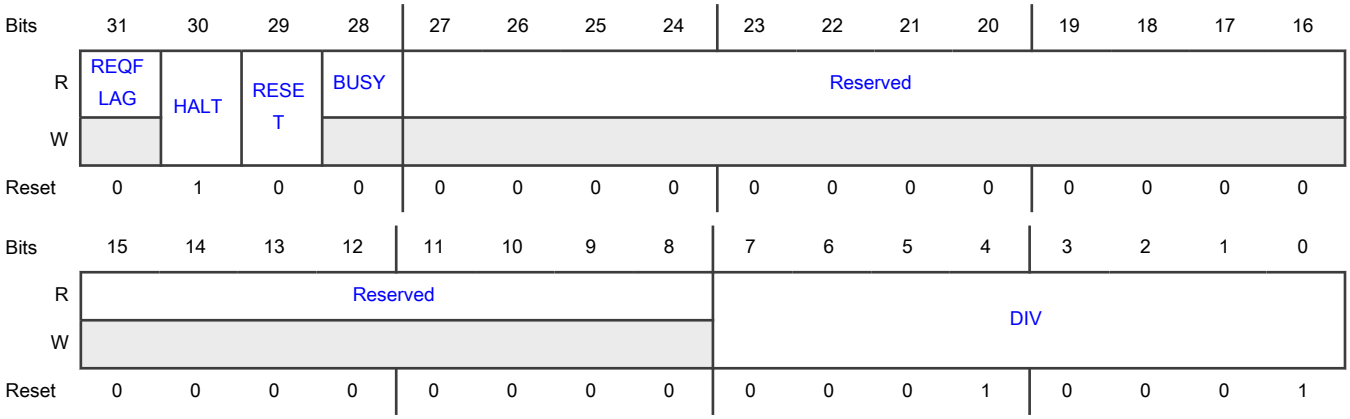
Offset

Register	Offset
DPHYESCTXCLKDIV	310h

Function

Controls the divider for the MIPI_DSI_Host DPHY transmit dphy_rxtx_clk divider in escape mode.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

28.7.5.28 VGPU Clock Source Select (VGPUCLKSEL)

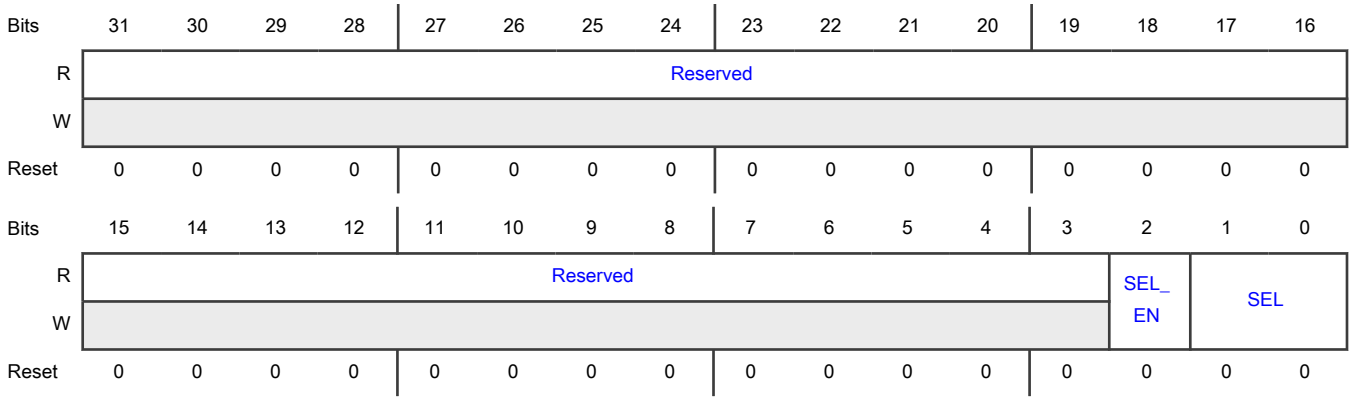
Offset

Register	Offset
VGPUCLKSEL	320h

Function

Selects a clock source for the VGPU clock.

Diagram



Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 SEL_EN	Clock Mux Output Enable Enables mux output for the VGPU clock to reduce power. 0b - Gates mux output for VGPU clock. 1b - Enables mux output for VGPU clock.
1-0 SEL	VGPU Clock Source Select Selects a clock source for the VGPU clock. 00b - baseclk_md2 01b - main_pll_pfd0 10b - fro0_max 11b - main_pll_pfd2

28.7.5.29 VGPU Clock Divider (VGPUCLKDIV)

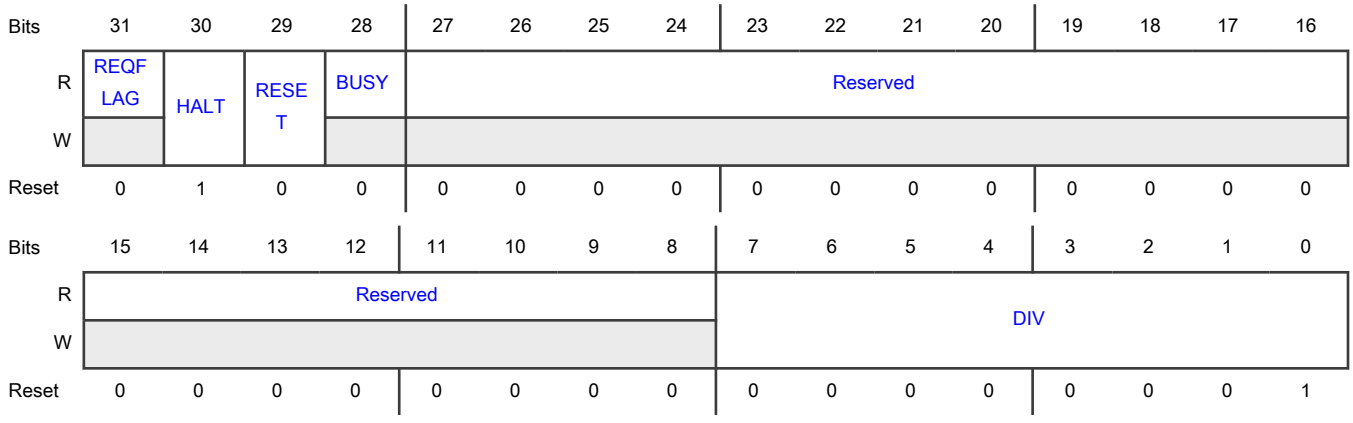
Offset

Register	Offset
VGPUCLKDIV	324h

Function

Controls the divider for the VGPU clock gpu_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for the VGPU clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

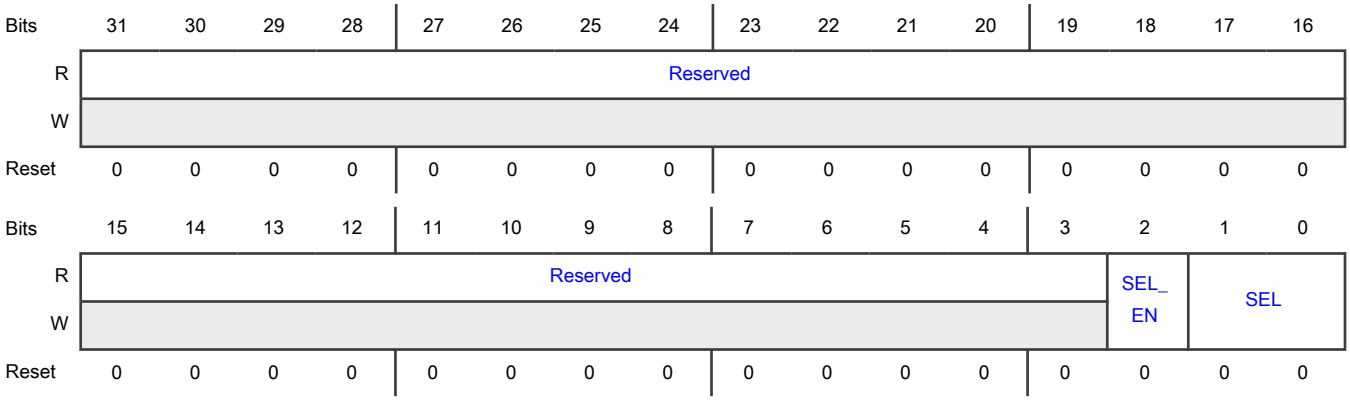
28.7.5.30 LPSPI14 Clock Source Select (LPSPI14CLKSEL)

Offset

Register	Offset
LPSPI14CLKSEL	328h

Function
Selects a clock source for the LPSPI14 clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for the LPSPI14 clock to reduce power. 0b - Gates mux output for LPSPI14 clock. 1b - Enables mux output for LPSPI14 clock.
1-0 SEL	Clock Source Select Selects a clock source for LPSPI14 clock: 00b - baseclk_md2 01b - fro0_max 10b - main_pll_pfd0 11b - fro1_max

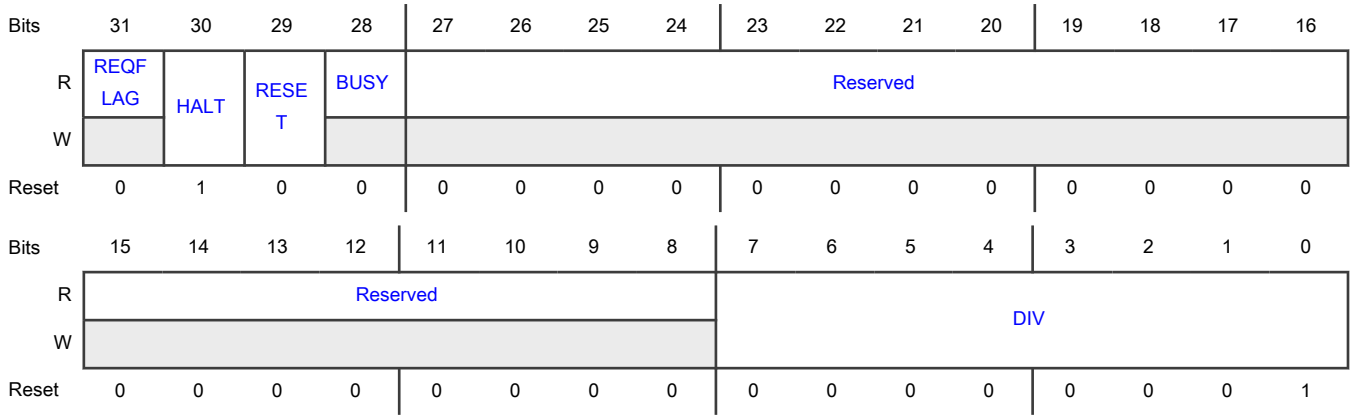
28.7.5.31 LPSPI14 Clock Divider (LPSPI14CLKDIV)

Offset

Register	Offset
LPSPI14CLKDIV	32Ch

Function
Controls the divider for the LPSPI14 clock lpspi14_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for the LPSPI14 clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

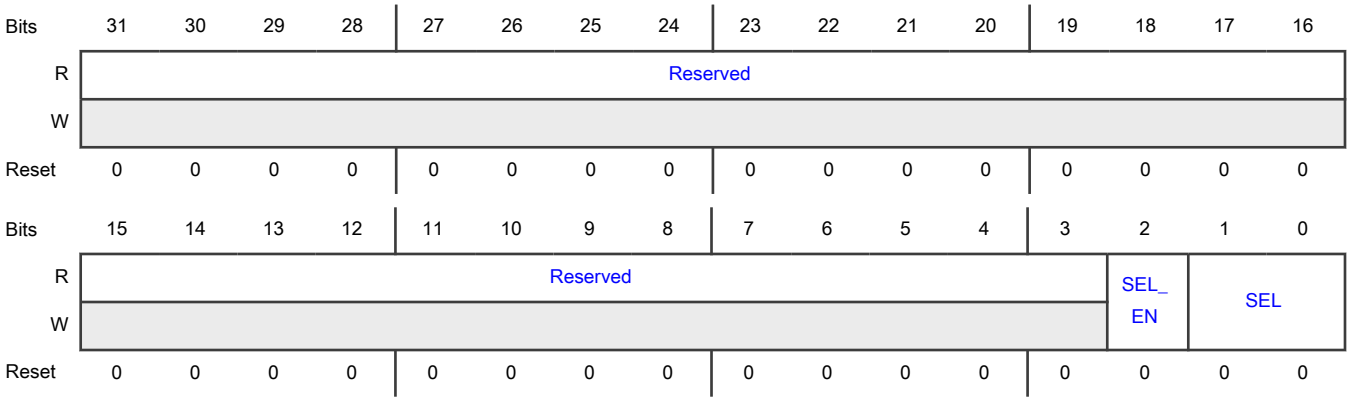
28.7.5.32 LPSPI16 Clock Source Select (LPSPI16CLKSEL)

Offset

Register	Offset
LPSPI16CLKSEL	330h

Function
Selects a clock source for the LPSPI16 clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for the LPSPI16 clock to reduce power. 0b - Gates mux output for LPSPI16 clock. 1b - Enables mux output for LPSPI16 clock.
1-0 SEL	LPSPI16 Clock Source Select Selects a clock source for the LPSPI16 clock. 00b - baseclk_md2 01b - fro0_max 10b - main_pll_pfd0 11b - fro1_max

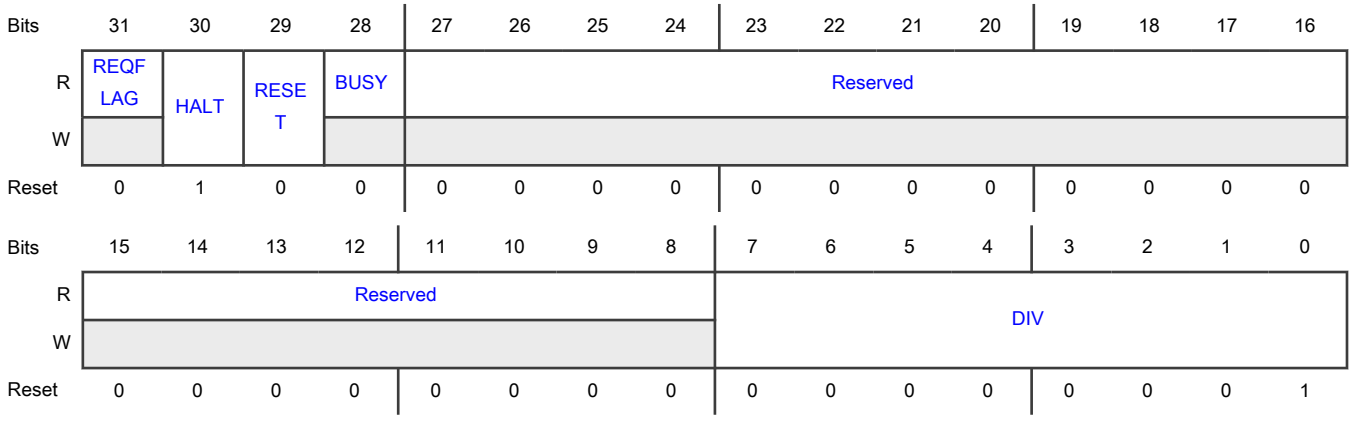
28.7.5.33 LPSPI16 Clock Divider (LPSPI16CLKDIV)

Offset

Register	Offset
LPSPI16CLKDIV	334h

Function
Controls the divider for the LPSPI16 clock lpspi16_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for the LPSPI16 clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

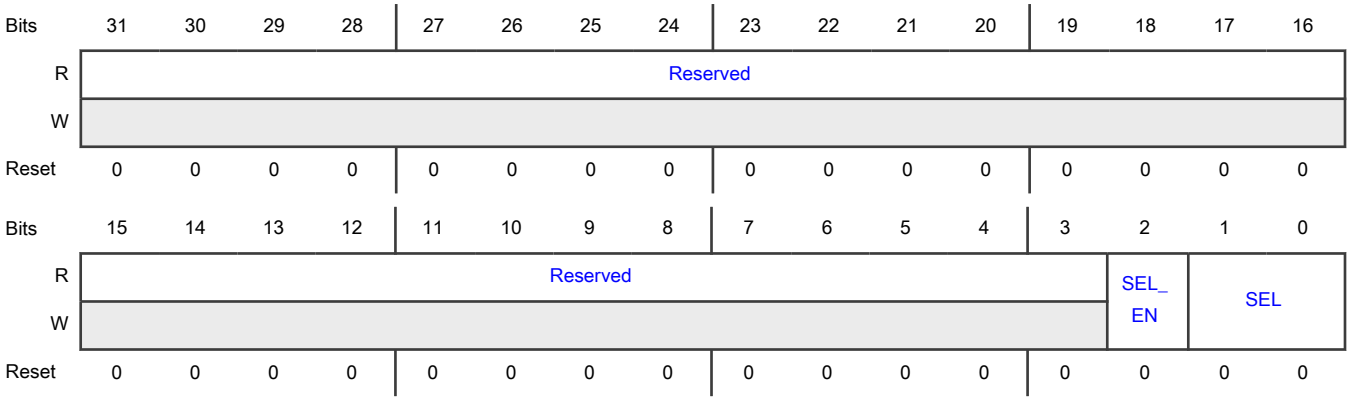
28.7.5.34 FLEXIO Clock Source Select (FLEXIOCLKSEL)

Offset

Register	Offset
FLEXIOCLKSEL	338h

Function
Selects a clock source for the FLEXIO clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for the FLEXIO clock to reduce power. 0b - Gates mux output for FLEXIO clock. 1b - Enables mux output for FLEXIO clock.
1-0 SEL	FLEXIO Clock Source Select Selects a clock source for the FLEXIO clock. 00b - baseclk_md2 01b - fro0_max 10b - fro1_max 11b - main_pll_pfd3

28.7.5.35 FLEXIO Clock Divider (FLEXIOCLKDIV)

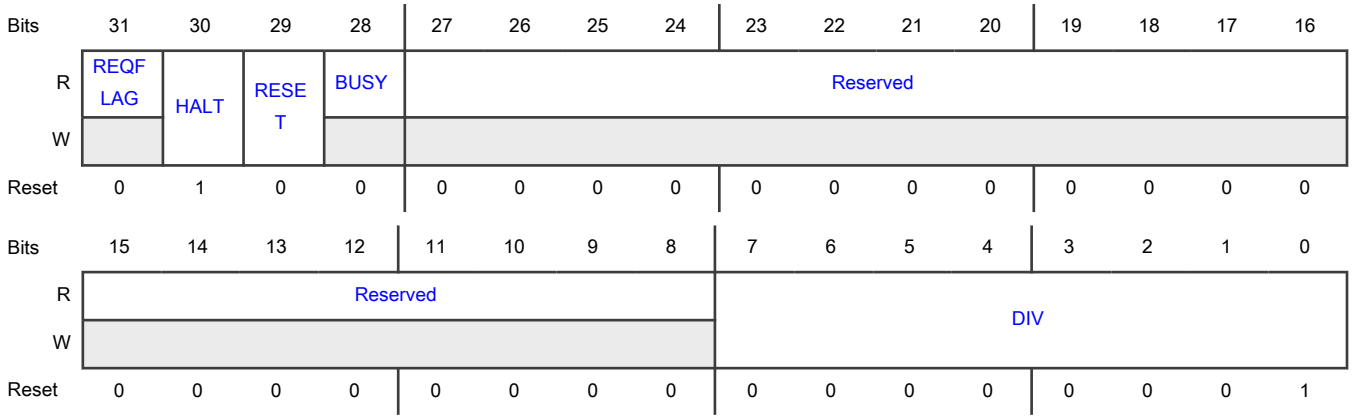
Offset

Register	Offset
FLEXIOCLKDIV	33Ch

Function

Controls the divider for the FLEXIO clock flexio_clk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a divider value for the FLEXIO clock: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

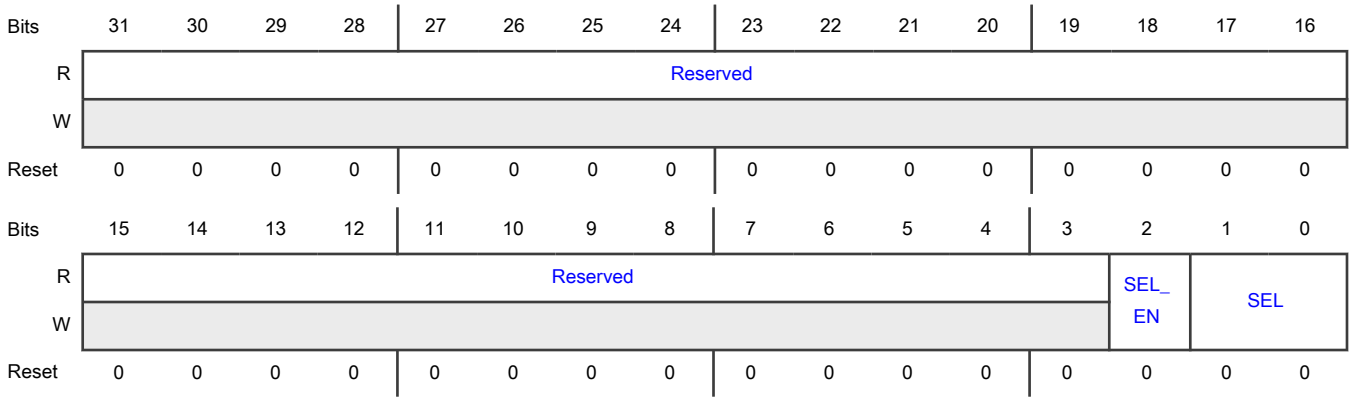
28.7.5.36 LCDIF Pixel Clock Source Select (LCDIFPIXELCLKSEL)

Offset

Register	Offset
LCDIFPIXELCLKSEL	340h

Function
Selects a clock source for the LCDIF pixel clock.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 SEL_EN	Clock Mux Output Enable Enables mux output for the LCDIF pixel clock to reduce power. 0b - Gates mux output for LCDIF clock. 1b - Enables mux output for LCDIF clock.
1-0 SEL	LCDIF Pixel Clock Source Select Selects a clock source for the LCDIF pixel clock. 00b - baseclk_md2 01b - main_pll_pfd2 10b - fro0_max 11b - audio_pll_pfd1

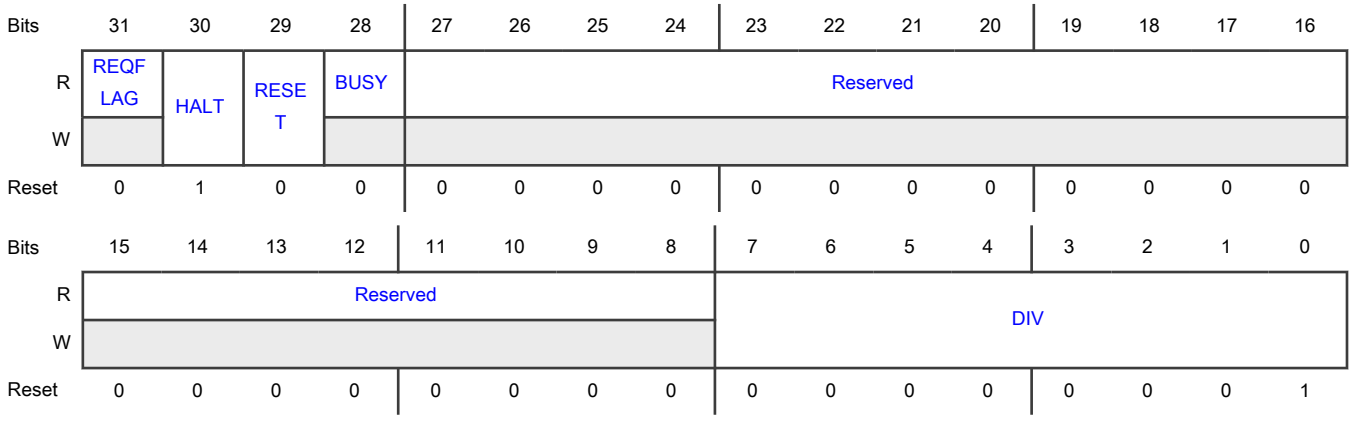
28.7.5.37 LCDIF Pixel Clock Divider (LCDIFPIXELCLKDIV)

Offset

Register	Offset
LCDIFPIXELCLKDIV	344h

Function
Controls the divider for the LCDIF pixel clock dcpixel_fclk.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - A change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down). 0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-8 —	Reserved
7-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 255: Divide by 256

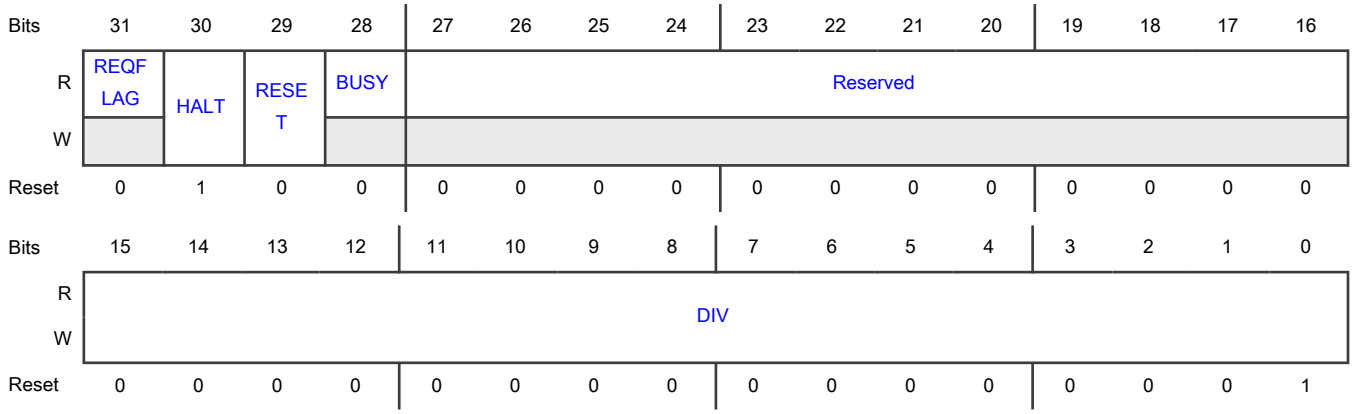
28.7.5.38 Low frequency Clock Divider (LOWFREQCLKDIV)

Offset

Register	Offset
LOWFREQCLKDIV	700h

Function
Controls the divider for the low frequency clock.

Diagram



Fields

Field	Function
31 REQFLAG	Request Flag Indicates the divider status. The clock to be divided must be running for this status to change. 0b - The change to the divider value has been finished. 1b - The change is being made to the divider value.
30 HALT	Divider Counter Halt Stops the divider counter. 0b - No effect 1b - Halts (stops) the divider counter.
29 RESET	Divider Counter Reset Resets the divider counter to use a new divider value, instead of completing the previous count. **NOTE** This is an asynchronous reset (should only be used when you do not care about clean shut down).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Resets the divider counter.
28 BUSY	Busy Flag Indicates the divider status. 0b - The CLKOUT is outputted with the new divider value. 1b - A change is being made to the divider value.
27-16 —	Reserved
15-0 DIV	Clock Divider Value Select Selects a clock divider value: • 0: Divide by 1 • 1: Divide by 2 • 2: Divide by 3 • . . . • 2047: Divide by 2048

28.7.5.39 VDD1_SENSE Media Peripheral Clock Control 0 (PSCCTL0_SENS)

Offset

Register	Offset
PSCCTL0_SENS	810h

Function

Allows software to control specific peripheral's clock. Writing an one to any assigned bit in this register enable the clock and allows the specified peripheral to operate. Writing 0 disables the clock. If VDD2_MEDIA power rail is OFF, this register will be reset after power on.

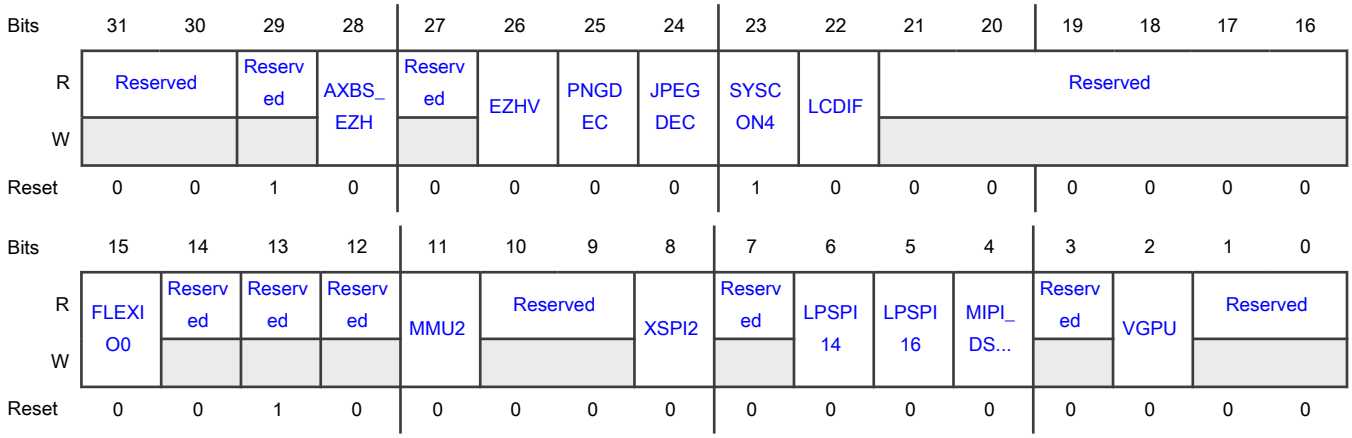
NOTE

This register can only be written by CPU1. CPU0 can not write this register but CPU0 can read it to query the real status of device's clock. CPU1 can write zero to PSCCTL0_SENS to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL0_MEDIA, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS for EZHV Clock Enables AXBS for EZHV clock. 0b - Disable 1b - Enable
27 —	Reserved
26 EZHV	EZHV Clock Enables EZHV clock. 0b - Disable 1b - Enable
25 PNGDEC	PNGDEC Clock Enables PNGDEC clock. 0b - Disable 1b - Enable
24 JPEGDEC	JPEGDEC Clock Enables JPEGDEC clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
23 SYSCON4	SYSCON4 Clock Enables SYSCON4 clock. 0b - Disable 1b - Enable
22 LCDIF	LCDIF Clock Enables LCDIF clock. 0b - Disable 1b - Enable
21-16 —	Reserved
15 FLEXIO0	FLEXIO0 Clock Enables FLEXIO0 clock. 0b - Disable 1b - Enable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Enables MMU2 clock. 0b - Disable 1b - Enable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Enables XSPI2 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Enables LPSPI14 clock. 0b - Disable 1b - Enable
5 LPSPI16	LPSPT16 Clock Enables LPSPI16 clock. 0b - Disable 1b - Enable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Enables MIPI_DSI_HOST clock. 0b - Disable 1b - Enable
3 —	Reserved
2 VGPU	VGPU Clock Enables VGPU clock. 0b - Disable 1b - Enable
1-0 —	Reserved

28.7.5.40 VDD1_SENSE Media Peripheral Clock Control 1 (PSCCTL1_SENS)

Offset

Register	Offset
PSCCTL1_SENS	814h

Function

Allows software to control specific peripheral's clock. Writing a one to 1 any assigned bit in this register enable the clock and allows the specified peripheral to operate. Writing 0 disables the clock. If VDD2_MEDIA power rail is OFF, this register will be reset after power on.

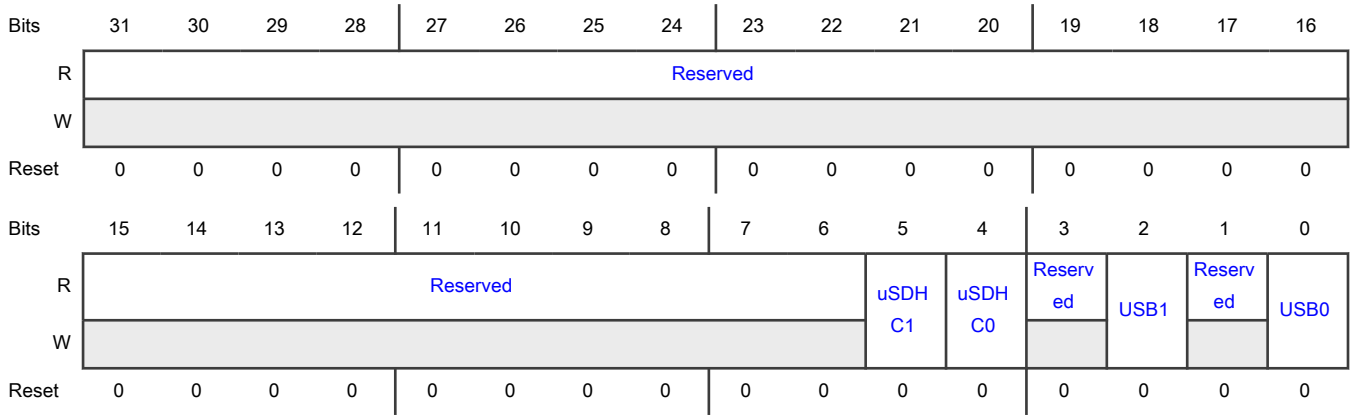
NOTE

This register can only be written by CPU1. CPU0 can not write this register but CPU0 can read it to query the really status of device's clock. CPU1 can write zero to PSCCTL1_SENS to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL1_MEDIA, since if one core enables the clock, it cannot be disabled by another core.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Clock Enables uSDHC1 clock. 0b - Disable 1b - Enable
4 uSDHC0	uSDHC0 Clock Enables uSDHC0 clock. 0b - Disable 1b - Enable
3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 USB1	USB1 Clock Enables USB1 clock. 0b - Disable 1b - Enable
1 —	Reserved
0 USB0	USB0 Clock Enables USB0 clock. 0b - Disable 1b - Enable

28.7.5.41 VDD1_SENSE Media Peripheral Clock Control 0 Set (PSCCTL0_SENS_SET)

Offset

Register	Offset
PSCCTL0_SENS_SET	840h

Function

Writing an one to a bit position in PSCCTL0_SENS_SET sets the corresponding position in PSCCTL0_SENS.

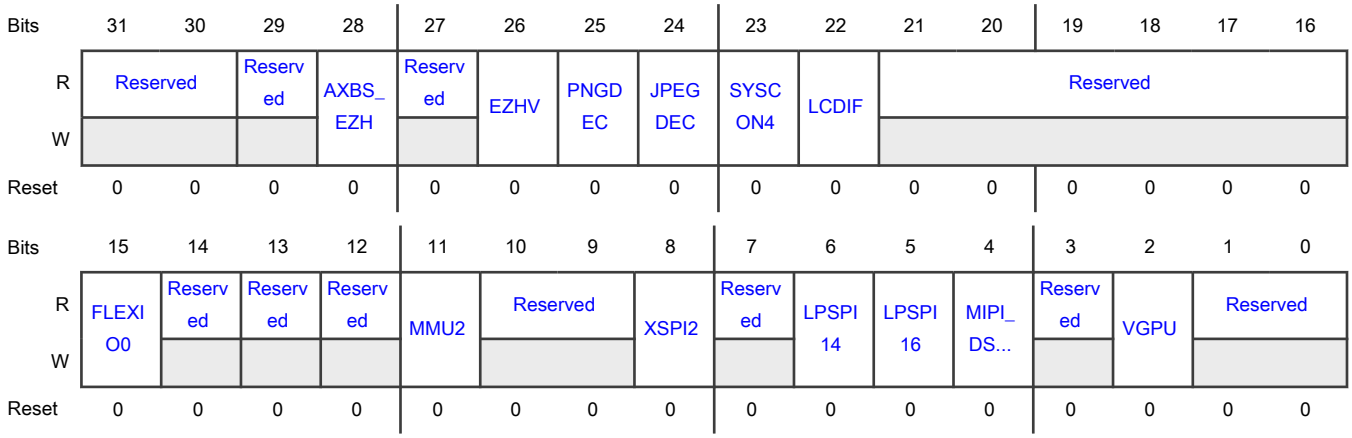
NOTE

This register can only be written by CPU1. CPU0 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS for EZHV Clock Enables AXBS for EZHV clock. 0b - No effect 1b - Enable
27 —	Reserved
26 EZHV	EZHV Clock Enables EZHV clock. 0b - No effect 1b - Enable
25 PNGDEC	PNGDEC Clock Enables PNGDEC clock. 0b - No effect 1b - Enable
24 JPEGDEC	JPEGDEC Clock Enables JPEGDEC clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Enable
23 SYSCON4	SYSCON4 Clock Enables SYSCON4 clock. 0b - No effect 1b - Enable
22 LCDIF	LCDIF Clock Enables LCDIF clock. 0b - No effect 1b - Enable
21-16 —	Reserved
15 FLEXIO0	FLEXIO0 Clock Enables FLEXIO0 clock. 0b - No effect 1b - Enable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Enables MMU2 clock. 0b - No effect 1b - Enable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Enables XSPI2 clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Enable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Enables LPSPI14 clock. 0b - No effect 1b - Enable
5 LPSPI16	LPSPI16 Clock Enables LPSPI16 clock. 0b - No effect 1b - Enable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Enables MIPI_DSI_HOST clock. 0b - No effect 1b - Enable
3 —	Reserved
2 VGPU	VGPU Clock Enables VGPU clock. 0b - No effect 1b - Enable
1-0 —	Reserved

28.7.5.42 VDD1_SENSE Media Peripheral Clock Control 1 Set (PSCCTL1_SENS_SET)

Offset

Register	Offset
PSCCTL1_SENS_SET	844h

Function

Writing an one to a bit position in PSCCTL1_SENS_SET sets the corresponding position in PSCCTL1_SENS.

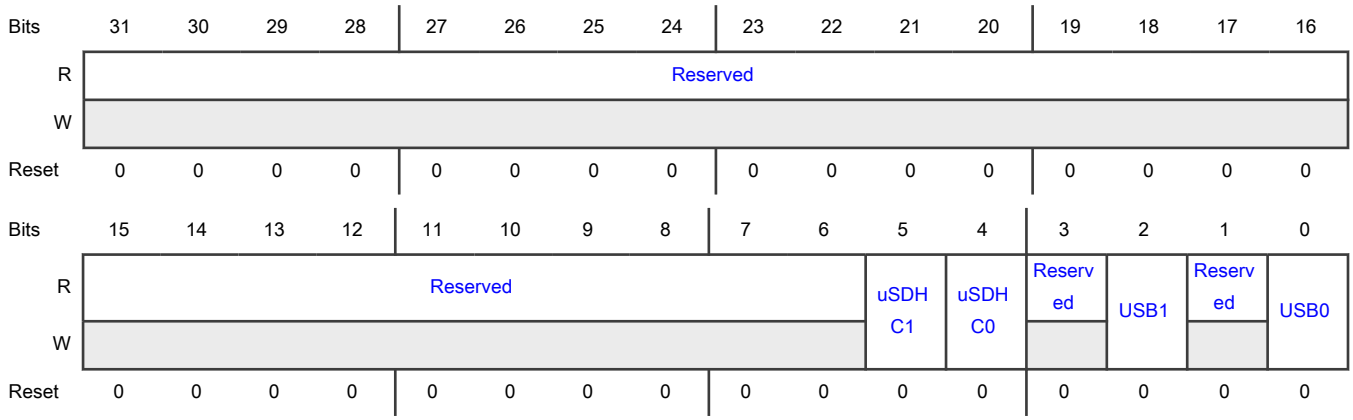
NOTE

This register can only be written by CPU1. CPU0 can not write this register.

NOTE

After enabling the module clock, the software should perform dummy read on the related register field for the clock enablement to be properly propagated.

Diagram



Fields

Field	Function
31-6 —	Reserved Slot reserved
5 uSDHC1	uSDHC1 Clock Enables uSDHC1 clock. 0b - No effect 1b - Enable
4 uSDHC0	uSDHC0 Clock Enables uSDHC0 clock. 0b - No effect 1b - Enable
3 —	Reserved
2 USB1	USB1 Clock Enables USB1 clock. 0b - No effect 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 —	Reserved
0 USB0	USB0 Clock Enables USB0 clock. 0b - No effect 1b - Enable

28.7.5.43 VDD1_SENSE Media Peripheral Clock Control 0 Clear (PSCCTL0_SENS_CLR)

Offset

Register	Offset
PSCCTL0_SENS_CLR	870h

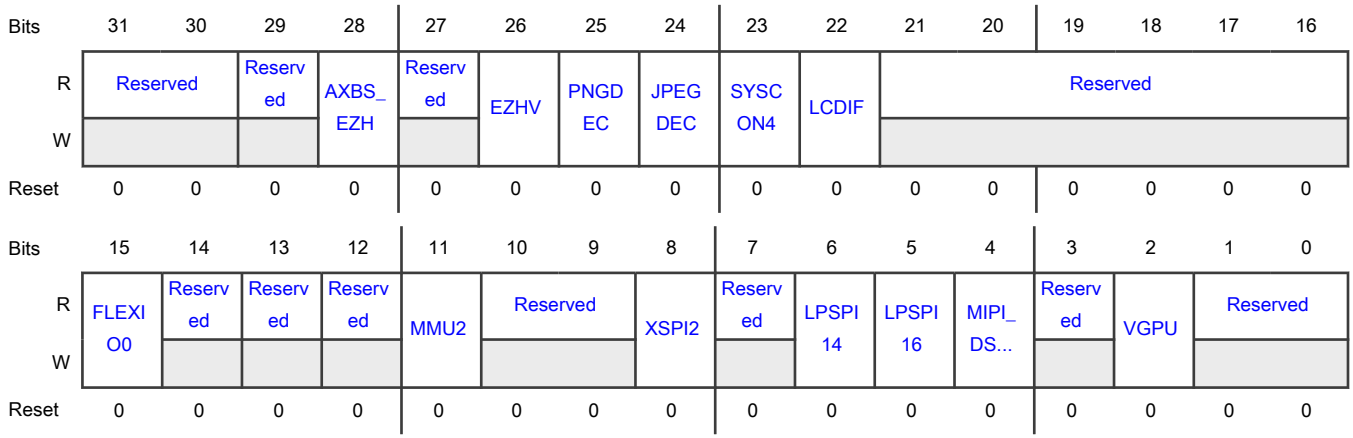
Function

Writing an one to a bit position in PSCCTL0_SENS_CLR clears the corresponding position in PSCCTL0_SENSE.

NOTE

This register can only be written by CPU1, CPU0 should not write this register. CPU1 can write 1 to PSCCTL1_SENS_CLR to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL0_MEDIA, since if one core enables the clock, it cannot be disabled by another core.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 AXBS_EZH	AXBS_EZHV Clock Disables EZH clock. 0b - No effect 1b - Disable
27 —	Reserved
26 EZHV	EZHV Clock Disables EZHV clock. 0b - No effect 1b - Disable
25 PNGDEC	PNG Decoder Clock Disables PNG decoder clock. 0b - No effect 1b - Disable
24 JPEGDEC	JPEGDEC Clock Disables JPEGDEC clock. 0b - No effect 1b - Disable
23 SYSCON4	SYSCON4 Clock Disables SYSCON4 clock. 0b - No effect 1b - Disable
22 LCDIF	LCDIF Clock Disables LCDIF clock. 0b - No effect 1b - Disable
21-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
15 FLEXIO0	FLEXIO0 Clock Disables FLEXIO0 clock control 0b - No effect 1b - Disable
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 MMU2	MMU2 Clock Disables MMU2 clock. 0b - No effect 1b - Disable
10-9 —	Reserved
8 XSPI2	XSPI2 Clock Disables XSPI2 clock. 0b - No effect 1b - Disable
7 —	Reserved
6 LPSPI14	LPSPI14 Clock Disables LPSPI14 clock. 0b - No effect 1b - Disable
5 LPSPI16	LPSPI16 Clock Disables LPSPI16 clock. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable
4 MIPI_DSI_HOST	MIPI_DSI_HOST Clock Disables MIPI_DSI_HOST clock. 0b - No effect 1b - Disable
3 —	Reserved
2 VGPU	VGPU Clock Disables VGPU clock. 0b - No effect 1b - Disable
1-0 —	Reserved

28.7.5.44 VDD1_SENSE Media Peripheral Clock Control 1 Clear (PSCCTL1_SENS_CLR)

Offset

Register	Offset
PSCCTL1_SENS_CLR	874h

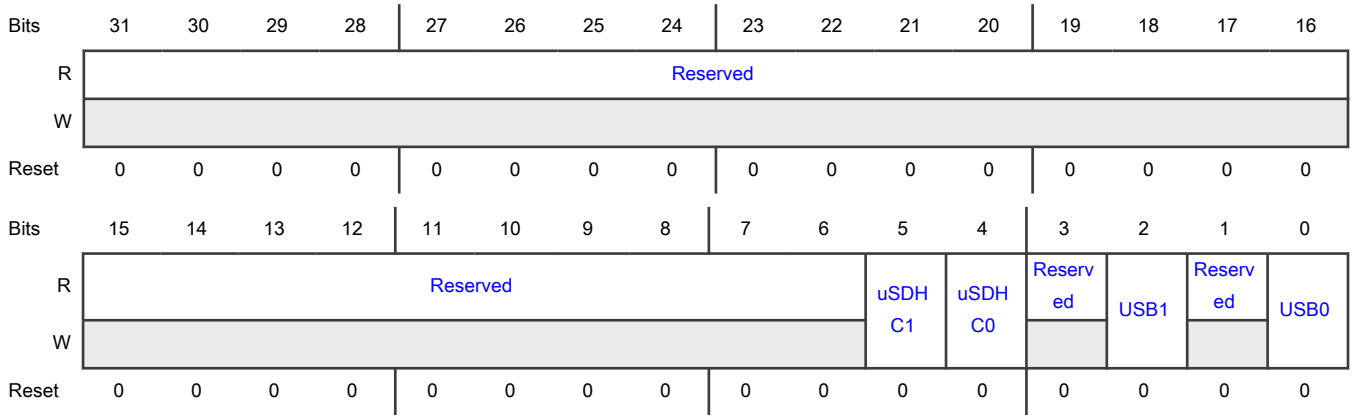
Function

Writing an one to a bit position in PSCCTL1_SENS_CLR clears the corresponding position in PSCCTL1_SENSE.

NOTE

This register can only be written by CPU1, CPU0 should not write this register. CPU1 can write 1 to PSCCTL0_SENS_CLR to disable the clock. But whether this clock is really disabled software need to read back the related bit from PSCCTL1_MEDIA, since if one core enables the clock, it cannot be disabled by another core.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Clock Disables uSDHC1 clock. 0b - No effect 1b - Disable
4 uSDHC0	uSDHC0 Clock Disables uSDHC0 clock. 0b - No effect 1b - Disable
3 —	Reserved
2 USB1	USB1 Clock Disables USB1 clock. 0b - No effect 1b - Disable
1 —	Reserved
0 USB0	USB0 Clock Disables USB0 clock. 0b - No effect 1b - Disable

28.7.5.45 One Source Clock Slice Enable for VDD1_SENSE Core (ONE_SRC_CLKSLICE_ENABLE_SENSE)

Offset

Register	Offset
ONE_SRC_CLKSLICE_ENABLE_SENSE	890h

Function

Controls the gating of one source clock slice. Multi source clock slices gating is controlled by enabling the SEL_EN bit field of a series of clock select source registers, for example, XSPI*n*CLKSEL.

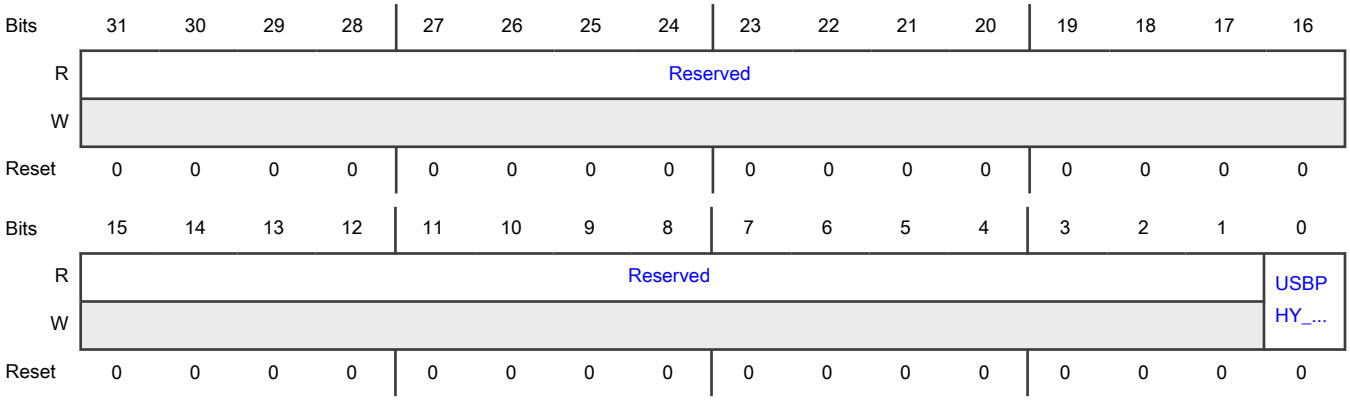
NOTE

Not all the clock select source registers have this SEL_EN bit for some clocks cannot be gated, such as base clock.

NOTE

This register is VDD1_SENSE core access only.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 USBPHY_REFCLK_EN	USBPHY0 Reference Clock Enables for USBPHY0 reference clock. This clock's source is osc_clk, please make sure the source clock is active before enabling the USBPHY0 reference clock. 0b - Gates USBPHY0 reference clock. 1b - Enables USBPHY0 reference clock.

Chapter 29
System Controller (SYSCON)

29.1 Chip-specific SYSCON information

Table 246. Reference links to related information

Topic	Related module	Reference
Full description	SYSCON	SYSCON
Peripheral memory map		Memory map overview

NOTE

The SYSCON module includes some security functionality that is not described here. Refer to the i.MX RT700 Security Reference Manual for more details.

29.1.1 Module instances

This device has five instances of the SYSCON modules.

Table 247. SYSCON instances

	SYSCON0	SYSCON1	SYSCON2	SYSCON3	SYSCON4
Located in	VDD2_COM	VDD1_SENSE	VDDN_COM	VDD1_SENSE	VDD2_MEDIA
Controls modules in	VDD2_COMP, VDD2_COM, VDD2_DSP	VDD1_SENSE (private region)	VDDN_COM	VDD1_SENSE (shared region)	VDDN_MEDIA, VDD2_MEDIA
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

29.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

29.1.3 GLIKEY protection

See GLIKEY protection for SYSCON registers that are protected by GLIKEY.

29.2 Overview

SYSCON configures buses, memory access, clocks, security, GPIO, core communication, interrupts, and associated peripheral control to these functions on the chip.

29.2.1 Block diagram

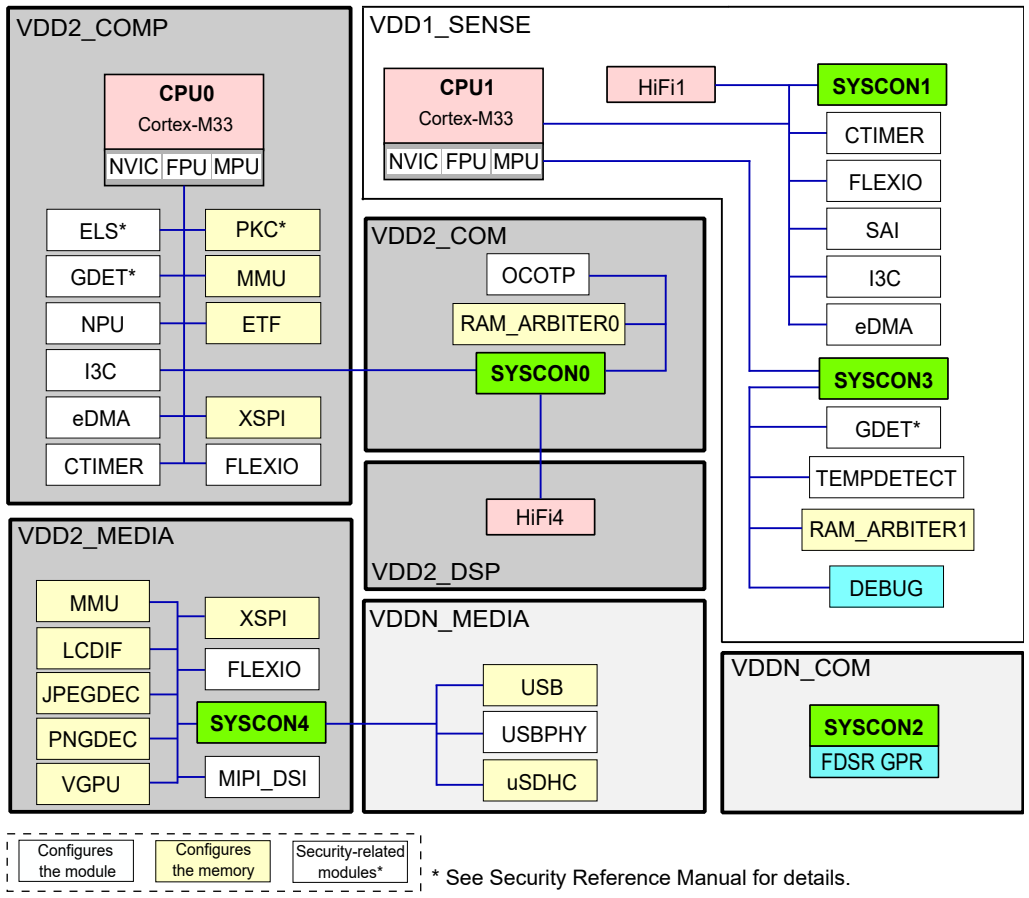


Figure 101. SYSCON block diagram

29.2.2 Features

SYSCON supports the following features and configurations:

- System configuration
  - CPU control and status (SYSCON0, 3)
  - EZH-V control (SYSCON4)
  - Non-Maskable Interrupt (NMI) source select (SYSCON0, 1)
  - Calibration of system tick timer (SYSCON0, 1)
- Memory configuration
  - Memory write mode configuration and speed trim (SYSCON0, 4)
  - Low-power control (SYSCON0, 1, 3, 4)
- Peripherals configuration
  - eDMA request configuration (SYSCON0, 1,)
  - Counter Timer (CTIMER) global start (SYSCON0, 1)
- Temperature detector (TEMPDETECT) controller (SYSCON3)
  - TEMPDETECT control

— TEMPDETECT interrupt set and clear

- General Purpose Input Output (GPIO) control
  - GPIO configuration (SYSCON0, 1)
- EZH-V and Arm core communication (SYSCON4)
  - Arm core to EZH-V interrupt control
  - EZH-V to Arm core interrupt control
- Gray-to-binary converter (SYSCON0, 3)
  - Allows decoding gray value coming from OS Event Timer (OSTIMER)

29.3 Functional description

29.3.1 TEMPDETECT controller

TEMPDETECT delivers a digital output that indicates if the junction temperature is above or below a given temperature threshold. See Temperature threshold. TEMPDETECT would normally be employed for the purpose of threshold.

Whenever the chip temperature crosses that temperature threshold, the sensor immediately signals that event to enable the system to react.

There are two programmable TEMPDETECTs in VDD1_SENSE domain, and each sensor generates a dedicated interrupt, as shown in the block diagram below.

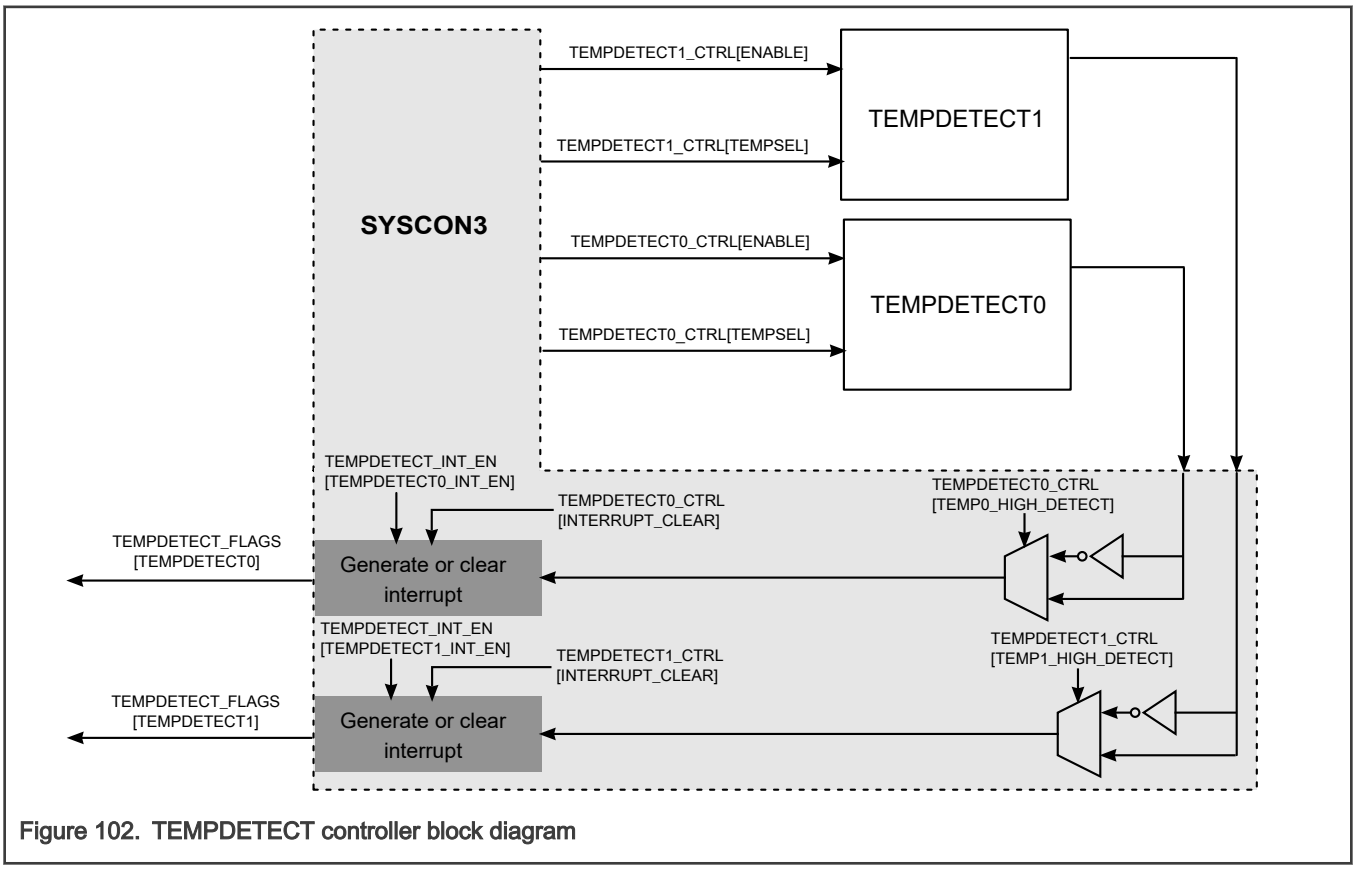


Figure 102. TEMPDETECT controller block diagram

29.3.1.1 Temperature threshold

The temperature threshold value is programmable from -45°C to +130°C.

NOTE

Ambient temperature range for the chip is -30°C to +85°C.

The approximate value of the temperature threshold $T_{TSH_{TYP}}$ (less than 0.25°C error) is calculated using the following polynomial regression:

$T_{TSH_{TYP}}(sel) = 0.000002673049*sel^3 + 0.0003734262*sel^2 + 0.4487042*sel - 46.98694$

Values of $T_{TSH_{TYP}}$ in function of TEMPSEL[7:0] (sel) are shown in Table 248.

Table 248. Temperature threshold values in function of TEMPSEL[7:0]

TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)
0	-46.75	64	-16.14	128	22.26	192	71.76
1	-46.34	65	-15.55	129	22.96	193	72.66
2	-45.85	66	-15.04	130	23.66	194	73.56
3	-45.45	67	-14.54	131	24.35	195	74.46
4	-45.04	68	-13.95	132	25.05	196	75.36
5	-44.55	69	-13.44	133	25.66	197	76.26
6	-44.15	70	-12.85	134	26.36	198	77.16
7	-43.74	71	-12.34	135	27.06	199	78.06
8	-43.25	72	-11.75	136	27.76	200	78.96
9	-42.84	73	-11.24	137	28.46	201	79.86
10	-42.35	74	-10.73	138	29.16	202	80.77
11	-41.95	75	-10.14	139	29.86	203	81.76
12	-41.45	76	-9.55	140	30.56	204	82.66
13	-41.05	77	-9.04	141	31.26	205	83.56
14	-40.55	78	-8.45	142	31.96	206	84.56
15	-40.15	79	-7.94	143	32.75	207	85.46
16	-39.65	80	-7.34	144	33.46	208	86.46
17	-39.24	81	-6.75	145	34.16	209	87.36
18	-38.75	82	-6.24	146	34.86	210	88.36
19	-38.34	83	-5.65	147	35.65	211	89.26
20	-37.85	84	-5.05	148	36.36	212	90.26
21	-37.44	85	-4.54	149	37.06	213	91.26
22	-36.95	86	-3.94	150	37.85	214	92.17
23	-36.45	87	-3.35	151	38.56	215	93.16
24	-36.04	88	-2.75	152	39.27	216	94.16
25	-35.55	89	-2.24	153	40.06	217	95.16

Table continues on the next page...

Table 248. Temperature threshold values in function of TEMPSEL[7:0] (continued)

TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)
26	-35.05	90	-1.64	154	40.76	218	96.16
27	-34.64	91	-1.04	155	41.56	219	97.16
28	-34.15	92	-0.44	156	42.26	220	98.16
29	-33.65	93	0.16	157	43.06	221	99.16
30	-33.24	94	0.76	158	43.86	222	100.20
31	-32.74	95	1.35	159	44.56	223	101.20
32	-32.24	96	1.96	160	45.36	224	102.20
33	-31.75	97	2.56	161	46.16	225	103.30
34	-31.25	98	3.16	162	46.95	226	104.30
35	-30.84	99	3.76	163	47.66	227	105.30
36	-30.34	100	4.36	164	48.46	228	106.40
37	-29.84	101	4.96	165	49.26	229	107.40
38	-29.35	102	5.56	166	50.06	230	108.40
39	-28.85	103	6.16	167	50.86	231	109.50
40	-28.35	104	6.76	168	51.66	232	110.60
41	-27.85	105	7.37	169	52.46	233	111.60
42	-27.35	106	8.05	170	53.26	234	112.70
43	-26.94	107	8.66	171	54.06	235	113.80
44	-26.44	108	9.26	172	54.86	236	114.80
45	-25.94	109	9.86	173	55.66	237	115.90
46	-25.44	110	10.55	174	56.46	238	117.00
47	-24.94	111	11.16	175	57.26	239	118.10
48	-24.43	112	11.76	176	58.15	240	119.20
49	-23.93	113	12.45	177	58.96	241	120.30
50	-23.35	114	13.06	178	59.76	242	121.40
51	-22.85	115	13.66	179	60.56	243	122.50
52	-22.35	116	14.36	180	61.46	244	123.60
53	-21.85	117	14.96	181	62.26	245	124.80
54	-21.35	118	15.66	182	63.07	246	125.90
55	-20.84	119	16.26	183	63.96	247	127.00
56	-20.34	120	16.96	184	64.86	248	128.20

Table continues on the next page...

Table 248. Temperature threshold values in function of TEMPSEL[7:0] (continued)

TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)	TEMPSEL [7:0]	$T_{TSH_{TYP}}$ (°C)
57	-19.84	121	17.56	185	65.66	249	129.30
58	-19.25	122	18.26	186	66.56	250	130.50
59	-18.75	123	18.95	187	67.36	251	131.60
60	-18.24	124	19.56	188	68.26	252	132.80
61	-17.74	125	20.26	189	69.07	253	133.90
62	-17.15	126	20.95	190	69.96	254	135.10
63	-16.65	127	21.56	191	70.86	255	136.20

29.3.1.2 TEMPDETECT configuration

All temperature detector inputs should be properly set before TEMPDETECT is enabled. During normal operation (TEMPDETECT enabled), system application can change the temperature threshold value but should not change any other configuration.

TEMPDETECT is enabled by setting TEMPDETECT0_CTRL[ENABLE] = 1 or TEMPDETECT1_CTRL[ENABLE] = 1. When TEMPDETECT is enabled, its output state only becomes valid after 200 µs. If TEMPDETECT is enabled, when the temperature threshold selection is changed, the output state only becomes valid after 200 µs.

29.3.2 GLIKEY protection

GLIKEY controls the access to some security sensitive registers on the chip, which includes the write-access and reset of these registers.

Below SYSCON registers are protected by GLIKEY by providing a 16-bit write enable input and a 16-bit reset input. Before accessing a SYSCON register, you must obtain the index bit high of the GLIKEY write enable inputs. Also, the SYSCON registers will be reset if the GLIKEY reset input is high.

Table 249. GLIKEY protected registers

Offset	Registers or bitfields	GLIKEY index	GLIKEY reset value
SYSCON0 registers protected by GLIKEY3			
130h	MMU0 Memory Control (MMU0_MEM_CTRL)	4h	NC[1]
140h	eDMA0 Memory Control (EDMA0_MEM_CTRL)	4h	NC[1]
144h	eDMA1 Memory Control (EDMA1_MEM_CTRL)	4h	NC[1]
14Ch	ETF Memory Control (ETF_MEM_CTRL)	4h	NC[1]
150h	MMU1 Memory Control (MMU1_MEM_CTRL)	4h	NC[1]
154h	XSPI0 Memory Control (XSPI0_MEM_CTRL)	4h	NC[1]
158h	XSPI1 Memory Control (XSPI1_MEM_CTRL)	4h	NC[1]
15Ch	CACHE64_CTRL0 Data Memory Control (XSPI0_DATA_MEM_CTRL)	4h	NC[1]
160h	CACHE64_CTRL1 Data Memory Control (XSPI1_DATA_MEM_CTRL)	4h	NC[1]
164h	NPU Memory Control (NPU_MEM_CTRL)	4h	NC[1]

Table continues on the next page...

Table 249. GLIKEY protected registers (continued)

Offset	Registers or bitfields	GLIKEY index	GLIKEY reset value
174h	CPU0 Memory Data Control (CM33_MEM_DATA_CTRL)	4h	NC[1]
178h	CPU0 Memory Tag Control (CM33_MEM_TAG_CTRL)	4h	NC[1]
208h	HiFi4 Memory Control (HIFI4_MEM_CTL)	4h	NC[1]
25Ch	CACHE64_CTRL0 Tag Memory Control (XSPI0_TAG_MEM_CTRL)	4h	NC[1]
260h	CACHE64_CTRL1 Tag Memory Control (XSPI1_TAG_MEM_CTRL)	4h	NC[1]
SYSCON1 registers protected by GLIKEY4			
144h	eDMA2 Memory Control (EDMA2_MEM_CTRL)	4h	NC[1]
148h	eDMA3 Memory Control (EDMA3_MEM_CTRL)	4h	NC[1]
SYSCON4 registers protected by GLIKEY5			
28h	MMU2 Memory Control (MMU2_MEM_CTRL)	4h	NC[1]
2Ch	XSPI2 Memory Control (XSPI2_MEM_CTRL)	4h	NC[1]
34h	LCDIF Memory Control (LCDIF_MEM_CTRL)	4h	NC[1]
3Ch	JPEGDEC Memory Control (JPEGDEC_MEM_CTRL)	4h	NC[1]
40h	PNGDEC Memory Control (PNGDEC_MEM_CTRL)	4h	NC[1]
48h	MIPI Memory Control (MIPI_MEM_CTRL)	4h	NC[1]
4Ch	VGPU Memory Control (VGPU_MEM_CTRL)	4h	NC[1]
510h	USB0 Memory Control (USB0_MEM_CTRL)	4h	NC[1]
514h	USB1 Memory Control (USB1_MEM_CTRL)	4h	NC[1]
518h	uSDHC0 Memory Control (USDHC0_MEM_CTRL)	4h	NC[1]
51Ch	uSDHC1 Memory Control (USDHC1_MEM_CTRL)	4h	NC[1]

1. NC = No connection. GLIKEY won't reset this register.

29.3.3 Clocking

Each SYSCON instance receives a single functional clock.

Table 250. SYSCON clocks

Clock	Description
Peripheral clock	Used for register access and related control logic. For information on generation and gating of this clock, see the CLKCTL chapter.

29.3.4 Interrupts

Table 251. SYSCON interrupts

Flag	Description
TEMPDETECT0 Interrupt	Flag generated from TEMPDETECT0, when the temperature is above or below the threshold
TEMPDETECT1 Interrupt	Flag generated from TEMPDETECT1, when the temperature is above or below the threshold

29.4 External signals

This module has no external signals.

29.5 Initialization

All SYSCON instances are enabled by default, no initialization is required.

The following table lists the registers in CLKCTL to enable the clock to each SYSCON instance.

Table 252. SYSCON clock enable register

SYSCON instance	CLKCTL instance	Clock enable register
SYSCON0	CLKCTL0	PSCCTL0[SYSCON0]
SYSCON1	CLKCTL1	PSCCTL0[SYSCON1]
SYSCON2	CLKCTL2	PSCCTL0[SYSCON2]
SYSCON3	CLKCTL3	PSCCTL0_COMP[SYSCON3] or PSCCTL0_SENS[SYSCON3]
SYSCON4	CLKCTL4	PSCCTL0_MEDIA[SYSCON4]

29.6 Memory map and register definition

29.6.1 SYSCON0 register descriptions

29.6.1.1 SYSCON0 memory map

SYSCON0 base address: 4000_2000h

Offset	Register	Width (In bits)	Access	Reset value
14h	NMI Source Select (NMISRC)	32	RW	0000_0000h
1Ch	CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)	32	RW	0000_0000h
78h	Bus Matrix Priority (AHBMATPRIO)	32	RW	0000_0000h
80h	Receive Event Pulse Generator (RXEVPULSEGEN)	32	W	0000_0000h
84h	Latched Cortex-M33 Transmit Event (LATCHED_CM33_TXEV)	32	RW	0000_0000h
90h	System Secure Tick Calibration (SYSTEM_STICK_CALIB)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
94h	System Non-Secure Tick Calibration (SYSTEM_NSTICK_CALIB)	32	RW	0000_0000h
98h	CPU0 Status (CPU0_STATUS)	32	R	0000_0000h
D0h	GPIO Synchronization Stages (GPIO_PSYNC)	32	RW	0000_0000h
114h	Automatic Clock Gate Override (AUTOCLKGATEOVERRIDE0)	32	RW	0003_FFFFh
118h	SRAM Clock Gating Control (SRAM_CLKGATE_CTRL)	32	RW	0003_FFFFh
130h	MMU0 Memory Control (MMU0_MEM_CTRL)	32	RW	0000_0000h
140h	eDMA0 Memory Control (EDMA0_MEM_CTRL)	32	RW	0000_0000h
144h	eDMA1 Memory Control (EDMA1_MEM_CTRL)	32	RW	0000_0000h
14Ch	ETF Memory Control (ETF_MEM_CTRL)	32	RW	0000_0000h
150h	MMU1 Memory Control (MMU1_MEM_CTRL)	32	RW	0000_0000h
154h	XSPI0 Memory Control (XSPI0_MEM_CTRL)	32	RW	0000_0000h
158h	XSPI1 Memory Control (XSPI1_MEM_CTRL)	32	RW	0000_0000h
15Ch	CACHE64_CTRL0 Data Memory Control (XSPI0_DATA_MEM_CTRL)	32	RW	0000_0000h
160h	CACHE64_CTRL1 Data Memory Control (XSPI1_DATA_MEM_CTRL)	32	RW	0000_0000h
164h	NPU Memory Control (NPU_MEM_CTRL)	32	RW	0000_0000h
174h	CPU0 Memory Data Control (CM33_MEM_DATA_CTRL)	32	RW	0000_0000h
178h	CPU0 Memory Tag Control (CM33_MEM_TAG_CTRL)	32	RW	0000_0000h
208h	HiFi4 Memory Control (HIFI4_MEM_CTL)	32	RW	0000_0000h
240h	SAI0-2 MCLK IO Direction Control (SAI0_MCLK_CTRL)	32	RW	0000_0000h
25Ch	CACHE64_CTRL0 Tag Memory Control (XSPI0_TAG_MEM_CTRL)	32	RW	0000_0000h
260h	CACHE64_CTRL1 Tag Memory Control (XSPI1_TAG_MEM_CTRL)	32	RW	0000_0000h
280h	VDD2_COMP Auto Gating Enable (COMP_AUTOGATE_EN)	32	RW	0000_0000h
300h	HiFi4 Stall (DSPSTALL)	32	RW	0000_0001h
304h	HiFi4 OCDHaltOnReset (OCDHALTONRESET)	32	RW	0000_0000h
308h	HiFi4 General Purpose Register 0 (HIFI4_GPR0)	32	R	0000_0000h
314h	HiFi4 General Purpose Register 1 (HIFI4_GPR1)	32	R	0000_0000h
318h	HiFi4 General Purpose Register 2 (HIFI4_GPR2)	32	R	0000_0000h
31Ch	HiFi4 DSP Vector Remap (DSP_VECT_REMAP)	32	RW	0000_0400h
420h	eDMA0 Request Enable 0 (EDMA0_EN0)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
424h	eDMA0 Request Enable 1 (EDMA0_EN1)	32	RW	0000_0000h
428h	eDMA0 Request Enable 2 (EDMA0_EN2)	32	RW	0000_0000h
42Ch	eDMA0 Request Enable 3 (EDMA0_EN3)	32	RW	0000_0000h
430h	eDMA1 Request Enable 0 (EDMA1_EN0)	32	RW	0000_0000h
434h	eDMA1 Request Enable 1 (EDMA1_EN1)	32	RW	0000_0000h
438h	eDMA1 Request Enable 2 (EDMA1_EN2)	32	RW	0000_0000h
43Ch	eDMA1 Request Enable 3 (EDMA1_EN3)	32	RW	0000_0000h
600h	AXBS Control (AXBS_CTRL)	32	RW	0000_0000h
628h	I3C Asynchronous Wake-up Control (I3C_ASYNC_WAKEUP_CTRL)	32	RW	0000_0000h
650h	Gray to Binary Converter - Gray Code [31:0] (GRAY_CODE_LSB)	32	RW	0000_0000h
654h	Gray to Binary Converter - Gray Code [63:32] (GRAY_CODE_MSB)	32	RW	0000_0000h
658h	Gray to Binary Converter - Binary Code [31:0] (BINARY_CODE_LSB)	32	R	0000_0000h
65Ch	Gray to Binary Converter - Binary Code [63:32] (BINARY_CODE_MSB)	32	R	0000_0000h
B38h	ELS Boot State 0 (ELS_BOOT_STATE0)	32	R	0000_0000h
B3Ch	ELS Boot State 1 (ELS_BOOT_STATE1)	32	R	0000_0000h
B40h	ELS Boot State 2 (ELS_BOOT_STATE2)	32	R	0000_0000h

29.6.1.2 NMI Source Select (NMISRC)

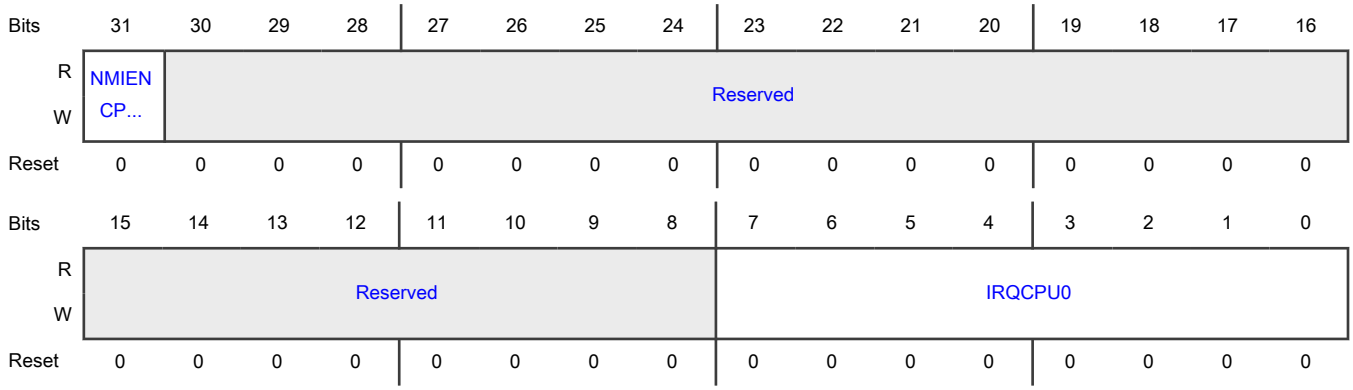
Offset

Register	Offset
NMISRC	14h

Function

Selects peripheral interrupts as source for the Non-Maskable Interrupt (NMI) of CPU0. The NMI source must first be disabled by writing 0 to the NMIENCPU0 bit to change the interrupt source for the NMI. The source can then be changed by updating the IRQCPU0 bits and re-enable the NMI source by setting NMIENCPU0.

Diagram



Fields

Field	Function
31 NMIENCPU0	Enables the Non-Maskable Interrupt (NMI) source selected by IRQCPU0 0b - Disables 1b - Enables
30-8 —	Reserved
7-0 IRQCPU0	The IRQ number of the interrupt that acts as the NMI for CPU0, if enabled by NMIENCPU0

29.6.1.3 CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)

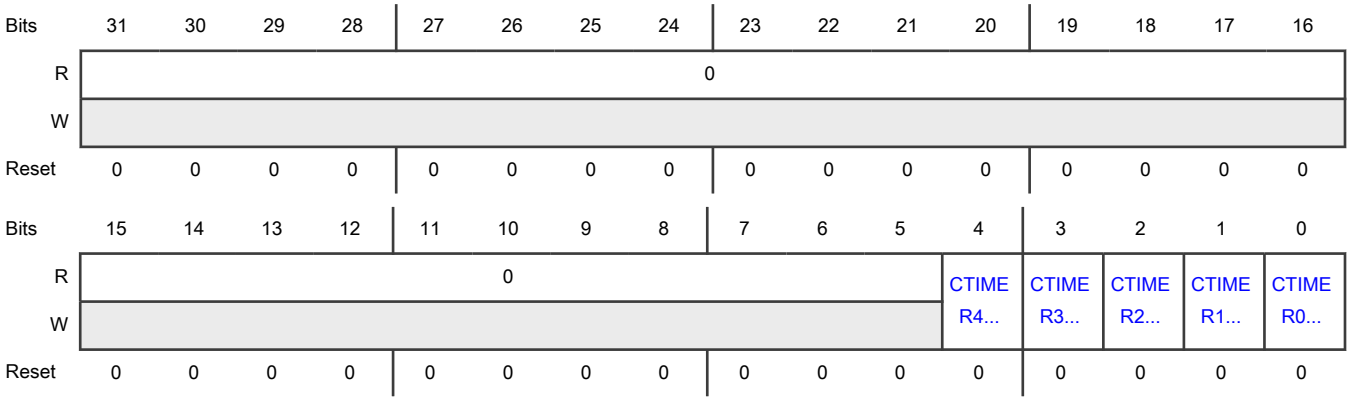
Offset

Register	Offset
CTIMERGLOBALSTARTEN	1Ch

Function

Configures CTIMER function clocks. PSCCTL2 register in CLKCTL0 also controls the clock for these CTIMERs. See CLKCTL chapter for details.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 CTIMER4_CLK_EN	CTIMER4 Function Clock Enable 0b - Disables 1b - Enables
3 CTIMER3_CLK_EN	CTIMER3 Function Clock Enable 0b - Disables 1b - Enables
2 CTIMER2_CLK_EN	CTIMER2 Function Clock Enable 0b - Disables 1b - Enables
1 CTIMER1_CLK_EN	CTIMER1 Function Clock Enable 0b - Disables 1b - Enables
0 CTIMER0_CLK_EN	CTIMER0 Function Clock Enable 0b - Disables 1b - Enables

29.6.1.4 Bus Matrix Priority (AHBMATPRIO)

Offset

Register	Offset
AHBMATPRIO	78h

Function

The multilayer AHB matrix arbitrates between several initiators, only if they attempt to access the same matrix target port at the same time. Care should be taken if the value in this register is changed, improper settings can seriously degrade performance. Priority values are 0 = highest, 3 = lowest. When the priority is the same, the initiator with the lower M number is given priority. See System bus diagram in Introduction chapter.

NOTE

Be careful when modifying this register as improper settings can seriously degrade the performance.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PBUS_M3		PBUS_M2		PBUS_M1		PBUS_M0		0						MBUS_M8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		MBUS_M6		MBUS_M5		MBUS_M4		MBUS_M3		MBUS_M2		MBUS_M1		MBUS_M0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 PBUS_M3	P-Bus M3 Priority Controls the priority for NIC_MEDIA1 initiators when going through the P-bus. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
29-28 PBUS_M2	P-Bus M2 Priority Controls the priority for the M-bus initiators when going through the P-bus. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
27-26 PBUS_M1	P-Bus M1 Priority Controls the priority for HiFi4 when going through the P-bus. 00b - Level 0 01b - Level 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Level 2 11b - Level 3
25-24 PBUS_M0	P-Bus M0 Priority Controls the priority for the CPU0 system bus accesses when going through the P-bus. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
23-18 —	Reserved
17-16 MBUS_M8	M-Bus M8 Priority Controls the priority for the ETR accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
15-14 —	Reserved
13-12 MBUS_M6	M-Bus M6 Priority Controls the priority for ELS accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
11-10 MBUS_M5	M-Bus M5 Priority Controls the priority for eDMA1 accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
9-8	M-Bus M4 Priority

Table continues on the next page...

Table continued from the previous page...

Field	Function
MBUS_M4	Controls the priority for eDMA0 accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
7-6 MBUS_M3	M-Bus M3 Priority Controls the priority for CPU0 code bus accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
5-4 MBUS_M2	M-Bus M2 Priority Controls the priority for XCACHE0 (CPU0 system bus cache) accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
3-2 MBUS_M1	M-Bus M1 Priority Controls the priority for XCACHE1 (CPU1 code bus cache) accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
1-0 MBUS_M0	M-Bus M0 Priority Controls the priority for the CPU0 system bus accesses when going through the M-bus. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3

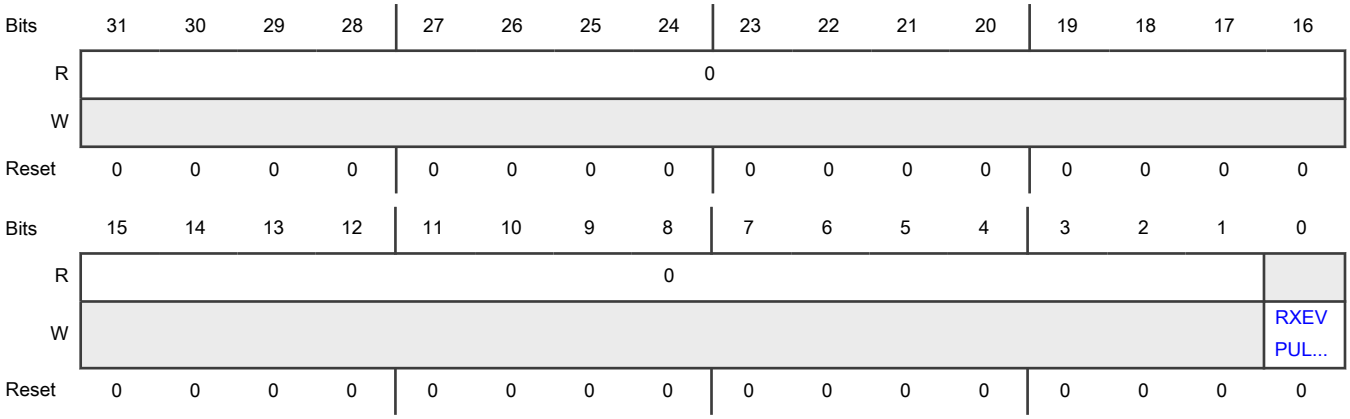
29.6.1.5 Receive Event Pulse Generator (RXEVPULSEGEN)

Offset

Register	Offset
RXEVPULSEGEN	80h

Function
Allows generating a pulse on CPU0 receive event input that will wake it up if it is stopped in a wait for event (WFE) instruction.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 RXEVPULSEGEN	Receive Event Pulse Generator Writing a 1 to this register will create a logic 1 pulse for one APB bus clock when set by HiFi4 or EZH-V. It is automatically cleared. 0b - No effect 1b - Pulse receive event high for one cycle

29.6.1.6 Latched Cortex-M33 Transmit Event (LATCHED_CM33_TXEV)

Offset

Register	Offset
LATCHED_CM33_TXEV	84h

Function
CPU0 TXEV (Cortex-M33 send event) is latched into the register and then cleared by EZH-V writing 1 to the flag.

CPU0 TXEV is a cycle pulse and one of the EZH-V interrupts, so the signal is latched in SYSCON0 and waiting to be cleared by EZH-V.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															LATCHED...
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 LATCHED_TXEV	Cortex-M33 Transmit Event Write a 1 to clear this register. 0b - No effect 1b - Clear the event flag

29.6.1.7 System Secure Tick Calibration (SYSTEM_STICK_CALIB)

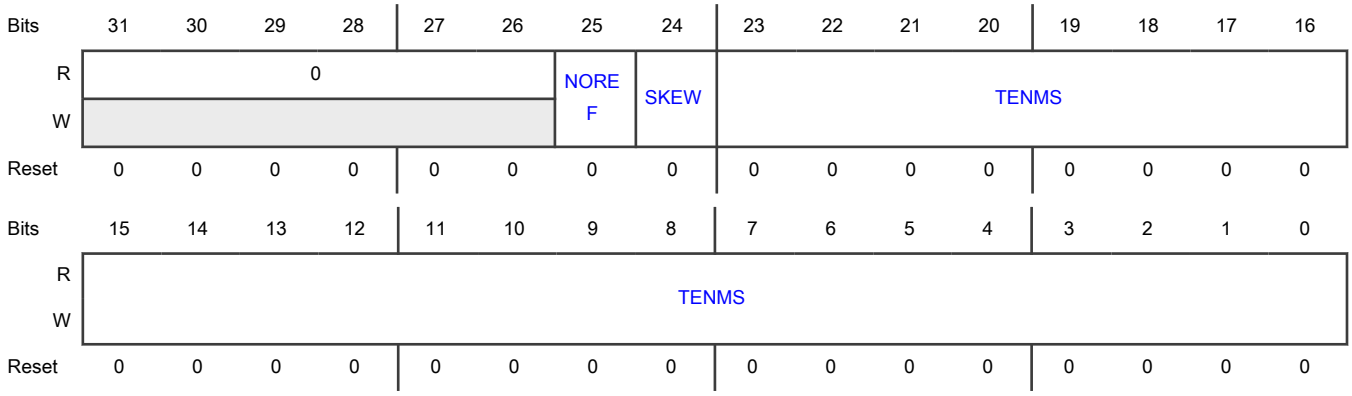
Offset

Register	Offset
SYSTEM_STICK_CALIB	90h

Function

Allows software to set up a default value for the SYST_CALIB register (see Arm documentation) in the System Tick Timer of secure part of CPU0.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 NOREF	No Reference Clock Controls whether the device provides a reference clock to the processor. 0b - Provides 1b - Not provide
24 SKEW	Skew Controls whether [TENMS] value is exact. 0b - Exact 1b - Inexact, or not given
23-0 TENMS	Ten Milliseconds Reload value for 10 ms (100 Hz) timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

29.6.1.8 System Non-Secure Tick Calibration (SYSTEM_NSTICK_CALIB)

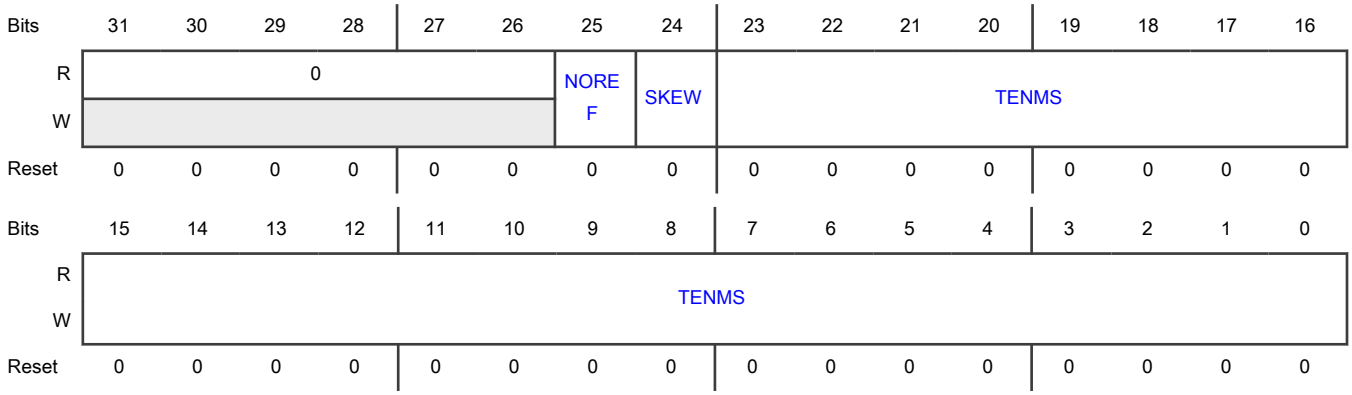
Offset

Register	Offset
SYSTEM_NSTICK_CALIB	94h

Function

Allows software to set up a default value for the SYST_CALIB register in the System Tick Timer of non-secure part of CPU0.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 NOREF	No Reference Clock Controls whether the device provides a reference clock to the processor: 0b - Provides 1b - Not provide
24 SKEW	Skew Controls whether [TENMS] value is exact. 0b - Exact 1b - Inexact, or not given
23-0 TENMS	Ten Milliseconds Reload value for 10 ms (100 Hz) timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

29.6.1.9 CPU0 Status (CPU0_STATUS)

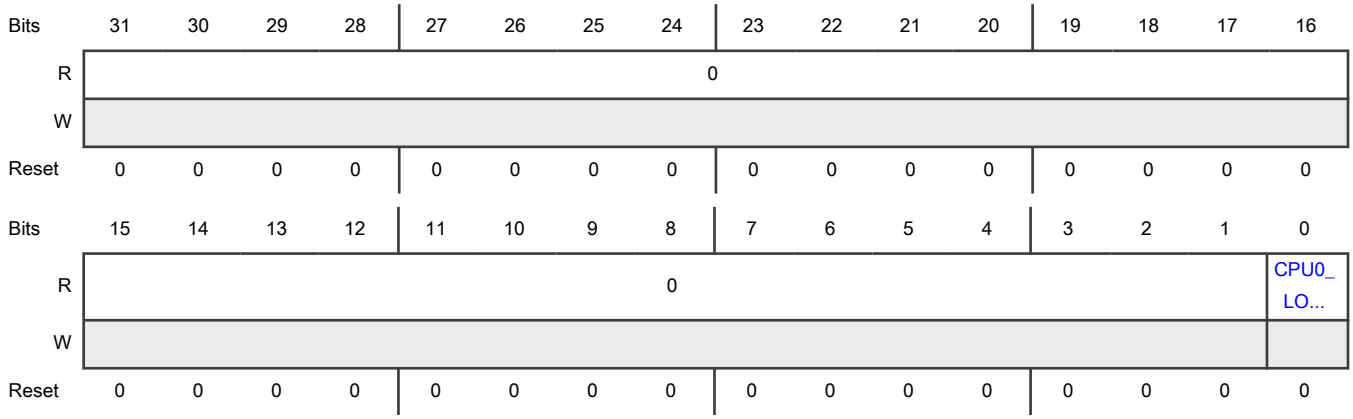
Offset

Register	Offset
CPU0_STATUS	98h

Function

Stores CPU0 status.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CPU0_LOCKUP	CPU0 Lockup State Indicates whether CPU0 is in lockup state. 0b - Not in lockup 1b - In lockup

29.6.1.10 GPIO Synchronization Stages (GPIO_PSYNC)

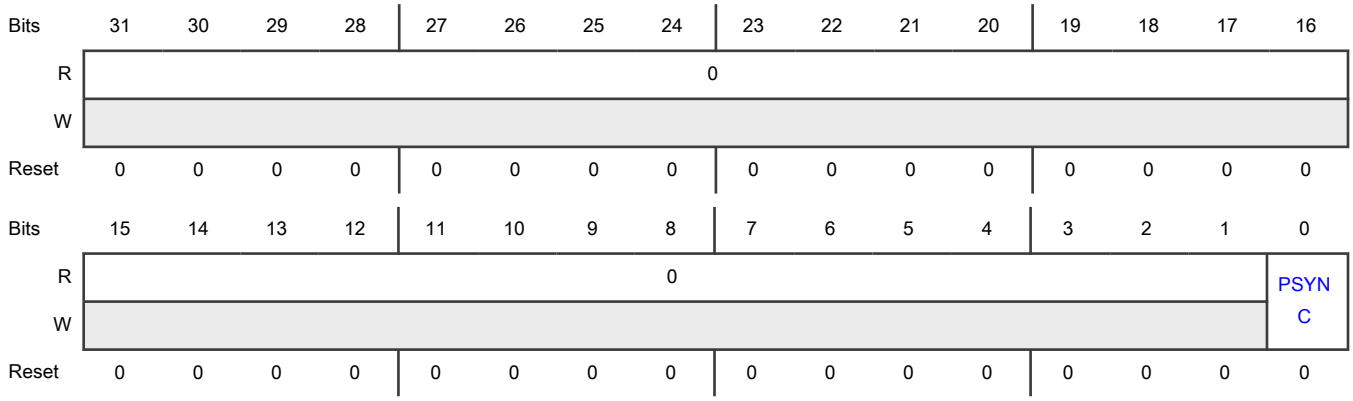
Offset

Register	Offset
GPIO_PSYNC	D0h

Function

Selects the amount of synchronization stages in the GPIO inputs.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 PSYNC	Synchronization Stage Setting The PINT block has two flip-flop synchronizers on the interrupt and pattern match detect circuits based on data inputs from the pads. The first flip-flop stage of synchronization is clocked on the falling edge of the peripheral clock. The second flip-flop stage is clocked on the rising edge of the peripheral clock. The synchronizers can be configured to be 1-stage synchronizers to reduce the signal latency at low peripheral clock frequencies. 0b - 2-stage sync 1b - 1-stage sync

29.6.1.11 Automatic Clock Gate Override (AUTOCLKGATEOVERRIDE0)

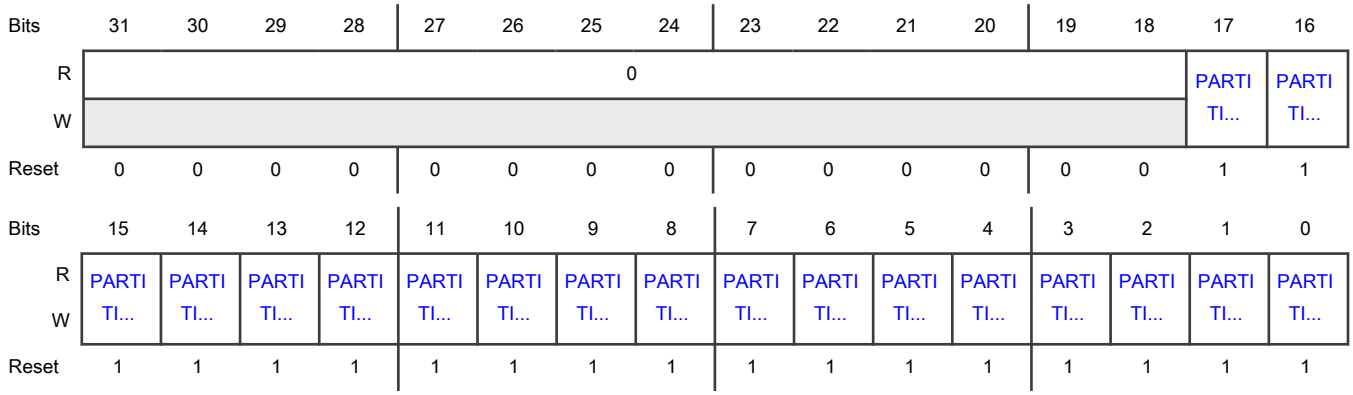
Offset

Register	Offset
AUTOCLKGATEOVERRIDE0	114h

Function

This register allows selectively disabling automatic clock gating for SRAM partitions. By default, automatic clock gating turns off clocks to each internal SRAM after 5 bus clocks with no activity. This saves power when the SRAMs are not used for a period of time. When turned off due to automatic clock gating, there is a 1-clock delay for the next access to an SRAM.

Diagram



Fields

Field	Function
31-18 —	Reserved
17-0 PARTITIONn	Partition n Controls clock gating for Partition 0 (bit 0) to Partition 17 (bit 17). 0b - Enables clock gating 1b - Continuous clocking

29.6.1.12 SRAM Clock Gating Control (SRAM_CLKGATE_CTRL)

Offset

Register	Offset
SRAM_CLKGATE_CTRL	118h

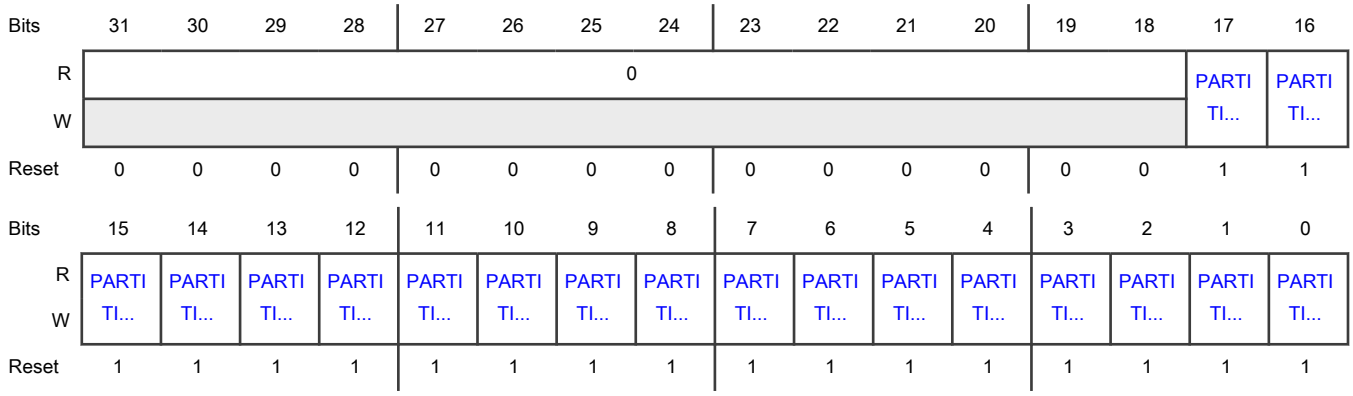
Function

If cleared, the corresponding SRAM partition input clock driven by COMPUTE_MAIN_CLK_1 will be gated off. This provides means to save power if the corresponding partition is not planned to be used for a significant amount of time. To get the power saving, the Compute CPU (CPU0) needs to do a dummy read to the partition to switch the partition clock to COMPUTE_MAIN_CLK_1, and then clear this bit. While the bit for an associated partition is cleared, no bus initiators will be able to access that partition unless the bit is set again.

NOTE

Any core will halt when accessing the gated partition, until CPU0 turns on the partition clock.

Diagram



Fields

Field	Function
31-18 —	Reserved
17-0 PARTITIONn	Partition n Gates off the bus clock input in VDD2_COMP domain for Partition 0 (bit 0) to Partition 17 (bit 17). 0b - Bus clock input gated 1b - No effects

29.6.1.13 MMU0 Memory Control (MMU0_MEM_CTRL)

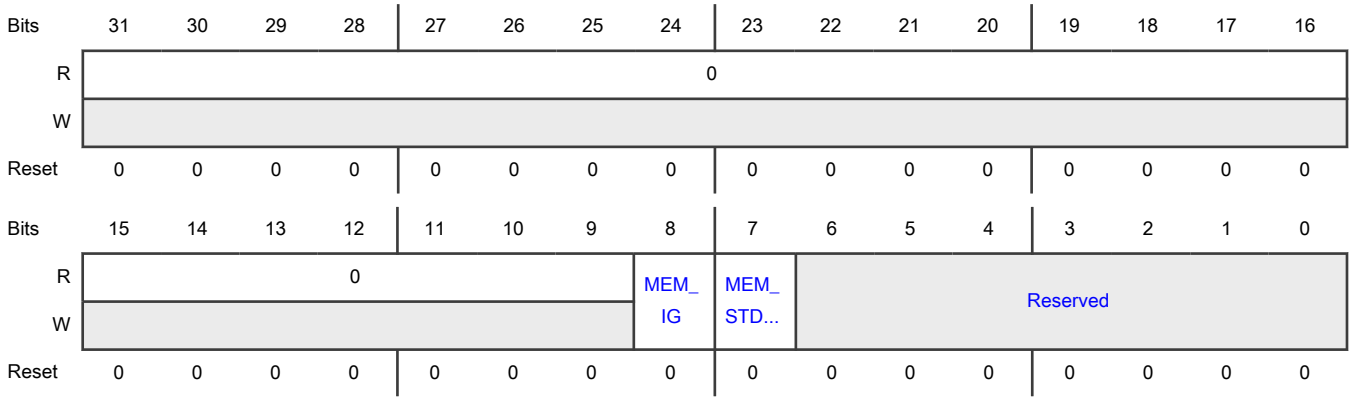
Offset

Register	Offset
MMU0_MEM_CTRL	130h

Function

Controls Memory Map Unit (MMU0) memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.14 eDMA0 Memory Control (EDMA0_MEM_CTRL)

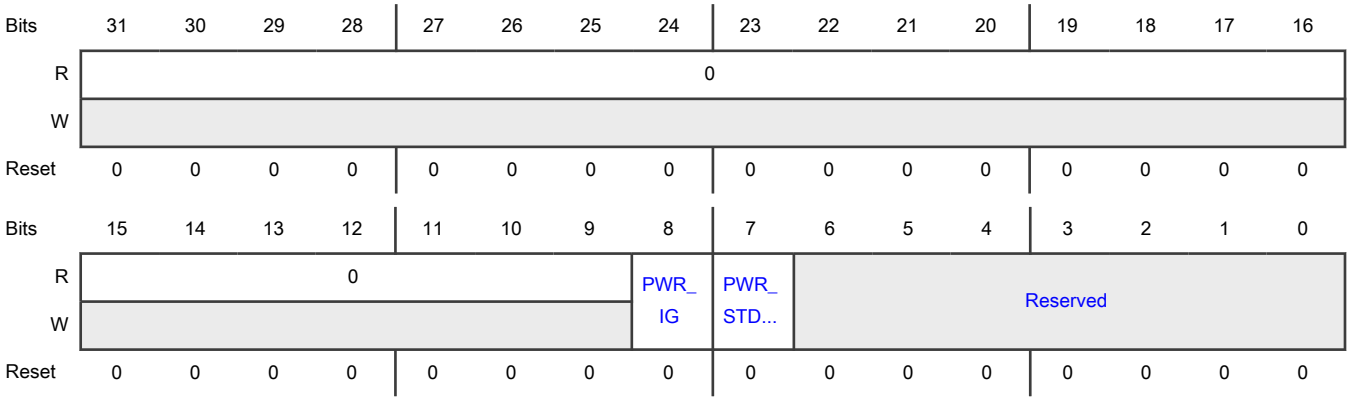
Offset

Register	Offset
EDMA0_MEM_CTRL	140h

Function

Controls Enhanced Direct Memory Access (eDMA) 0 memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.15 eDMA1 Memory Control (EDMA1_MEM_CTRL)

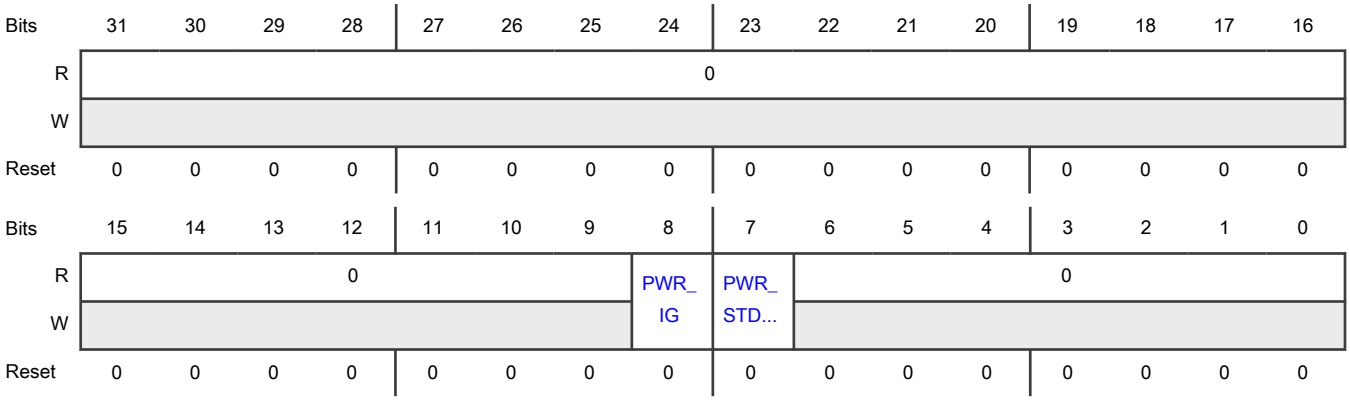
Offset

Register	Offset
EDMA1_MEM_CTRL	144h

Function

Controls Enhanced Direct Memory Access (eDMA) 1 memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
6-0 —	Reserved

29.6.1.16 ETF Memory Control (ETF_MEM_CTRL)

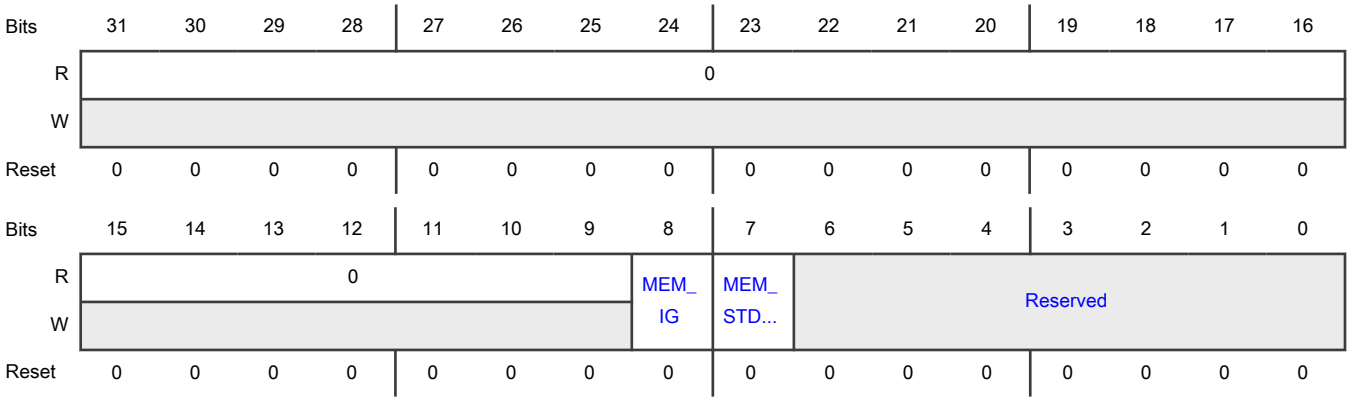
Offset

Register	Offset
ETF_MEM_CTRL	14Ch

Function

Configures input and power gating for ETF memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7	Memory Standby Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_STDBY	When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.17 MMU1 Memory Control (MMU1_MEM_CTRL)

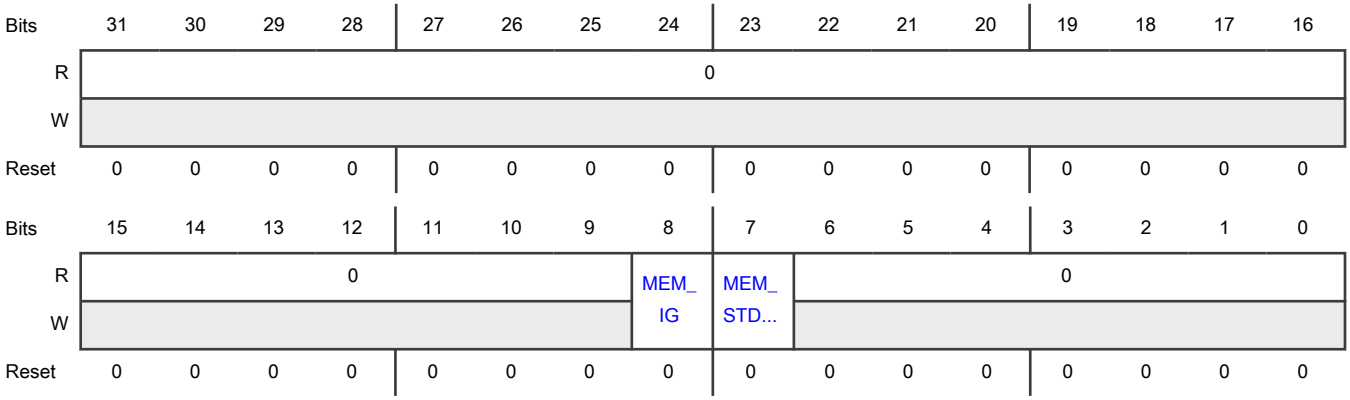
Offset

Register	Offset
MMU1_MEM_CTRL	150h

Function

Controls Memory Map Unit (MMU1) memory.

Diagram



Fields

Field	Function
31-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.18 XSPI0 Memory Control (XSPI0_MEM_CTRL)

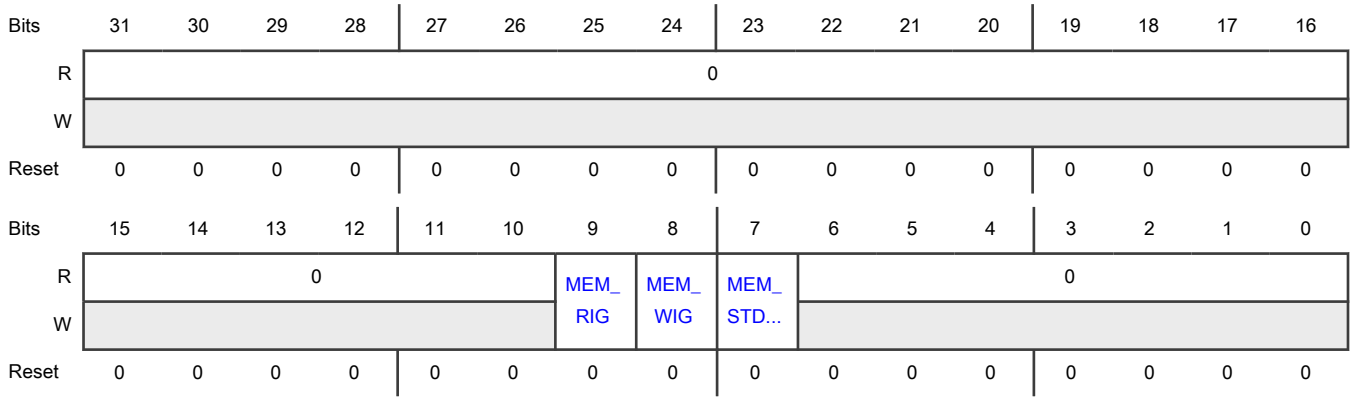
Offset

Register	Offset
XSPI0_MEM_CTRL	154h

Function

Controls XSPI0 memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

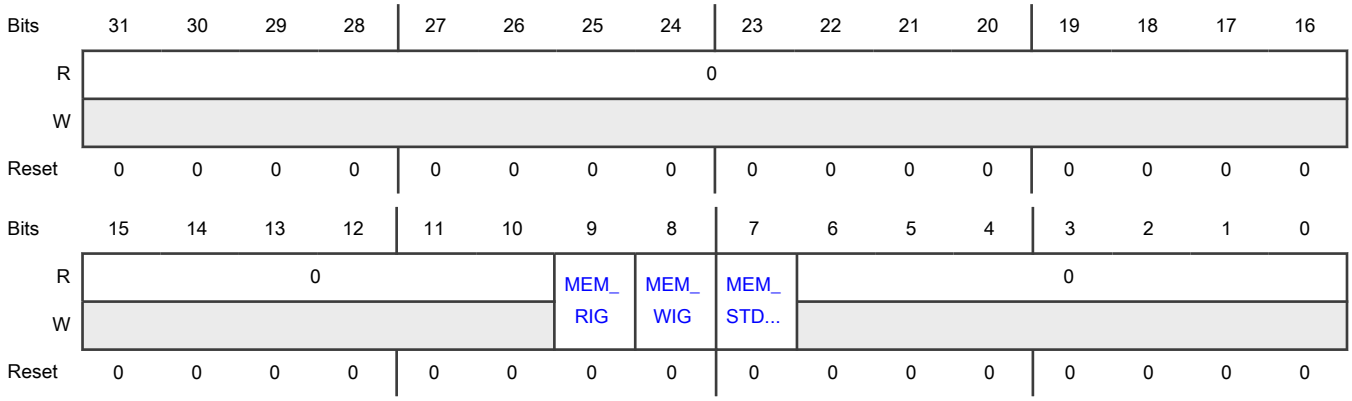
29.6.1.19 XSPI1 Memory Control (XSPI1_MEM_CTRL)

Offset

Register	Offset
XSPI1_MEM_CTRL	158h

Function
Controls XSPI1 memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.20 CACHE64_CTRL0 Data Memory Control (XSPI0_DATA_MEM_CTRL)

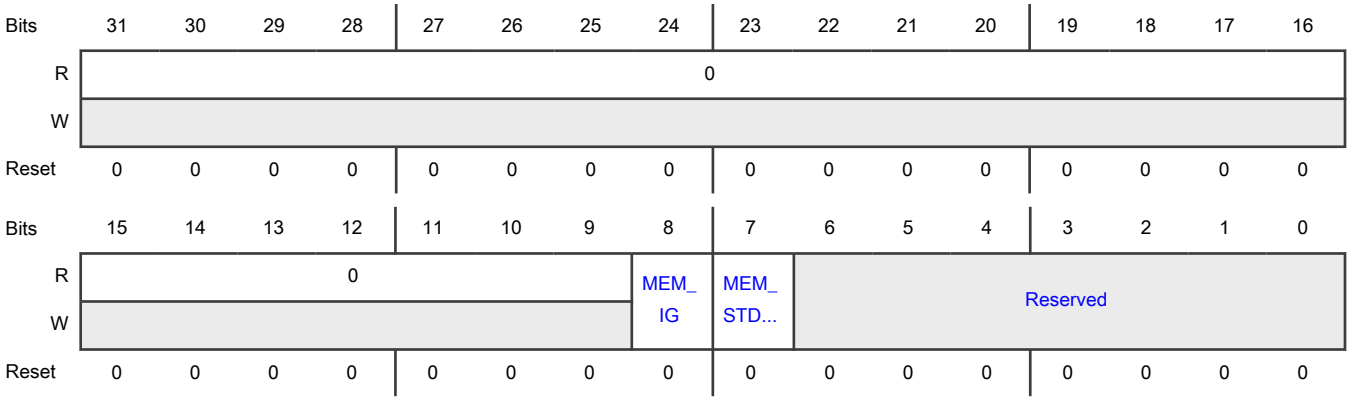
Offset

Register	Offset
XSPI0_DATA_MEM_CTRL	15Ch

Function

Controls CACHE64_CTRL0 data memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Active
6-0 —	Reserved

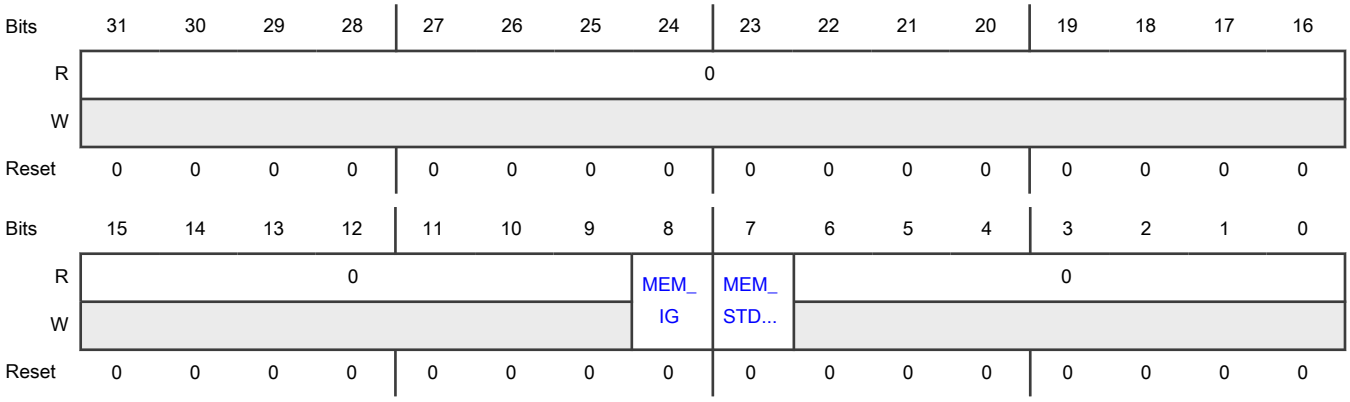
29.6.1.21 CACHE64_CTRL1 Data Memory Control (XSPI1_DATA_MEM_CTRL)

Offset

Register	Offset
XSPI1_DATA_MEM_CTRL	160h

Function
Controls CACHE64_CTRL1 data memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.22 NPU Memory Control (NPU_MEM_CTRL)

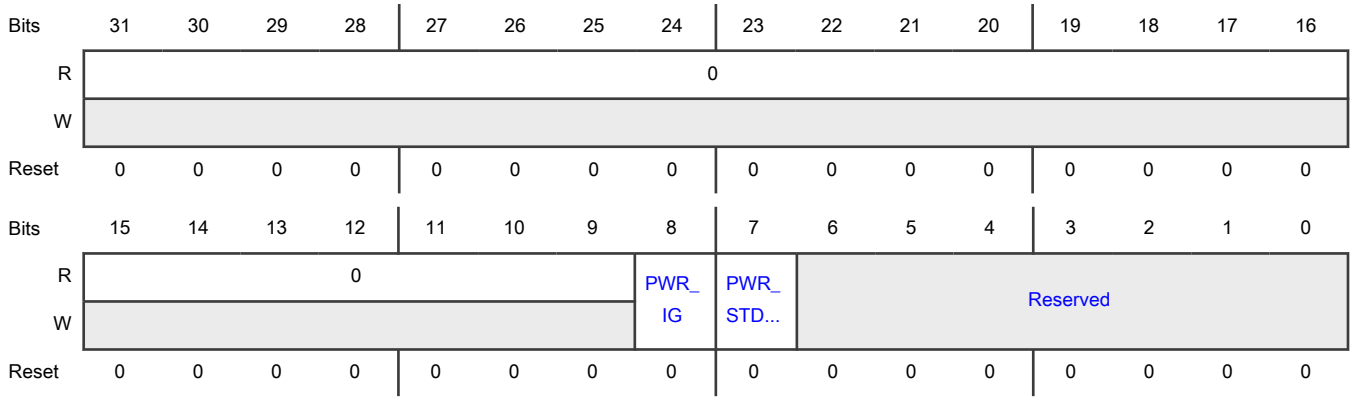
Offset

Register	Offset
NPU_MEM_CTRL	164h

Function

Controls NPU memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.23 CPU0 Memory Data Control (CM33_MEM_DATA_CTRL)

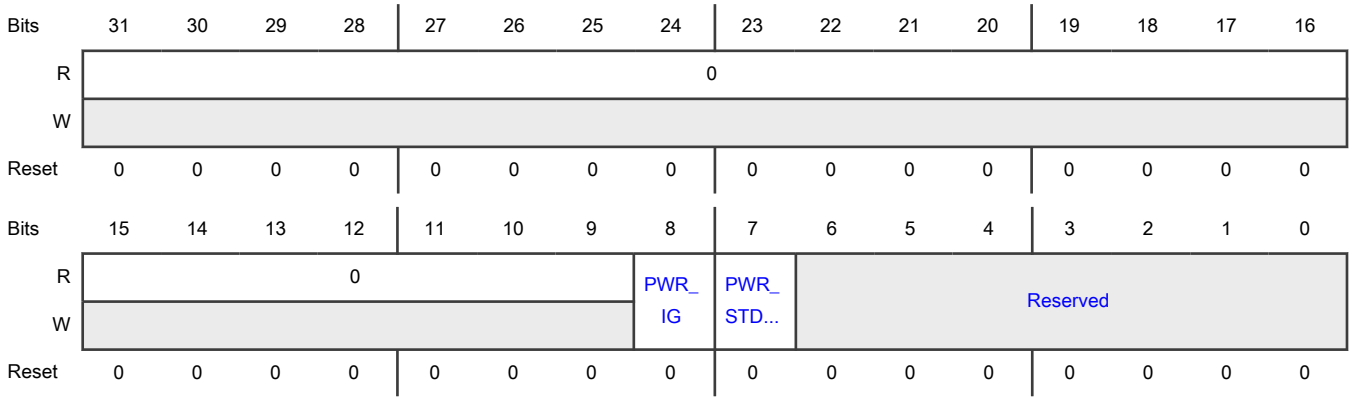
Offset

Register	Offset
CM33_MEM_DATA_CTRL	174h

Function

Configures input and power gating for CPU0 XCACHE data memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.24 CPU0 Memory Tag Control (CM33_MEM_TAG_CTRL)

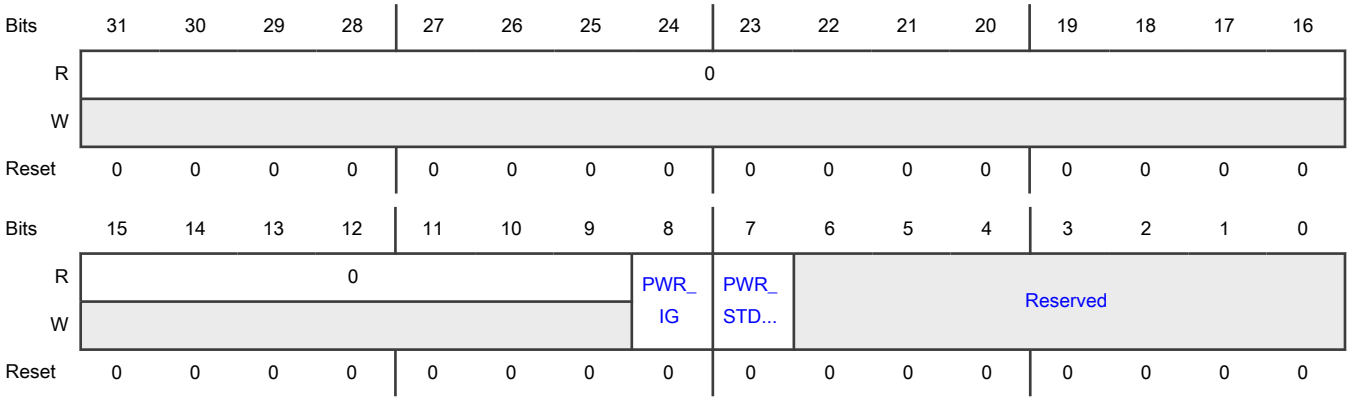
Offset

Register	Offset
CM33_MEM_TAG_CTRL	178h

Function

Configures input and power gating for CPU0 XCACHE tag memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.25 HiFi4 Memory Control (HIFI4_MEM_CTL)

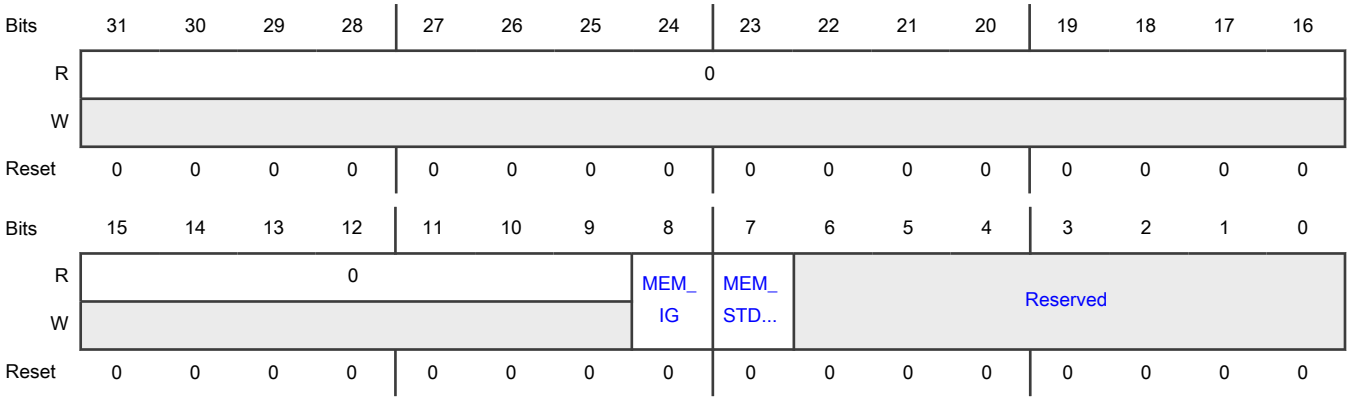
Offset

Register	Offset
HIFI4_MEM_CTL	208h

Function

Configures input and power gating for HiFi4 memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
6-0 —	Reserved

29.6.1.26 SAI0-2 MCLK IO Direction Control (SAI0_MCLK_CTRL)

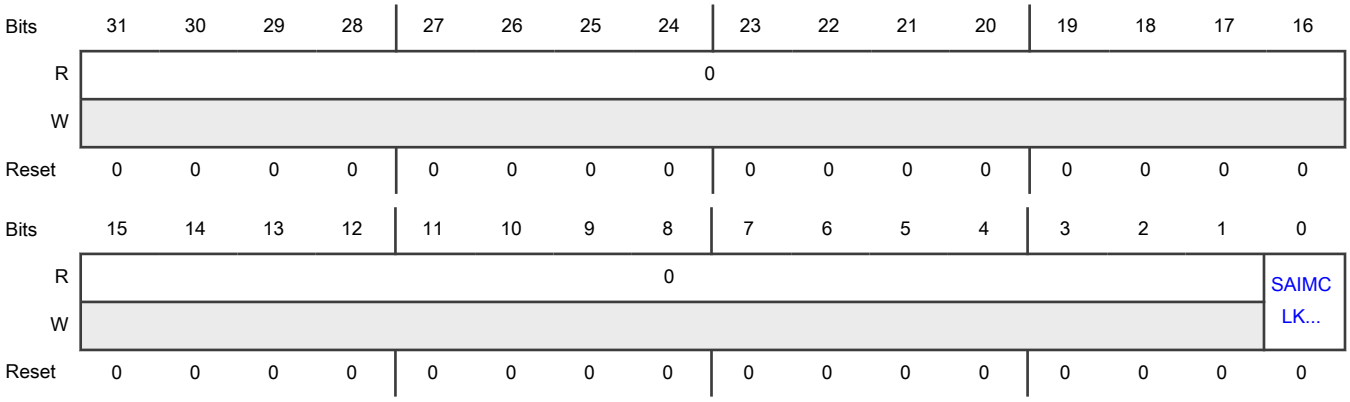
Offset

Register	Offset
SAI0_MCLK_CTRL	240h

Function

Controls MCLK input and output direction for Serial Audio Interface 0, 1, and 2 (SAI0-2).

Diagram



Fields

Field	Function
31-1 —	Reserved
0 SAIMCLKDIR	SAI0-2 MCLK Direction Control Specifies whether MCLK is output or input signal. IOPCTL needs to select MCLK function. 0b - Input 1b - Output

29.6.1.27 CACHE64_CTRL0 Tag Memory Control (XSPI0_TAG_MEM_CTRL)

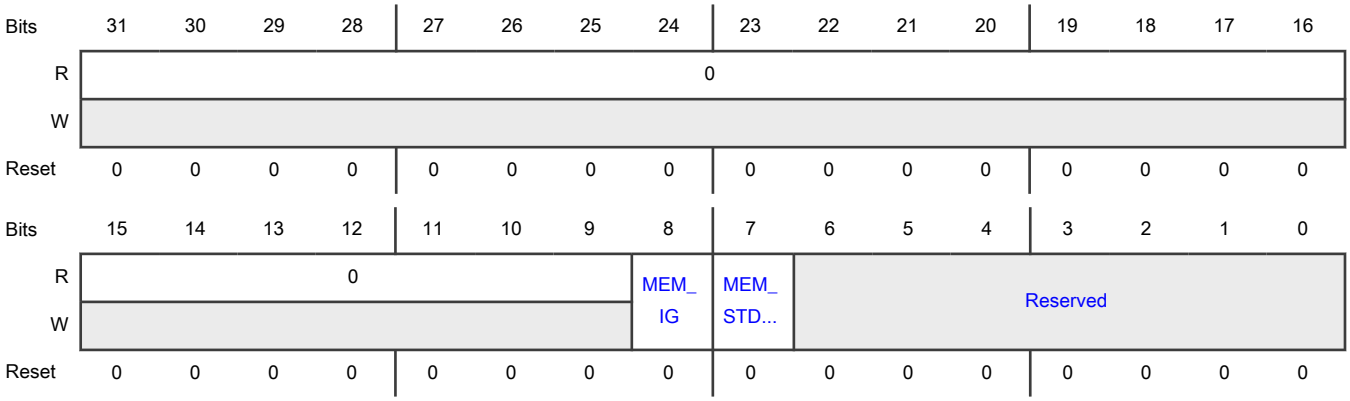
Offset

Register	Offset
XSPI0_TAG_MEM_CTRL	25Ch

Function

Controls CACHE64_CTRL0 tag memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.28 CACHE64_CTRL1 Tag Memory Control (XSPI1_TAG_MEM_CTRL)

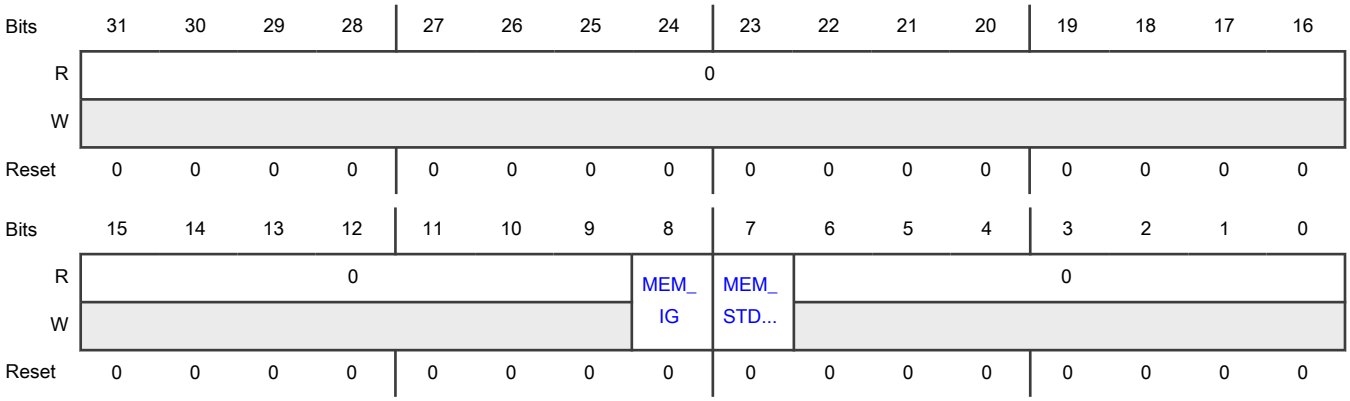
Offset

Register	Offset
XSPI1_TAG_MEM_CTRL	260h

Function

Configures input and power gating for CACHE64_CTRL1 tag memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.1.29 VDD2_COMP Auto Gating Enable (COMP_AUTOGATE_EN)

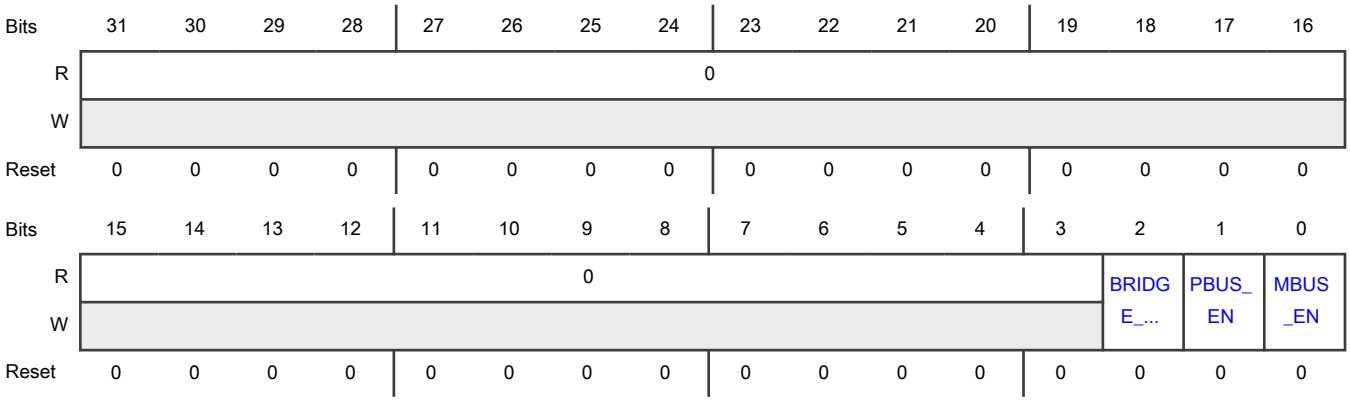
Offset

Register	Offset
COMP_AUTOGATE_EN	280h

Function

Controls clock gating for M-bus, P-bus, AHB2APB bridge, AIPS and AXBS in VDD2_COMP domain. This register allows selectively enabling automatic clock gating for the modules. By default, automatic clock gating turns off clocks to the modules after 2 bus clocks with no bus transaction activity. This saves power when the corresponding modules are not used for a period of time.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 BRIDGE_EN	Automatic Clock Gating Enable for AHB2APB bridge, AIPS and AXBS Controls clock gating for AHB2APB bridge, AIPS and AXBS in VDD2_COMP domain. You can set it to disable the module's clock in order to save power when the module is not working. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating
1 PBUS_EN	Automatic Clock Gating Enable for P-bus Controls clock gating for P-bus in VDD2_COMP domain. You can set it to disable the module's clock in order to save power when the module is not working. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating
0 MBUS_EN	Automatic Clock Gating Enable for M-bus Controls clock gating for M-bus in VDD2_COMP domain. You can set it to disable the module's clock in order to save power when the module is not working. **NOTE** Once CPU0 enters Deep Sleep when accessing RAM arbiter 0, if any other initiator needs to access RAM arbiter 0, you must disable this bit before CPU0 enters Deep Sleep. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating

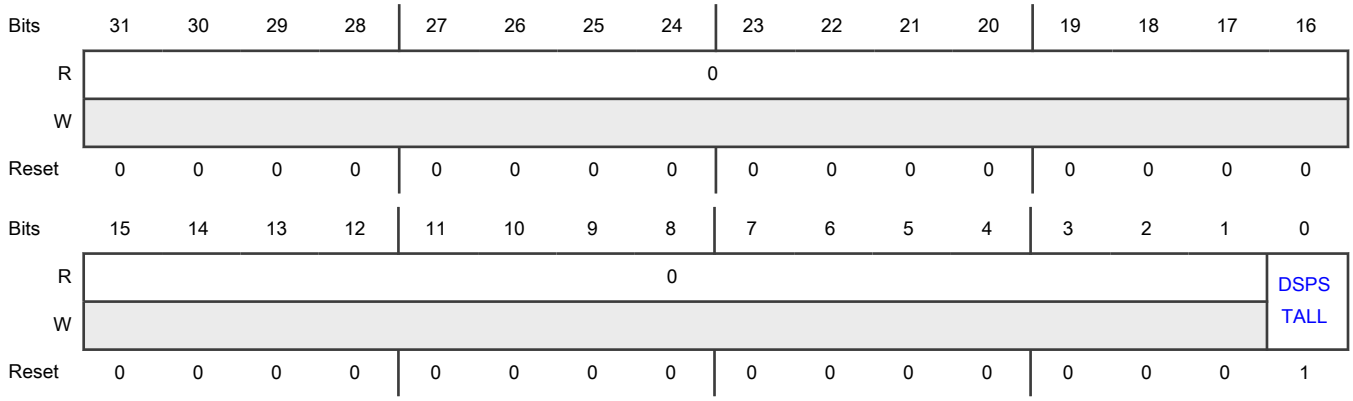
29.6.1.30 HiFi4 Stall (DSPSTALL)

Offset

Register	Offset
DSPSTALL	300h

Function
Controls the operation of HiFi4.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 DSPSTALL	Run and Stall Control 0b - Run (normal mode) 1b - Stall mode

29.6.1.31 HiFi4 OCDHaltOnReset (OCDHALTONRESET)

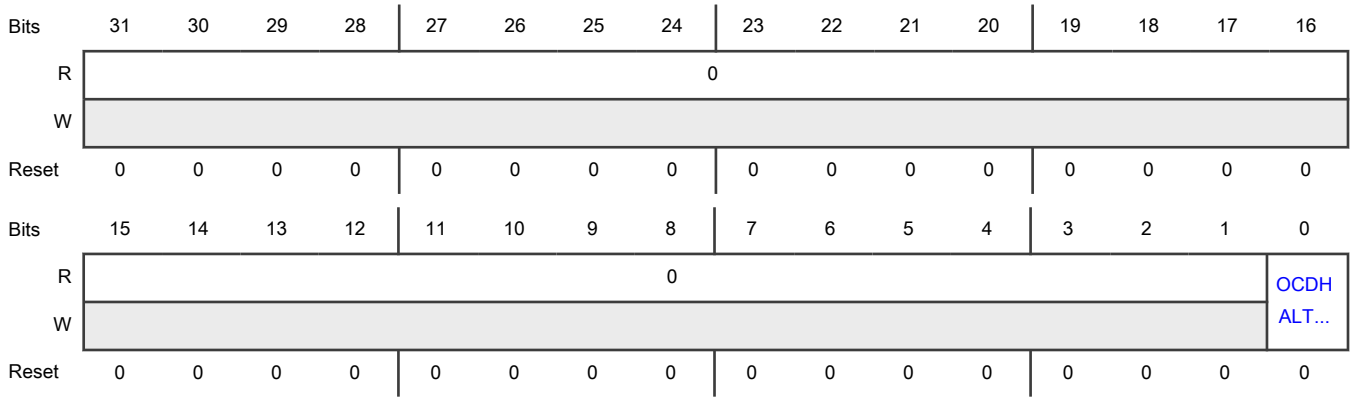
Offset

Register	Offset
OCDHALTONRESET	304h

Function

HiFi4 enters On-Chip Debug (OCD) Halt mode if this signal is sampled asserted on reset.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 OCDHALTONRESET	OCDHaltOnReset 0b - Allow normal operation when HiFi4 exits reset (starts fetching and executing instructions from memory) 1b - Force HiFi4 to enter OCD Halt mode when HiFi4 exits reset

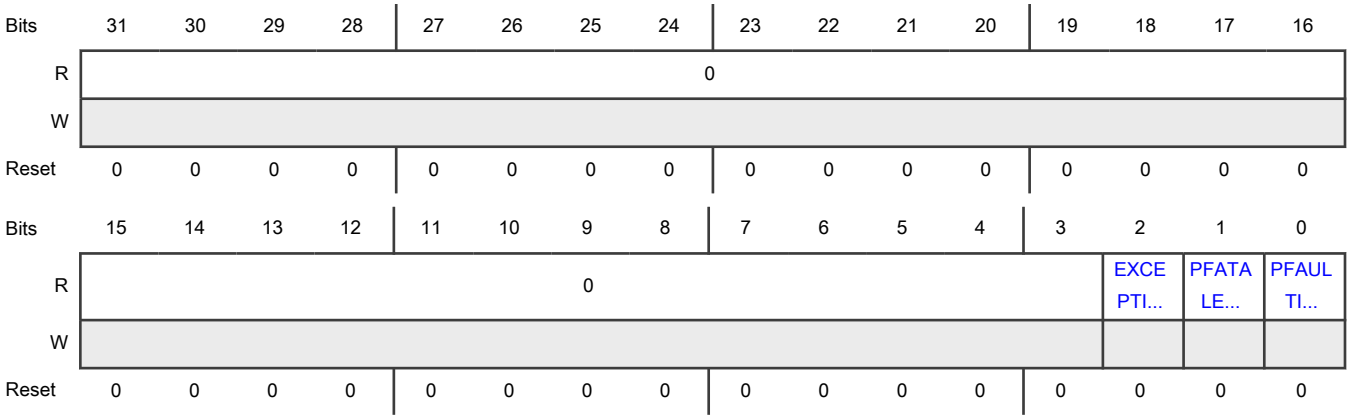
29.6.1.32 HiFi4 General Purpose Register 0 (HIFI4_GPR0)

Offset

Register	Offset
HIFI4_GPR0	308h

Function
General purpose register assigned to capture HiFi4 status signals.

Diagram



Fields

Field	Function
31-3 —	Reserved
2	DoubleExceptionError

Table continues on the next page...

Table continued from the previous page...

Field	Function
EXCEPTIONERR	Single-cycle assertion every time double exception faults occur. Unlike [PFATALERR], this signal is not sticky. The system can use the information presented on [PFATALERR] and HIFI4_GPR1[PFAULTINFO] and decide whether to reset the HIFI4 core or the whole system. 0b - No error occurs 1b - Error occurs
1 PFATALERR	PFatalError Asserted when a fatal error condition occurs due to AXI parity or ECC errors. The PFatalError signal is sticky and is cleared by HiFi4 reset. 0b - No error occurs 1b - Error occurs
0 PFAULTINFOVLD	PFaultInfoValid Asserted when the PFaultInfo signal changes its value. The system can use its rising edge as an enabler to capture PFaultInfo. 0b - Not asserted 1b - Asserted

29.6.1.33 HiFi4 General Purpose Register 1 (HIFI4_GPR1)

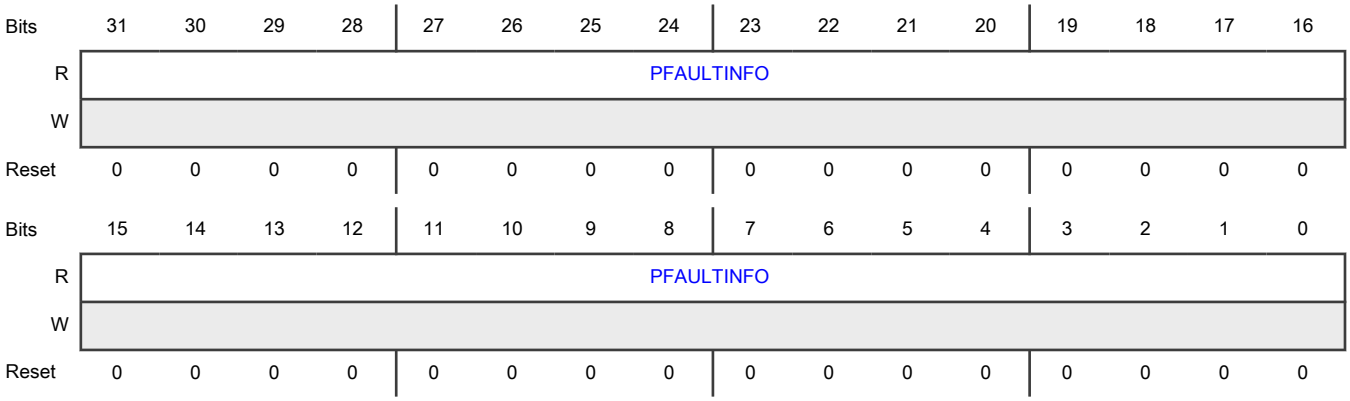
Offset

Register	Offset
HIFI4_GPR1	314h

Function

General purpose register assigned to capture HiFi4 status signals.

Diagram



Fields

Field	Function
31-0 PFAULTINFO	PFaultInfo This signal mirrors HiFi4's internal Fault Information Register and provides the source and severity of the fault. For detailed information, see "PFaultInfo and Fault Information Register (FIR)" section in HiFi4 chapter.

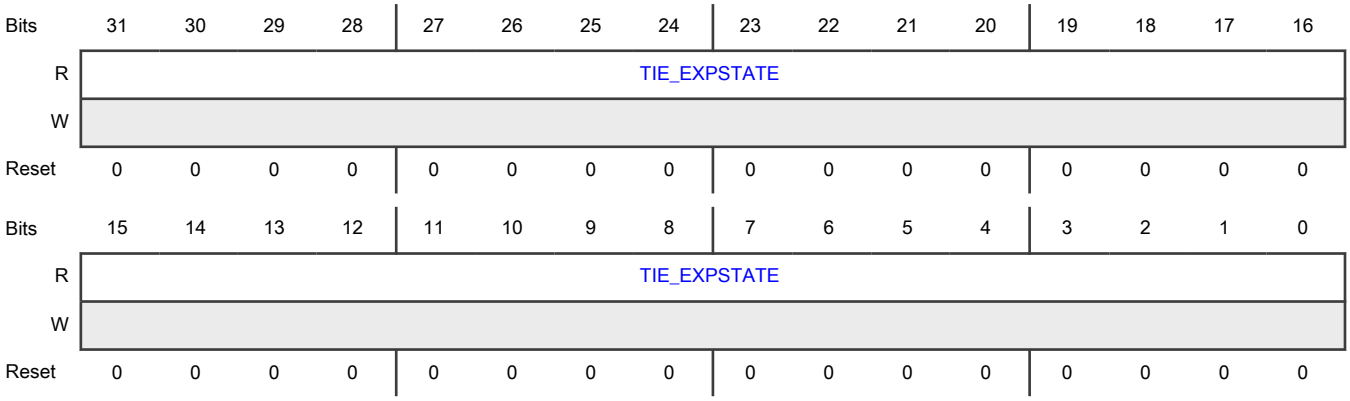
29.6.1.34 HiFi4 General Purpose Register 2 (HIFI4_GPR2)

Offset

Register	Offset
HIFI4_GPR2	318h

Function
General purpose register assigned to capture HiFi4 status signals.

Diagram



Fields

Field	Function
31-0 TIE_EXPSTATE	GPIO32 Option TIE Output State Indicates the state of “tie_expstate” used to capture OSTIMER and to trigger EZH-V.

29.6.1.35 HiFi4 DSP Vector Remap (DSP_VECT_REMAP)

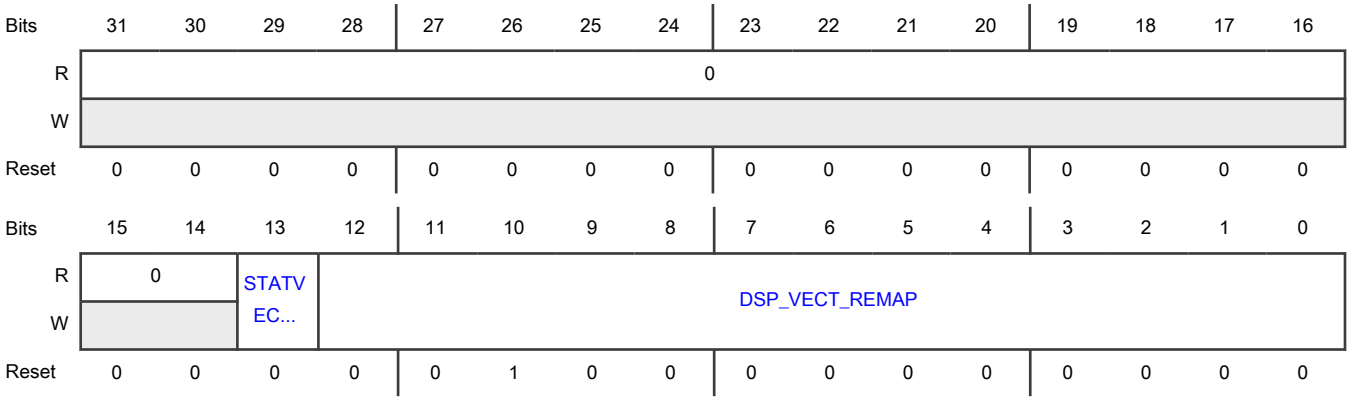
Offset

Register	Offset
DSP_VECT_REMAP	31Ch

Function

Configures HiFi4 remap function.

Diagram



Fields

Field	Function
31-14 —	Reserved
13 STATVECSELECT	HiFi4 Remap Function Enable Enables remap function on HiFi4. When enabled, remaps 2010_0000h to another address in shared RAM. HiFi4 vector base address is always at 2402_0000h although remap function is enabled. This address is HiFi4 local TCM. 0b - Remap function disabled 1b - Remap function enabled
12-0 DSP_VECT_REMAP	DSP Vector Remap Remaps ARADDR[22:10] when [STATVECSELECT] is set and ARADDR hits 1 KB from 2010_0000h. **NOTE** Remapping the vectors will leave a "hole" of 1 KB at the default vector location which HiFi4 DSP will never be able to access (CPU0 and all other initiators will have access though).

29.6.1.36 eDMA0 Request Enable 0 (EDMA0_EN0)

Offset

Register	Offset
EDMA0_EN0	420h

Function
Enables Request 0 for eDMA0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SDADC_F...	SDADC_F...	ADC0_FI...	ADC0_FI...	SCT0_DM...	SCT0_DM...	CTIMER4...	CTIMER4...	CTIMER3...	CTIMER3...	CTIMER2...	CTIMER2...	CTIMER1...	CTIMER1...	CTIMER0...	CTIMER0...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	XSPI2_T...	XSPI2_R...	XSPI1_T...	XSPI1_R...	XSPI0_T...	XSPI0_R...	MICFIL_...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 SDADC_FIFO1_EN	SDADC FIFO1 Enable 0b - Disables 1b - Enables
30 SDADC_FIFO0_EN	SDADC FIFO0 Enable 0b - Disables 1b - Enables
29 ADC0_FIFO1_EN	ADC0 FIFO1 Enable 0b - Disables 1b - Enables
28 ADC0_FIFO0_EN	ADC0 FIFO0 Enable 0b - Disables 1b - Enables
27	SCT0 DMA B Enable 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
SCT0_DMAB_EN	1b - Enables
26 SCT0_DMAA_EN	SCT0 DMA A Enable 0b - Disables 1b - Enables
25 CTIMER4_M1_EN	CTIMER4 M1 Enable 0b - Disables 1b - Enables
24 CTIMER4_M0_EN	CTIMER4 M0 Enable 0b - Disables 1b - Enables
23 CTIMER3_M1_EN	CTIMER3 M1 Enable 0b - Disables 1b - Enables
22 CTIMER3_M0_EN	CTIMER3 M0 Enable 0b - Disables 1b - Enables
21 CTIMER2_M1_EN	CTIMER2 M1 Enable 0b - Disables 1b - Enables
20 CTIMER2_M0_EN	CTIMER2 M0 Enable 0b - Disables 1b - Enables
19 CTIMER1_M1_EN	CTIMER1 M1 Enable 0b - Disables 1b - Enables
18 CTIMER1_M0_EN	CTIMER1 M0 Enable 0b - Disables 1b - Enables
17 CTIMER0_M1_EN	CTIMER0 M1 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 CTIMER0_M0_EN	CTIMER0 M0 Enable 0b - Disables 1b - Enables
15 PINT0_IRQ7_EN	PINT0 Interrupt 7 Enable 0b - Disables 1b - Enables
14 PINT0_IRQ6_EN	PINT0 Interrupt 6 Enable 0b - Disables 1b - Enables
13 PINT0_IRQ5_EN	PINT0 Interrupt 5 Enable 0b - Disables 1b - Enables
12 PINT0_IRQ4_EN	PINT0 Interrupt 4 Enable 0b - Disables 1b - Enables
11 PINT0_IRQ3_EN	PINT0 Interrupt 3 Enable 0b - Disables 1b - Enables
10 PINT0_IRQ2_EN	PINT0 Interrupt 2 Enable 0b - Disables 1b - Enables
9 PINT0_IRQ1_EN	PINT0 Interrupt 1 Enable 0b - Disables 1b - Enables
8 PINT0_IRQ0_EN	PINT0 Interrupt 0 Enable 0b - Disables 1b - Enables
7 XSPI2_TX_EN	XSPI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
6 XSPI2_RX_EN	XSPI2 RX FIFO Drain Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
5 XSPI1_TX_EN	XSPI1 TX FIFO Fill Enable 0b - Disables 1b - Enables
4 XSPI1_RX_EN	XSPI1 RX FIFO Drain Enable 0b - Disables 1b - Enables
3 XSPI0_TX_EN	XSPI0 TX FIFO Fill Enable 0b - Disables 1b - Enables
2 XSPI0_RX_EN	XSPI0 RX FIFO Drain Enable 0b - Disables 1b - Enables
1 MICFIL_EN	MICFIL Enable 0b - Disables 1b - Enables
0 —	Reserved

29.6.1.37 eDMA0 Request Enable 1 (EDMA0_EN1)

Offset

Register	Offset
EDMA0_EN1	424h

Function
Enables Request 1 for eDMA0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LP_FLEX...	LP_FLEX...	LP_FLEX...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	Reserved	ACMP_EN	SDADC_F...	SDADC_F...	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LP_FLEXCOMM9_RX_EN	LP_FLEXCOMM9 RX FIFO Drain Enable 0b - Disables 1b - Enables
30 LP_FLEXCOMM8_TX_EN	LP_FLEXCOMM8 TX FIFO Fill Enable 0b - Disables 1b - Enables
29 LP_FLEXCOMM8_RX_EN	LP_FLEXCOMM8 RX FIFO Drain Enable 0b - Disables 1b - Enables
28 LP_FLEXCOMM7_TX_EN	LP_FLEXCOMM7 TX FIFO Fill Enable 0b - Disables 1b - Enables
27 LP_FLEXCOMM7_RX_EN	LP_FLEXCOMM7 RX FIFO Drain Enable 0b - Disables 1b - Enables
26 LP_FLEXCOMM6_TX_EN	LP_FLEXCOMM6 TX FIFO Fill Enable 0b - Disables 1b - Enables
25 LP_FLEXCOMM6_RX_EN	LP_FLEXCOMM6 RX FIFO Drain Enable 0b - Disables 1b - Enables
24	LP_FLEXCOMM5 TX FIFO Fill Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM5_TX_EN	0b - Disables 1b - Enables
23 LP_FLEXCOMM5_RX_EN	LP_FLEXCOMM5 RX FIFO Drain Enable 0b - Disables 1b - Enables
22 LP_FLEXCOMM4_TX_EN	LP_FLEXCOMM4 TX FIFO Fill Enable 0b - Disables 1b - Enables
21 LP_FLEXCOMM4_RX_EN	LP_FLEXCOMM4 RX FIFO Drain Enable 0b - Disables 1b - Enables
20 LP_FLEXCOMM3_TX_EN	LP_FLEXCOMM3 TX FIFO Fill Enable 0b - Disables 1b - Enables
19 LP_FLEXCOMM3_RX_EN	LP_FLEXCOMM3 RX FIFO Drain Enable 0b - Disables 1b - Enables
18 LP_FLEXCOMM2_TX_EN	LP_FLEXCOMM2 TX FIFO Fill Enable 0b - Disables 1b - Enables
17 LP_FLEXCOMM2_RX_EN	LP_FLEXCOMM2 RX FIFO Drain Enable 0b - Disables 1b - Enables
16 LP_FLEXCOMM1_TX_EN	LP_FLEXCOMM1 TX FIFO Fill Enable 0b - Disables 1b - Enables
15 LP_FLEXCOMM1_RX_EN	LP_FLEXCOMM1 RX FIFO Drain Enable 0b - Disables 1b - Enables
14 LP_FLEXCOMM0_TX_EN	LP_FLEXCOMM0 TX FIFO Fill Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 LP_FLEXCOMM0_RX_EN	LP_FLEXCOMM0 RX FIFO Drain Enable 0b - Disables 1b - Enables
12 FLEXIO_SHFT_TIMER7_EN	FLEXIO Shift Timer 7 Enable 0b - Disables 1b - Enables
11 FLEXIO_SHFT_TIMER6_EN	FLEXIO Shift Timer 6 Enable 0b - Disables 1b - Enables
10 FLEXIO_SHFT_TIMER5_EN	FLEXIO Shift Timer 5 Enable 0b - Disables 1b - Enables
9 FLEXIO_SHFT_TIMER4_EN	FLEXIO Shift Timer 4 Enable 0b - Disables 1b - Enables
8 FLEXIO_SHFT_TIMER3_EN	FLEXIO Shift Timer 3 Enable 0b - Disables 1b - Enables
7 FLEXIO_SHFT_TIMER2_EN	FLEXIO Shift Timer 2 Enable 0b - Disables 1b - Enables
6 FLEXIO_SHFT_TIMER1_EN	FLEXIO Shift Timer 1 Enable 0b - Disables 1b - Enables
5 FLEXIO_SHFT_TIMER0_EN	FLEXIO Shift Timer 0 Enable 0b - Disables 1b - Enables
4-3 —	Reserved
2 ACMP_EN	ACMP0 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 SDADC_FIFO3_EN	SDADC FIFO3 Enable 0b - Disables 1b - Enables
0 SDADC_FIFO2_EN	SDADC FIFO2 Enable 0b - Disables 1b - Enables

29.6.1.38 eDMA0 Request Enable 2 (EDMA0_EN2)

Offset

Register	Offset
EDMA0_EN2	428h

Function

Enables Request 2 for eDMA0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO4_D...	GPIO3_D...	GPIO3_D...	GPIO2_D...	GPIO2_D...	GPIO1_D...	GPIO1_D...	GPIO0_D...	GPIO0_D...	SAI2_TX...	SAI2_RX...	SAI1_TX...	SAI1_RX...	SAI0_TX...	SAI0_RX...	I3C1_TX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	I3C1_RX...	I3C0_TX...	I3C0_RX...	LPSPI16...	LPSPI16...	LPSPI14...	LPSPI14...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO4_DMA0_EN	GPIO4 eDMA Request 0 Enable 0b - Disables 1b - Enables
30	GPIO3 eDMA Request 1 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
GPIO3_DMA1_EN	0b - Disables 1b - Enables
29 GPIO3_DMA0_EN	GPIO3 eDMA Request 0 Enable 0b - Disables 1b - Enables
28 GPIO2_DMA1_EN	GPIO2 eDMA Request 1 Enable 0b - Disables 1b - Enables
27 GPIO2_DMA0_EN	GPIO2 eDMA Request 0 Enable 0b - Disables 1b - Enables
26 GPIO1_DMA1_EN	GPIO1 eDMA Request 1 Enable 0b - Disables 1b - Enables
25 GPIO1_DMA0_EN	GPIO1 eDMA Request 0 Enable 0b - Disables 1b - Enables
24 GPIO0_DMA1_EN	GPIO0 eDMA Request 1 Enable 0b - Disables 1b - Enables
23 GPIO0_DMA0_EN	GPIO0 eDMA Request 0 Enable 0b - Disables 1b - Enables
22 SAI2_TX_EN	SAI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
21 SAI2_RX_EN	SAI2 RX FIFO Drain Enable 0b - Disables 1b - Enables
20 SAI1_TX_EN	SAI1 TX FIFO Fill Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
19 SAI1_RX_EN	SAI1 RX FIFO Drain Enable 0b - Disables 1b - Enables
18 SAI0_TX_EN	SAI0 TX FIFO Fill Enable 0b - Disables 1b - Enables
17 SAI0_RX_EN	SAI0 RX FIFO Drain Enable 0b - Disables 1b - Enables
16 I3C1_TX_EN	I3C1 TX FIFO Fill Enable 0b - Disables 1b - Enables
15 I3C1_RX_EN	I3C1 RX FIFO Drain Enable 0b - Disables 1b - Enables
14 I3C0_TX_EN	I3C0 TX FIFO Fill Enable 0b - Disables 1b - Enables
13 I3C0_RX_EN	I3C0 RX FIFO Drain Enable 0b - Disables 1b - Enables
12 LPSPI16_TX_EN	LPSPI16 TX FIFO Fill Enable 0b - Disables 1b - Enables
11 LPSPI16_RX_EN	LPSPI16 RX FIFO Drain Enable 0b - Disables 1b - Enables
10 LPSPI14_TX_EN	LPSPI14 TX FIFO Fill Enable 0b - Disables 1b - Enables
9	LPSPI14 RX FIFO Drain Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPSPI14_RX_EN	0b - Disables 1b - Enables
8 LP_FLEXCOMM13_TX_EN	LP_FLEXCOMM13 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 LP_FLEXCOMM13_RX_EN	LP_FLEXCOMM13 RX FIFO Drain Enable 0b - Disables 1b - Enables
6 LP_FLEXCOMM12_TX_EN	LP_FLEXCOMM12 TX FIFO Fill Enable 0b - Disables 1b - Enables
5 LP_FLEXCOMM12_RX_EN	LP_FLEXCOMM12 RX FIFO Drain Enable 0b - Disables 1b - Enables
4 LP_FLEXCOMM11_TX_EN	LP_FLEXCOMM11 TX FIFO Fill Enable 0b - Disables 1b - Enables
3 LP_FLEXCOMM11_RX_EN	LP_FLEXCOMM11 RX FIFO Drain Enable 0b - Disables 1b - Enables
2 LP_FLEXCOMM10_TX_EN	LP_FLEXCOMM10 TX FIFO Fill Enable 0b - Disables 1b - Enables
1 LP_FLEXCOMM10_RX_EN	LP_FLEXCOMM10 RX FIFO Drain Enable 0b - Disables 1b - Enables
0 LP_FLEXCOMM9_TX_EN	LP_FLEXCOMM9 TX FIFO Fill Enable 0b - Disables 1b - Enables

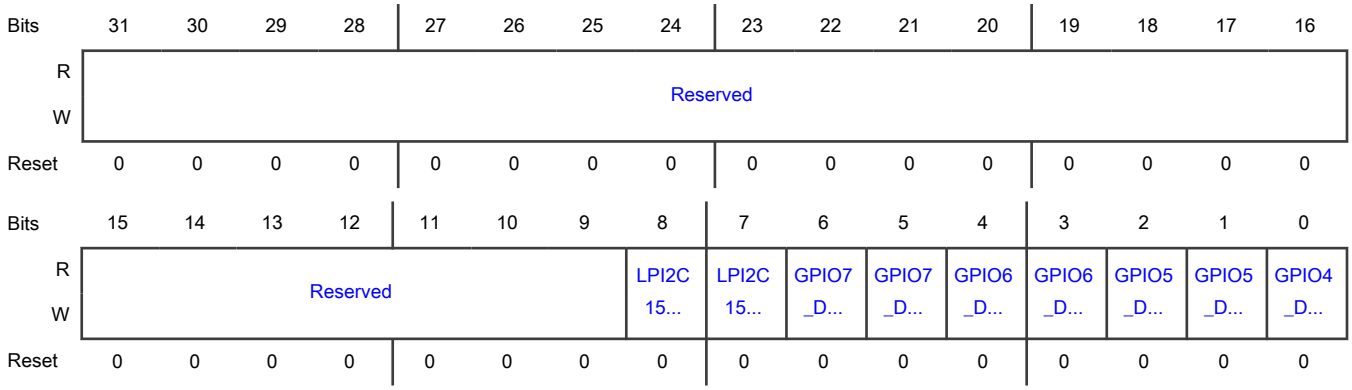
29.6.1.39 eDMA0 Request Enable 3 (EDMA0_EN3)

Offset

Register	Offset
EDMA0_EN3	42Ch

Function
Enables Request 3 for eDMA0.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 LPI2C15_TX_EN	LPI2C15 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 LPI2C15_RX_EN	LPI2C15 RX FIFO Drain Enable 0b - Disables 1b - Enables
6 GPIO7_DMA1_EN	GPIO7 eDMA Request 1 Enable 0b - Disables 1b - Enables
5 GPIO7_DMA0_EN	GPIO7 eDMA Request 0 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 GPIO6_DMA1_EN	GPIO6 eDMA Request 1 Enable 0b - Disables 1b - Enables
3 GPIO6_DMA0_EN	GPIO6 eDMA Request 0 Enable 0b - Disables 1b - Enables
2 GPIO5_DMA1_EN	GPIO5 eDMA Request 1 Enable 0b - Disables 1b - Enables
1 GPIO5_DMA0_EN	GPIO5 eDMA Request 0 Enable 0b - Disables 1b - Enables
0 GPIO4_DMA1_EN	GPIO4 eDMA Request 1 Enable 0b - Disables 1b - Enables

29.6.1.40 eDMA1 Request Enable 0 (EDMA1_EN0)

Offset

Register	Offset
EDMA1_EN0	430h

Function

Enables Request 0 for eDMA1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SDADC_F...	SDADC_F...	ADC0_FI...	ADC0_FI...	SCT0_DM...	SCT0_DM...	CTIMER4...	CTIMER4...	CTIMER3...	CTIMER3...	CTIMER2...	CTIMER2...	CTIMER1...	CTIMER1...	CTIMER0...	CTIMER0...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	PINT0_I...	XSPI2_T...	XSPI2_R...	XSPI1_T...	XSPI1_R...	XSPI0_T...	XSPI0_R...	MICFIL_...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 SDADC_FIFO1_EN	SDADC FIFO1 Enable 0b - Disables 1b - Enables
30 SDADC_FIFO0_EN	SDADC FIFO0 Enable 0b - Disables 1b - Enables
29 ADC0_FIFO1_EN	ADC0 FIFO1 Enable 0b - Disables 1b - Enables
28 ADC0_FIFO0_EN	ADC0 FIFO0 Enable 0b - Disables 1b - Enables
27 SCT0_DMAB_EN	SCT0 DMA B Enable 0b - Disables 1b - Enables
26 SCT0_DMAA_EN	SCT0 DMA A Enable 0b - Disables 1b - Enables
25 CTIMER4_M1_EN	CTIMER4 M1 Enable 0b - Disables 1b - Enables
24 CTIMER4_M0_EN	CTIMER4 M0 Enable 0b - Disables 1b - Enables
23 CTIMER3_M1_EN	CTIMER3 M1 Enable 0b - Disables 1b - Enables
22 CTIMER3_M0_EN	CTIMER3 M0 Enable 0b - Disables 1b - Enables
21	CTIMER2 M1 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER2_M1_EN	0b - Disables 1b - Enables
20 CTIMER2_M0_EN	CTIMER2 M0 Enable 0b - Disables 1b - Enables
19 CTIMER1_M1_EN	CTIMER1 M1 Enable 0b - Disables 1b - Enables
18 CTIMER1_M0_EN	CTIMER1 M0 Enable 0b - Disables 1b - Enables
17 CTIMER0_M1_EN	CTIMER0 M1 Enable 0b - Disables 1b - Enables
16 CTIMER0_M0_EN	CTIMER0 M0 Enable 0b - Disables 1b - Enables
15 PINT0_IRQ7_EN	PINT0 Interrupt 7 Enable 0b - Disables 1b - Enables
14 PINT0_IRQ6_EN	PINT0 Interrupt 6 Enable 0b - Disables 1b - Enables
13 PINT0_IRQ5_EN	PINT0 Interrupt 5 Enable 0b - Disables 1b - Enables
12 PINT0_IRQ4_EN	PINT0 Interrupt 4 Enable 0b - Disables 1b - Enables
11 PINT0_IRQ3_EN	PINT0 Interrupt 3 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 PINT0_IRQ2_EN	PINT0 Interrupt 2 Enable 0b - Disables 1b - Enables
9 PINT0_IRQ1_EN	PINT0 Interrupt 1 Enable 0b - Disables 1b - Enables
8 PINT0_IRQ0_EN	PINT0 Interrupt 0 Enable 0b - Disables 1b - Enables
7 XSPI2_TX_EN	XSPI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
6 XSPI2_RX_EN	XSPI2 RX FIFO Drain Enable 0b - Disables 1b - Enables
5 XSPI1_TX_EN	XSPI1 TX FIFO Fill Enable 0b - Disables 1b - Enables
4 XSPI1_RX_EN	XSPI1 RX FIFO Drain Enable 0b - Disables 1b - Enables
3 XSPI0_TX_EN	XSPI0 TX FIFO Fill Enable 0b - Disables 1b - Enables
2 XSPI0_RX_EN	XSPI0 RX FIFO Drain Enable 0b - Disables 1b - Enables
1 MICFIL_EN	MICFIL Enable 0b - Disables 1b - Enables
0	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	

29.6.1.41 eDMA1 Request Enable 1 (EDMA1_EN1)

Offset

Register	Offset
EDMA1_EN1	434h

Function
Enables Request 1 for eDMA1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LP_FLEX...	LP_FLEX...	LP_FLEX...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	Reserved		ACMP_EN	SDADC_F...	SDADC_F...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LP_FLEXCOMM9_RX_EN	LP_FLEXCOMM9 RX FIFO Drain Enable 0b - Disables 1b - Enables
30 LP_FLEXCOMM8_TX_EN	LP_FLEXCOMM8 TX FIFO Fill Enable 0b - Disables 1b - Enables
29 LP_FLEXCOMM8_RX_EN	LP_FLEXCOMM8 RX FIFO Drain Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
28 LP_FLEXCOMM7_TX_EN	LP_FLEXCOMM7 TX FIFO Fill Enable 0b - Disables 1b - Enables
27 LP_FLEXCOMM7_RX_EN	LP_FLEXCOMM7 RX FIFO Drain Enable 0b - Disables 1b - Enables
26 LP_FLEXCOMM6_TX_EN	LP_FLEXCOMM6 TX FIFO Fill Enable 0b - Disables 1b - Enables
25 LP_FLEXCOMM6_RX_EN	LP_FLEXCOMM6 RX FIFO Drain Enable 0b - Disables 1b - Enables
24 LP_FLEXCOMM5_TX_EN	LP_FLEXCOMM5 TX FIFO Fill Enable 0b - Disables 1b - Enables
23 LP_FLEXCOMM5_RX_EN	LP_FLEXCOMM5 RX FIFO Drain Enable 0b - Disables 1b - Enables
22 LP_FLEXCOMM4_TX_EN	LP_FLEXCOMM4 TX FIFO Fill Enable 0b - Disables 1b - Enables
21 LP_FLEXCOMM4_RX_EN	LP_FLEXCOMM4 RX FIFO Drain Enable 0b - Disables 1b - Enables
20 LP_FLEXCOMM3_TX_EN	LP_FLEXCOMM3 TX FIFO Fill Enable 0b - Disables 1b - Enables
19 LP_FLEXCOMM3_RX_EN	LP_FLEXCOMM3 RX FIFO Drain Enable 0b - Disables 1b - Enables
18	LP_FLEXCOMM2 TX FIFO Fill Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM2_TX_EN	0b - Disables 1b - Enables
17 LP_FLEXCOMM2_RX_EN	LP_FLEXCOMM2 RX FIFO Drain Enable 0b - Disables 1b - Enables
16 LP_FLEXCOMM1_TX_EN	LP_FLEXCOMM1 TX FIFO Fill Enable 0b - Disables 1b - Enables
15 LP_FLEXCOMM1_RX_EN	LP_FLEXCOMM1 RX FIFO Drain Enable 0b - Disables 1b - Enables
14 LP_FLEXCOMM0_TX_EN	LP_FLEXCOMM0 TX FIFO Fill Enable 0b - Disables 1b - Enables
13 LP_FLEXCOMM0_RX_EN	LP_FLEXCOMM0 RX FIFO Drain Enable 0b - Disables 1b - Enables
12 FLEXIO_SHFT_TIMER7_EN	FLEXIO Shift Timer 7 Enable 0b - Disables 1b - Enables
11 FLEXIO_SHFT_TIMER6_EN	FLEXIO Shift Timer 6 Enable 0b - Disables 1b - Enables
10 FLEXIO_SHFT_TIMER5_EN	FLEXIO Shift Timer 5 Enable 0b - Disables 1b - Enables
9 FLEXIO_SHFT_TIMER4_EN	FLEXIO Shift Timer 4 Enable 0b - Disables 1b - Enables
8 FLEXIO_SHFT_TIMER3_EN	FLEXIO Shift Timer 3 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 FLEXIO_SHFT_TIMER2_EN	FLEXIO Shift Timer 2 Enable 0b - Disables 1b - Enables
6 FLEXIO_SHFT_TIMER1_EN	FLEXIO Shift Timer 1 Enable 0b - Disables 1b - Enables
5 FLEXIO_SHFT_TIMER0_EN	FLEXIO Shift Timer 0 Enable 0b - Disables 1b - Enables
4-3 —	Reserved
2 ACMP_EN	ACMP0 Enable 0b - Disables 1b - Enables
1 SDADC_FIFO3_EN	SDADC FIFO3 Enable 0b - Disables 1b - Enables
0 SDADC_FIFO2_EN	SDADC FIFO2 Enable 0b - Disables 1b - Enables

29.6.1.42 eDMA1 Request Enable 2 (EDMA1_EN2)

Offset

Register	Offset
EDMA1_EN2	438h

Function
Enables Request 2 for eDMA1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPIO4_D...	GPIO3_D...	GPIO3_D...	GPIO2_D...	GPIO2_D...	GPIO1_D...	GPIO1_D...	GPIO0_D...	GPIO0_D...	SAI2_TX...	SAI2_RX...	SAI1_TX...	SAI1_RX...	SAI0_TX...	SAI0_RX...	I3C1_TX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	I3C1_RX...	I3C0_TX...	I3C0_RX...	LPSPI16...	LPSPI16...	LPSPI14...	LPSPI14...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO4_DMA0_EN	GPIO4 eDMA Request 0 Enable 0b - Disables 1b - Enables
30 GPIO3_DMA1_EN	GPIO3 eDMA Request 1 Enable 0b - Disables 1b - Enables
29 GPIO3_DMA0_EN	GPIO3 eDMA Request 0 Enable 0b - Disables 1b - Enables
28 GPIO2_DMA1_EN	GPIO2 eDMA Request 1 Enable 0b - Disables 1b - Enables
27 GPIO2_DMA0_EN	GPIO2 eDMA Request 0 Enable 0b - Disables 1b - Enables
26 GPIO1_DMA1_EN	GPIO1 eDMA Request 1 Enable 0b - Disables 1b - Enables
25 GPIO1_DMA0_EN	GPIO1 eDMA Request 0 Enable 0b - Disables 1b - Enables
24	GPIO0 eDMA Request 1 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
GPIO0_DMA1_EN	0b - Disables 1b - Enables
23 GPIO0_DMA0_EN	GPIO0 eDMA Request 0 Enable 0b - Disables 1b - Enables
22 SAI2_TX_EN	SAI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
21 SAI2_RX_EN	SAI2 RX FIFO Drain Enable 0b - Disables 1b - Enables
20 SAI1_TX_EN	SAI1 TX FIFO Fill Enable 0b - Disables 1b - Enables
19 SAI1_RX_EN	SAI1 RX FIFO Drain Enable 0b - Disables 1b - Enables
18 SAI0_TX_EN	SAI0 TX FIFO Fill Enable 0b - Disables 1b - Enables
17 SAI0_RX_EN	SAI0 RX FIFO Drain Enable 0b - Disables 1b - Enables
16 I3C1_TX_EN	I3C1 TX FIFO Fill Enable 0b - Disables 1b - Enables
15 I3C1_RX_EN	I3C1 RX FIFO Drain Enable 0b - Disables 1b - Enables
14 I3C0_TX_EN	I3C0 TX FIFO Fill Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 I3C0_RX_EN	I3C0 RX FIFO Drain Enable 0b - Disables 1b - Enables
12 LPSPI16_TX_EN	LPSPI16 TX FIFO Fill Enable 0b - Disables 1b - Enables
11 LPSPI16_RX_EN	LPSPI16 RX FIFO Drain Enable 0b - Disables 1b - Enables
10 LPSPI14_TX_EN	LPSPI14 TX FIFO Fill Enable 0b - Disables 1b - Enables
9 LPSPI14_RX_EN	LPSPI14 RX FIFO Drain Enable 0b - Disables 1b - Enables
8 LP_FLEXCOMM13_TX_EN	LP_FLEXCOMM13 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 LP_FLEXCOMM13_RX_EN	LP_FLEXCOMM13 RX FIFO Drain Enable 0b - Disables 1b - Enables
6 LP_FLEXCOMM12_TX_EN	LP_FLEXCOMM12 TX FIFO Fill Enable 0b - Disables 1b - Enables
5 LP_FLEXCOMM12_RX_EN	LP_FLEXCOMM12 RX FIFO Drain Enable 0b - Disables 1b - Enables
4 LP_FLEXCOMM11_TX_EN	LP_FLEXCOMM11 TX FIFO Fill Enable 0b - Disables 1b - Enables
3	LP_FLEXCOMM11 RX FIFO Drain Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM11_RX_EN	0b - Disables 1b - Enables
2 LP_FLEXCOMM10_TX_EN	LP_FLEXCOMM10 TX FIFO Fill Enable 0b - Disables 1b - Enables
1 LP_FLEXCOMM10_RX_EN	LP_FLEXCOMM10 RX FIFO Drain Enable 0b - Disables 1b - Enables
0 LP_FLEXCOMM9_TX_EN	LP_FLEXCOMM9 TX FIFO Fill Enable 0b - Disables 1b - Enables

29.6.1.43 eDMA1 Request Enable 3 (EDMA1_EN3)

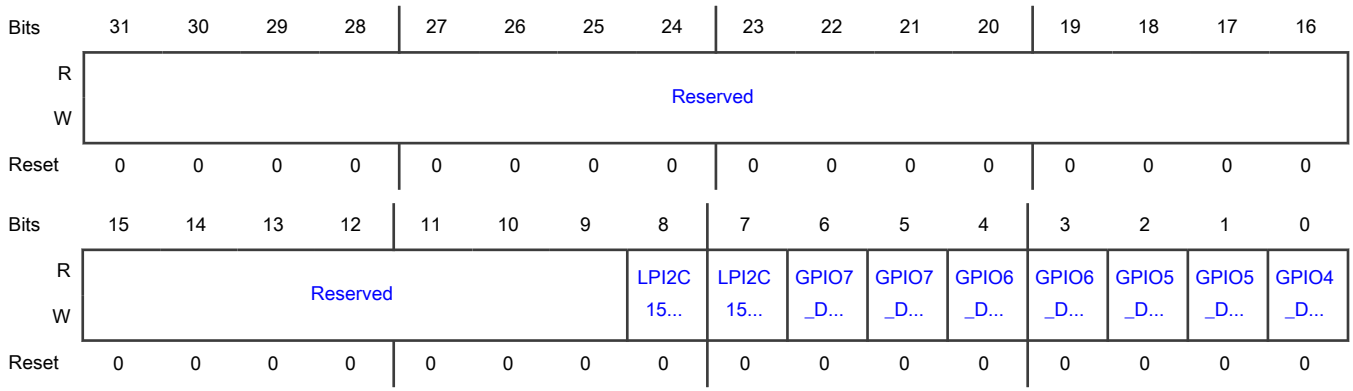
Offset

Register	Offset
EDMA1_EN3	43Ch

Function

Enables Request 3 for eDMA1.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 LPI2C15_TX_EN	LPI2C15 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 LPI2C15_RX_EN	LPI2C15 RX FIFO Drain Enable 0b - Disables 1b - Enables
6 GPIO7_DMA1_EN	GPIO7 eDMA Request 1 Enable 0b - Disables 1b - Enables
5 GPIO7_DMA0_EN	GPIO7 eDMA Request 0 Enable 0b - Disables 1b - Enables
4 GPIO6_DMA1_EN	GPIO6 eDMA Request 1 Enable 0b - Disables 1b - Enables
3 GPIO6_DMA0_EN	GPIO6 eDMA Request 0 Enable 0b - Disables 1b - Enables
2 GPIO5_DMA1_EN	GPIO5 eDMA Request 1 Enable 0b - Disables 1b - Enables
1 GPIO5_DMA0_EN	GPIO5 eDMA Request 0 Enable 0b - Disables 1b - Enables
0 GPIO4_DMA1_EN	GPIO4 eDMA Request 1 Enable 0b - Disables 1b - Enables

29.6.1.44 AXBS Control (AXBS_CTRL)

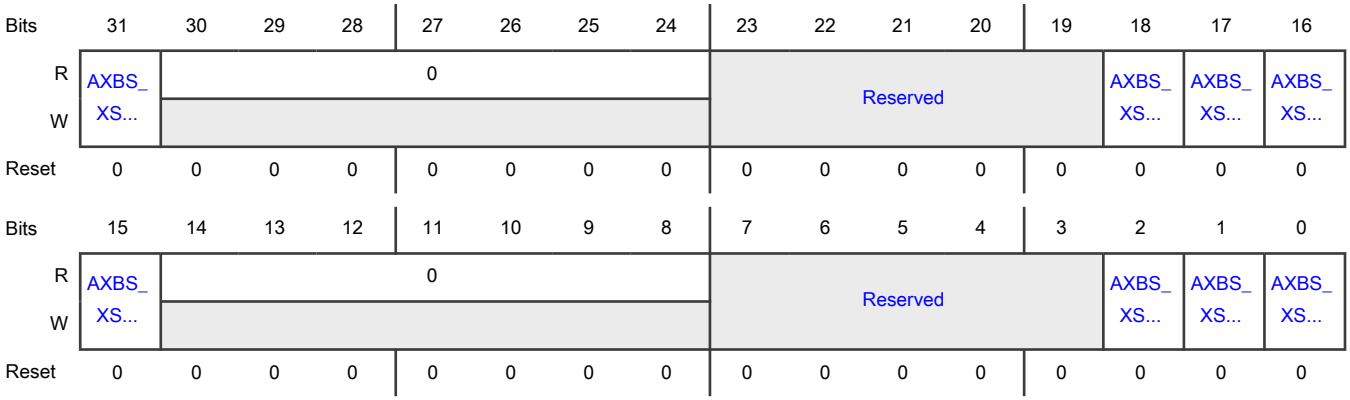
Offset

Register	Offset
AXBS_CTRL	600h

Function

Controls Crossbar Switch (AXBS) target arbitration for XSPI0 and XSPI1. For detailed information on target arbitration with fixed priority and round-robin modes, see the AXBS chapter.

Diagram



Fields

Field	Function
31 AXBS_XSPI1_FORCE_RR	AXBS for XSPI1 Force Round-Robin Controls if the XSPI1 AXBS switch will use round-robin or fixed priority arbitration. 0b - Fixed priority arbitration 1b - Round-robin arbitration
30-24 —	Reserved
23-19 —	Reserved
18 AXBS_XSPI1_PRIORITY_M2	AXBS for XSPI1 M2 Priority Controls the priority for NIC_DSP bus initiators accessing XSPI1 (accesses are not cached). When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High

Table continues on the next page...

Table continued from the previous page...

Field	Function
17 AXBS_XSPI1_PRIORITY_M1	AXBS for XSPI1 M1 Priority Controls the priority for NIC_MEDIA1 bus initiators accessing XSPI1 (accesses are not cached). When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High
16 AXBS_XSPI1_PRIORITY_M0	AXBS for XSPI1 M0 Priority Controls the priority for M-bus initiators accessing XSPI1 (accesses can be cached by CACHE64_CTRL1). When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High
15 AXBS_XSPI0_FORCE_RR	AXBS for XSPI0 Force Round-Robin Controls if the XSPI0 AXBS switch will use round-robin or fixed priority arbitration. 0b - Fixed priority arbitration 1b - Round-robin arbitration
14-8 —	Reserved
7-3 —	Reserved
2 AXBS_XSPI0_PRIORITY_M2	AXBS for XSPI0 M2 Priority Controls the priority for NIC_DSP bus initiators accessing XSPI0 (accesses are not cached). When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High
1 AXBS_XSPI0_PRIORITY_M1	AXBS for XSPI0 M1 Priority Controls the priority for NIC_MEDIA1 bus initiators accessing XSPI0 (accesses are not cached). When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High
0 AXBS_XSPI0_PRIORITY_M0	AXBS for XSPI0 M0 Priority Controls the priority for M-bus initiators accessing XSPI0 (accesses can be cached by CACHE64_CTRL0).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When the priority is the same, the initiator with the lower M number is given priority. 0b - Low 1b - High

29.6.1.45 I3C Asynchronous Wake-up Control (I3C_ASYNC_WAKEUP_CTRL)

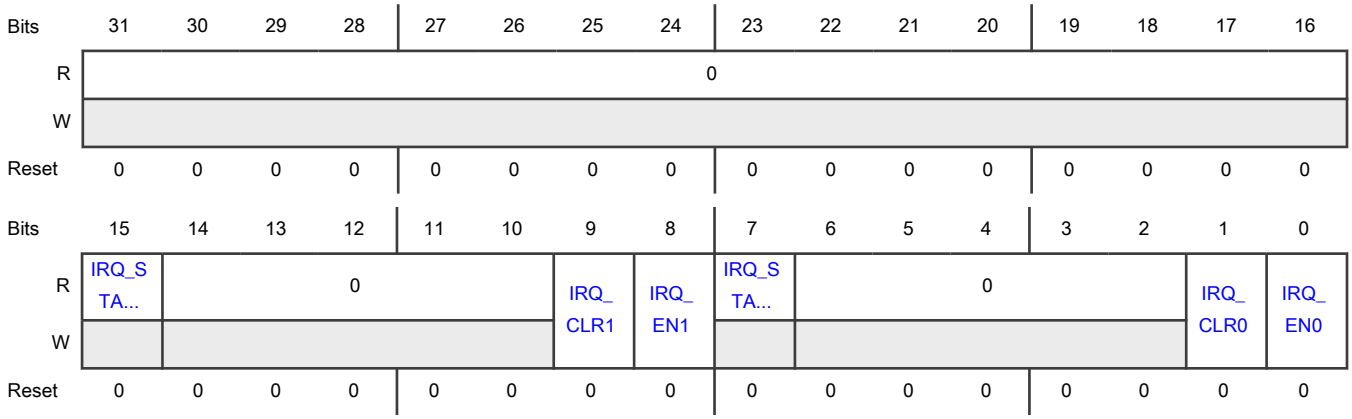
Offset

Register	Offset
I3C_ASYNC_WAKEUP_CTRL	628h

Function

Controls Improved Inter-Integrated Circuit 0-1 (I3C0 and I3C1) asynchronous wake-up in VDD2_COMP domain.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 IRQ_STATUS1	I3C1 Asynchronous Wake-up Interrupt Status For controller, the interrupt is enabled by [IRQ_EN1] bit. For target mode, the interrupt is enabled by I3C1.SINTSET[MATCHED]. 0b - Interrupt disabled 1b - Interrupt enabled

Table continues on the next page...

Table continued from the previous page...

Field	Function
14-10 —	Reserved
9 IRQ_CLR1	I3C1 Asynchronous Wake-up Interrupt Clear The interrupt will be held until this field is set to 1. 0b - Interrupt not cleared 1b - Interrupt cleared
8 IRQ_EN1	I3C1 Controller Mode Asynchronous Wake-up Interrupt Enable If enabled, the interrupt happens when I3C1 is in controller mode, FCLK is stopped and SDA is driven low by target. 0b - Interrupt disabled 1b - Interrupt enabled
7 IRQ_STATUS0	I3C0 Asynchronous Wake-up Interrupt Status For controller, the interrupt is enabled by [IRQ_EN0] bit. For target mode, the interrupt is enabled by I3C0.SINTSET[MATCHED]. 0b - Interrupt not asserted 1b - Interrupt asserted
6-2 —	Reserved
1 IRQ_CLR0	I3C0 Asynchronous Wake-up Interrupt Clear The interrupt will be held until this field is set to 1. 0b - Interrupt not cleared 1b - Interrupt cleared
0 IRQ_EN0	I3C0 Controller Mode Asynchronous Wake-up Interrupt Enable If enabled, the interrupt happens when I3C0 is in controller mode, FCLK is stopped and SDA is driven low by target. 0b - Interrupt disabled 1b - Interrupt enabled

29.6.1.46 Gray to Binary Converter - Gray Code [31:0] (GRAY_CODE_LSB)

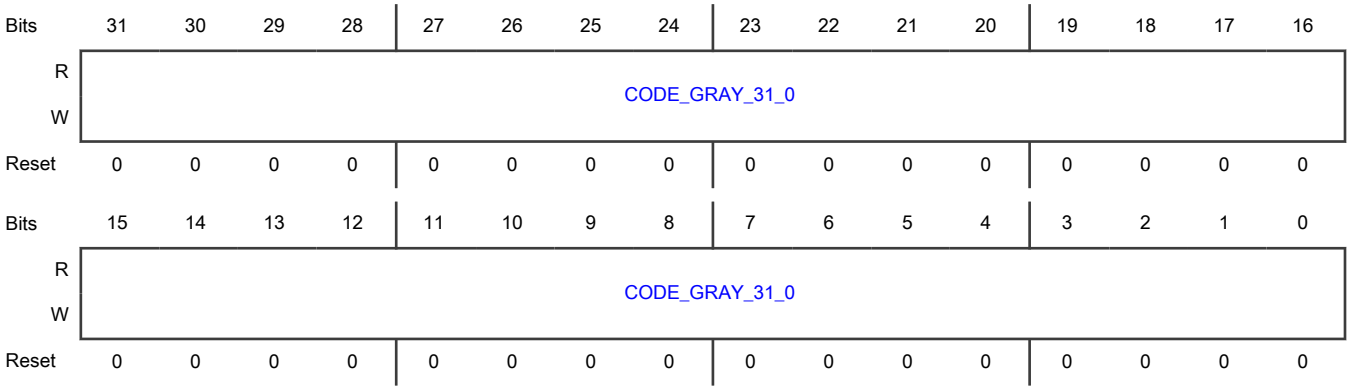
Offset

Register	Offset
GRAY_CODE_LSB	650h

Function

Contains the least-significant portion of the Gray code to be converted back to binary. CPU0 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_GRAY_31_0	Gray Code [31:0] Indicates the least-significant 32 bits of the 64-bit Gray code to be converted.

29.6.1.47 Gray to Binary Converter - Gray Code [63:32] (GRAY_CODE_MSB)

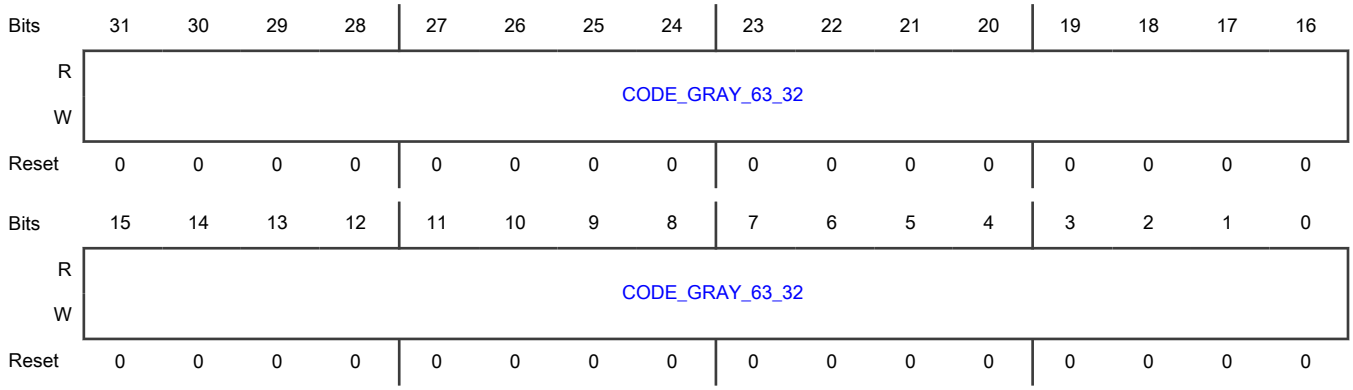
Offset

Register	Offset
GRAY_CODE_MSB	654h

Function

Contains the most-significant portion of the Gray code to be converted back to binary. CPU0 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_GRAY_63_32	Gray Code [63:32] Indicates the most-significant 32 bits of the 64-bit Gray code to be converted.

29.6.1.48 Gray to Binary Converter - Binary Code [31:0] (BINARY_CODE_LSB)

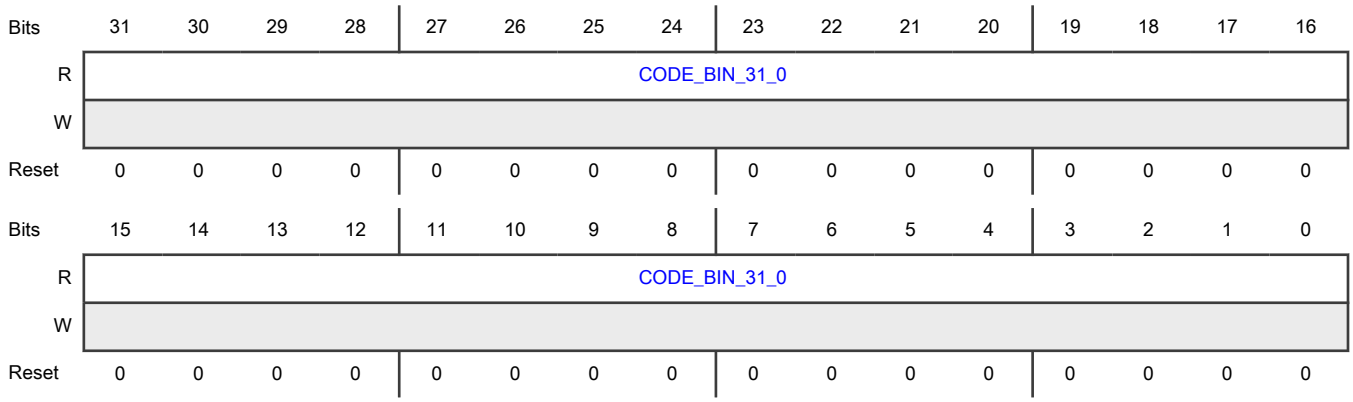
Offset

Register	Offset
BINARY_CODE_LSB	658h

Function

Contains the least-significant portion of the code converted from Gray to binary coding. CPU0 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_BIN_31_0	Binary Code [31:0] Indicates the least-significant 32 bits of the 64-bit converted code.

29.6.1.49 Gray to Binary Converter - Binary Code [63:32] (BINARY_CODE_MSB)

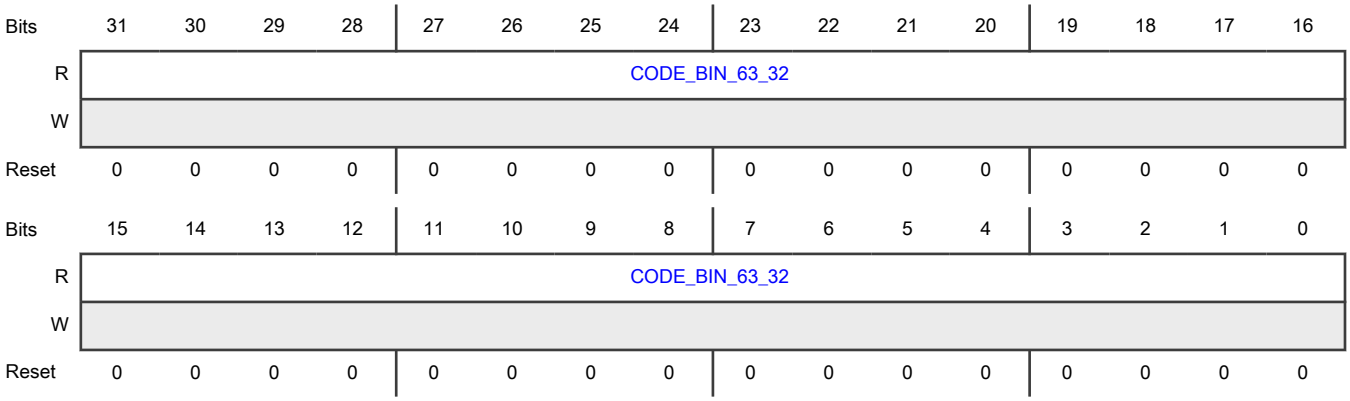
Offset

Register	Offset
BINARY_CODE_MSB	65Ch

Function

Contains the most-significant portion of the code converted from Gray to binary coding. CPU0 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_BIN_63_32	Binary Code [63:32] Indicates the most-significant 32 bits of the 64-bit converted code.

29.6.1.50 ELS Boot State 0 (ELS_BOOT_STATE0)

Offset

Register	Offset
ELS_BOOT_STATE0	B38h

Function

Used by Boot-ROM to record the state of boot process. When user application starts executing, it is possible to check status of the ROM operations by reading this register and checking the corresponding status bit.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DED_ERR_DET			BOOT_PIN_STATE			0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XSPI1_I...	XSPI0_I...	ITRC_ZE...	DEBUG_A...	TZ_PRES...	CDI_COMP	CDI_DEV...	SB3_IMA...	IMAGE_TYP		PSRAM		SB_JUMP	BOOT_SRC		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-25 DED_ERR_DET	DED Error Detect 010b - Error detected 101b - No error
24-22 BOOT_PIN_STATE	Boot Pin State Indicates boot pin state at boot time. 000b - kBootPinSrc_eMMC_uSDHC0 001b - kBootSrc_XSPI0 010b - kBootSrc_ISP_AUTO 011b - kBootSrc_XSPI1 100b - Boot pin not working
21-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
15 XSPI1_IPED_EN	XSPI1 IPED Enable 0b - Disabled 1b - Enabled
14 XSPI0_IPED_EN	XSPI0 IPED Enable 0b - Disabled 1b - Enabled
13 ITRC_ZEROIZE	ITRC Zeroize Event Indicates whether ITRC zeroize event is handled in this session of boot. 0b - Fail 1b - ITRC zeroize done
12 DEBUG_AUTH	Debug Authentication Indicates whether debug authentication is done in this session prior to boot. 0b - Fail 1b - Debug authentication done
11 TZ_PRESET_DATA	TrustZone Preset Data Indicates whether TrustZone preset data is loaded during this boot. 0b - Fail 1b - TrustZone preset data loaded
10 CDI_COMP	CDI per DICE Specification Indicates whether CDI per DICE specification is computed on this boot. 0b - Fail 1b - CDI compute done
9 CDI_DEV_KEY	CDI Based Device Keys for CSR Harvesting Indicates whether CDI based device keys are derived for CSR harvesting on this boot. 0b - Fail 1b - CDI device keys derived
8 SB3_IMAGE	SB3 Image Receive Indicates whether SB3 image is receved or not. 0b - Fail 1b - Success

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-6 IMAGE_TYP	Image Type 00b - Signed image (ECDSA) 01b - CRC image (CRC)
5-4 PSRAM	Image XIP or Copy-to-PSRAM 00b - PSRAM is disabled 01b - PSRAM is enabled
3 SB_JUMP	SB3 Containing SB_JUMP Command Indicates whether the received SB3 containing SB_JUMP command is used. 0b - Fail 1b - Success
2-0 BOOT_SRC	Boot Image Source Indicates the boot image source used during this boot: XSPI NOR, eMMC, or XSPI NAND. 000b - XSPI NOR instance 0 001b - XSPI NOR instance 1 010b - eMMC instance 0 011b - eMMC instance 1 100b - Recovery SPI flash image 101b - Serial boot image (execute ISP command used) 110b - XSPI NAND instance 0 111b - XSPI NAND instance 1

29.6.1.51 ELS Boot State 1 (ELS_BOOT_STATE1)

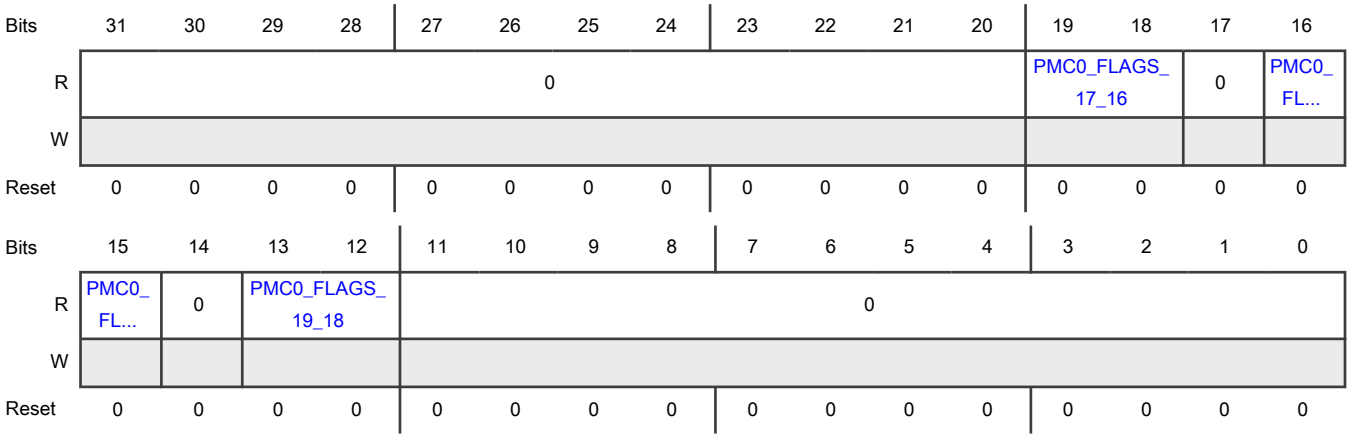
Offset

Register	Offset
ELS_BOOT_STATE1	B3Ch

Function

Used by Boot-ROM to record the state of boot process. When user application starts executing, it is possible to check status of the ROM operations by reading this register and checking the corresponding status bit.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-18 PMC0_FLAGS_17_16	PMC0.FLAGS[17:16] Reflects PMC0.FLAGS[AGDET1F] (VDD1 glitch detector flag) and [AGDET2F] (VDD2 glitch detector flag) at ROM startup.
17 —	Reserved
16-15 PMC0_FLAGS_22_21	PMC0.FLAGS[22:21] Reflects PMC0.FLAGS[HVDVDD1F] (VDD1 high-voltage detector flag) and [HVDVDD2F] (VDD2 high-voltage detector flag) at ROM startup.
14 —	Reserved
13-12 PMC0_FLAGS_19_18	PMC0.FLAGS[19:18] Reflects PMC0.FLAGS[LVDVDD1F] (VDD1 low-voltage detector flag) and [LVDVDD2F] (VDD2 low-voltage detector flag) at ROM startup.
11-0 —	Reserved

29.6.1.52 ELS Boot State 2 (ELS_BOOT_STATE2)

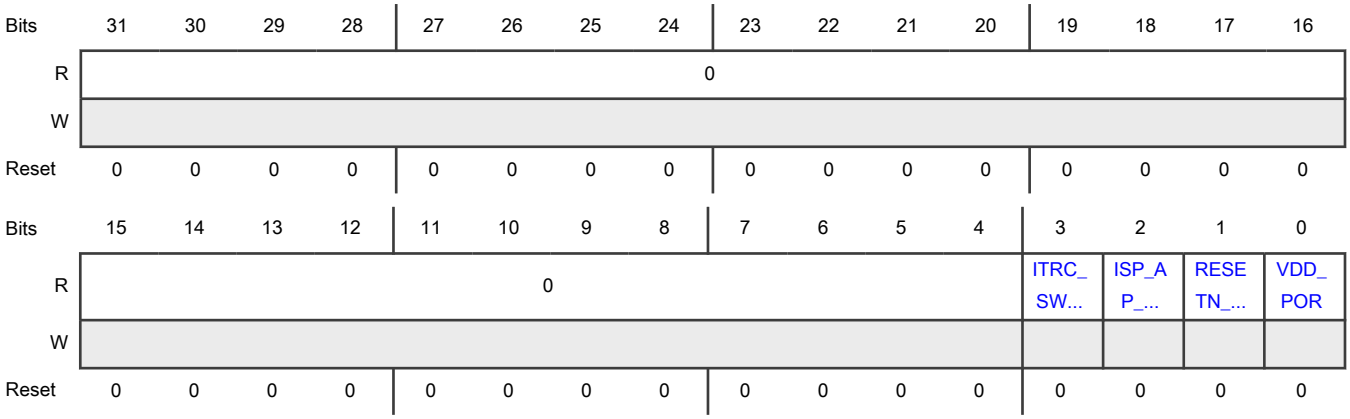
Offset

Register	Offset
ELS_BOOT_STATE2	B40h

Function

Used by Boot-ROM to record the state of boot process. When user application starts executing, it is possible to check status of the ROM operations by reading this register and checking the corresponding status bit.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 ITRC_SW_RST	ITRC_SW Reset Indicates an ITRC_SW (intrusion and tamper response controller software) reset event. 0b - No reset event 1b - Reset event detected
2 ISP_AP_RST	ISP_AP Reset Indicates an ISP_AP (debug mailbox) reset event. 0b - No reset event 1b - Reset event detected
1 RESETN_RST	RESETN Reset Indicates a RESETN reset event.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No reset event 1b - Reset event detected
0 VDD_POR	VDD POR Indicates a VDD power-on reset event. 0b - No reset event 1b - Reset event detected

29.6.2 SYSCON1 register descriptions

29.6.2.1 SYSCON1 memory map

SYSCON1 base address: 4004_2000h

Offset	Register	Width (In bits)	Access	Reset value
14h	NMI Source Select (NMISRC)	32	RW	0000_0000h
1Ch	CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)	32	RW	0000_0000h
78h	Bus Matrix Priority (AHBMATPRIO)	32	RW	0000_0000h
90h	System Secure Tick Calibration (SYSTEM_STICK_CALIB)	32	RW	0000_0000h
94h	System Non-Secure Tick Calibration (SYSTEM_NSTICK_CALIB)	32	RW	0000_0000h
D0h	GPIO Synchronization Stages (GPIO_PSYNC)	32	RW	0000_0000h
144h	eDMA2 Memory Control (EDMA2_MEM_CTRL)	32	RW	0000_0000h
148h	eDMA3 Memory Control (EDMA3_MEM_CTRL)	32	RW	0000_0000h
240h	SAI3 MCLK IO Direction Control (SAI3_MCLK_CTRL)	32	RW	0000_0000h
300h	HiFi1 DSP Stall (DSPSTALL)	32	RW	0000_0001h
304h	HiFi1 OCDHaltOnReset (OCDHALTONRESET)	32	RW	0000_0000h
308h	HiFi1 General Purpose Register 0 (HIFI1_GPR0)	32	R	0000_0000h
314h	HiFi1 General Purpose Register 1 (HIFI1_GPR1)	32	R	0000_0000h
318h	HiFi1 General Purpose Register 2 (HIFI1_GPR2)	32	R	0000_0000h
31Ch	HiFi1 DSP Vector Remap (DSP_VECT_REMAP)	32	RW	0000_0600h
420h	eDMA2 Request Enable 0 (EDMA2_EN0)	32	RW	0000_0000h
424h	eDMA2 Request Enable 1 (EDMA2_EN1)	32	RW	0000_0000h
430h	eDMA3 Request Enable 0 (EDMA3_EN0)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
434h	eDMA3 Request Enable 1 (EDMA3_EN1)	32	RW	0000_0000h
628h	I3C Asynchronous Wake-up Control (I3C_ASYNC_WAKEUP_CTRL)	32	RW	0000_0000h

29.6.2.2 NMI Source Select (NMISRC)

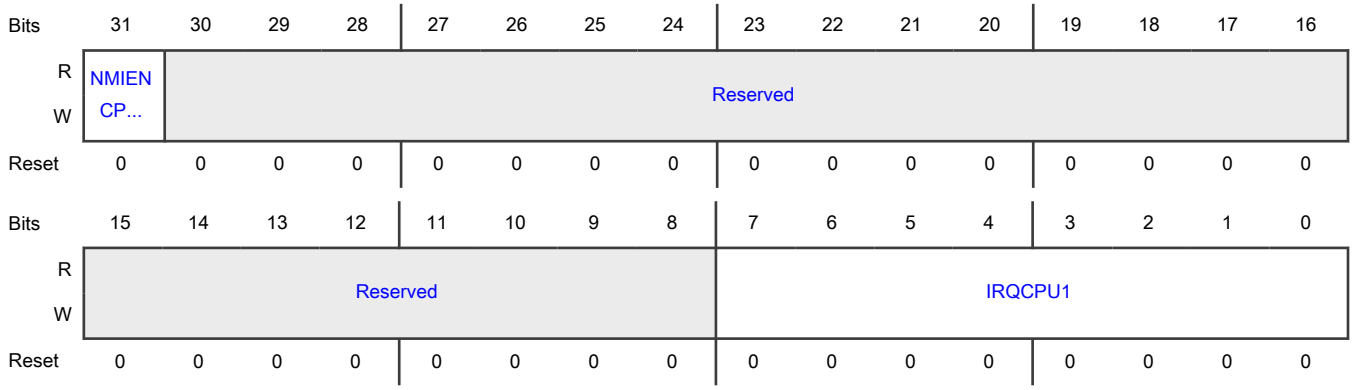
Offset

Register	Offset
NMISRC	14h

Function

Selects peripheral interrupts as source for the Non-Maskable Interrupt (NMI) of CPU1. The NMI source must first be disabled by writing 0 to the NMIENCPU1 bit to change the interrupt source for the NMI. The source can then be changed by updating the IRQCPU1 bits and re-enable the NMI source by setting NMIENCPU1.

Diagram



Fields

Field	Function
31 NMIENCPU1	Enables the Non-Maskable Interrupt (NMI) source selected by IRQCPU1 0b - Disables 1b - Enables
30-8 —	Reserved
7-0 IRQCPU1	The IRQ number of the interrupt that acts as the Non-Maskable Interrupt (NMI) for CPU1, if enabled by NMIENCPU1

29.6.2.3 CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)

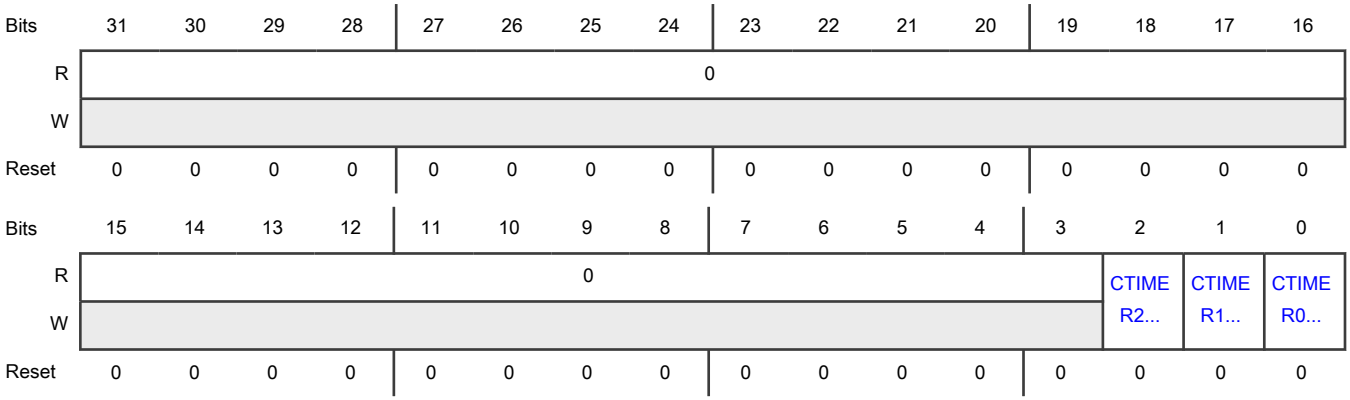
Offset

Register	Offset
CTIMERGLOBALSTARTEN	1Ch

Function

Configures CTIMER function clocks. PSCCTL1 register in CLKCTL1 also controls the clock for these CTIMERs. See CLKCTL chapter for details.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 CTIMER2_CLK_EN	CTIMER7 Function Clock Enable 0b - Disables 1b - Enables
1 CTIMER1_CLK_EN	CTIMER6 Function Clock Enable 0b - Disables 1b - Enables
0 CTIMER0_CLK_EN	CTIMER5 Function Clock Enable 0b - Disables 1b - Enables

29.6.2.4 Bus Matrix Priority (AHBMATPRIO)

Offset

Register	Offset
AHBMATPRIO	78h

Function

The multilayer AHB matrix arbitrates between several initiators, only if they attempt to access the same matrix target port at the same time. Care should be taken if the value in this register is changed, improper settings can seriously degrade performance. Priority values are 0 = highest, 3 = lowest. When the priority is the same, the initiator with the lower M number is given priority. See System bus diagram in Introduction chapter.

NOTE

Be careful when modifying this register as improper settings can seriously degrade the performance.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				SBUS_M5		SBUS_M4		SBUS_M3		SBUS_M2		SBUS_M1		SBUS_M0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11-10 SBUS_M5	S-Bus M5 Priority Controls the priority for HiFi1 accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
9-8 SBUS_M4	S-Bus M4 Priority Controls the priority for eDMA3 accesses.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
7-6 SBUS_M3	S-Bus M3 Priority Controls the priority for eDMA2 accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
5-4 SBUS_M2	S-Bus M2 Priority Controls the priority for CPU1 code bus accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
3-2 SBUS_M1	S-Bus M1 Priority Controls the priority for CPU1 system bus accesses. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3
1-0 SBUS_M0	S-Bus M0 Priority Controls the priority for NIC_MEDIA1 bus initiators going through the S-bus. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3

29.6.2.5 System Secure Tick Calibration (SYSTEM_STICK_CALIB)

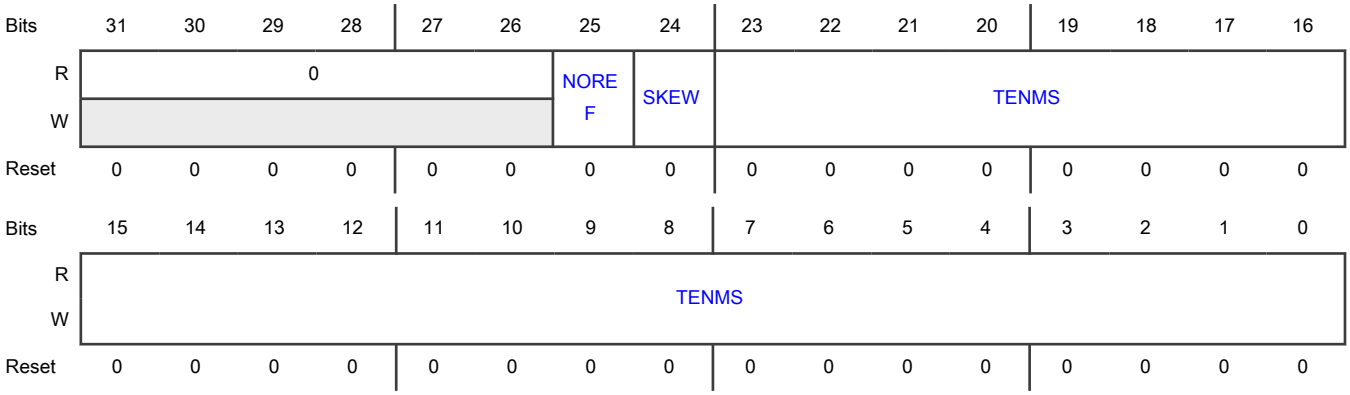
Offset

Register	Offset
SYSTEM_STICK_CALIB	90h

Function

Allows software to set up a default value for the SYST_CALIB register (see Arm documentation) in the System Tick Timer of secure part of CPU1.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 NOREF	No Reference Clock Controls whether the device provides a reference clock to the processor. 0b - Provides 1b - Not provide
24 SKEW	SKEW Controls whether [TENMS] value is exact. 0b - Exact 1b - Inexact, or not given
23-0 TENMS	Ten Milliseconds Reload value for 10 ms (100 Hz) timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

29.6.2.6 System Non-Secure Tick Calibration (SYSTEM_NSTICK_CALIB)

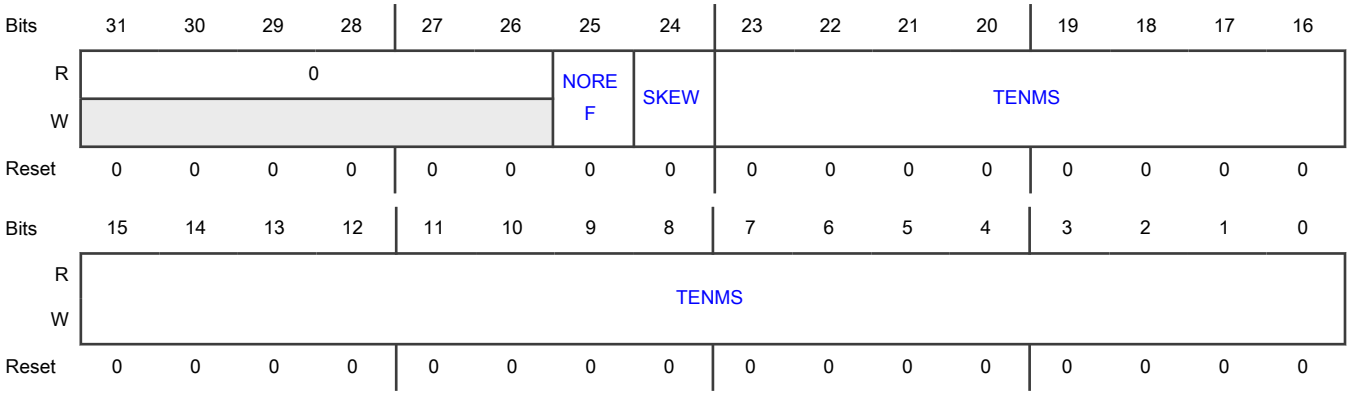
Offset

Register	Offset
SYSTEM_NSTICK_CALIB	94h

Function

Allows software to set up a default value for the SYST_CALIB register in the System Tick Timer of non-secure part of CPU1.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 NOREF	No Reference Clock Controls whether the device provides a reference clock to the processor. 0b - Provides 1b - Not provide
24 SKEW	SKEW Controls whether [TENMS] value is exact. 0b - Exact 1b - Inexact, or not given
23-0 TENMS	Ten Milliseconds Reloads value for 10 ms (100 Hz) timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

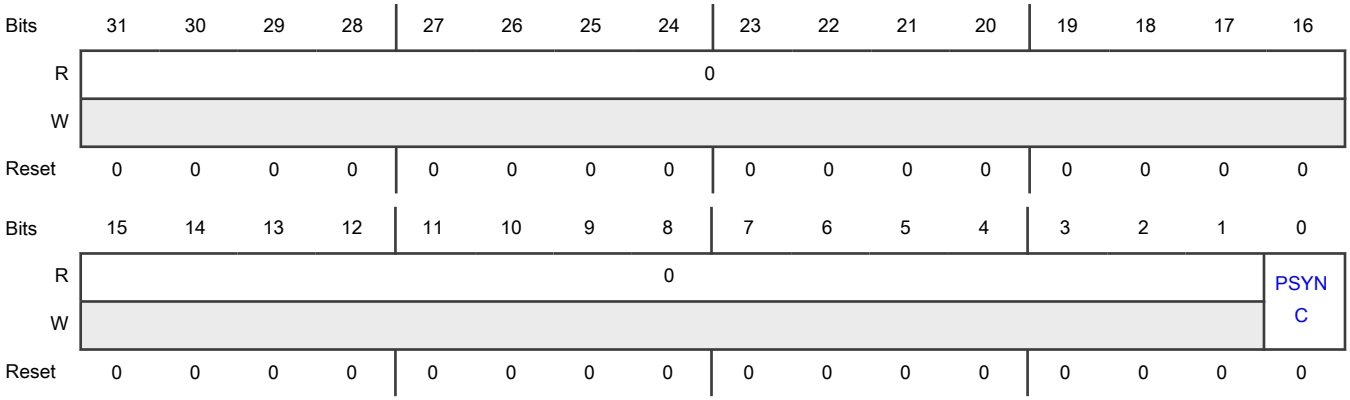
29.6.2.7 GPIO Synchronization Stages (GPIO_PSYNC)

Offset

Register	Offset
GPIO_PSYNC	D0h

Function
Selects the amount of synchronization stages in the GPIO inputs.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 PSYNC	Synchronization Stage Setting The PINT block has two flip-flop synchronizers on the interrupt and pattern match detect circuits based on data inputs from the pads. The first flip-flop stage of synchronization is clocked on the falling edge of the peripheral clock. The second flip-flop stage is clocked on the rising edge of the peripheral clock. The synchronizers can be configured to be 1-stage synchronizers to reduce the signal latency at low peripheral clock frequencies. 0b - 2-stage sync 1b - 1-stage sync

29.6.2.8 eDMA2 Memory Control (EDMA2_MEM_CTRL)

Offset

Register	Offset
EDMA2_MEM_CTRL	144h

Function

Configures input and power gating for eDMA2 memory.

NOTE

Ideally, [PWR_IG] and [PWR_STDBY] fields should be cleared before warm reset if the bits were set. If this is not possible, the application programmer should make sure that [PWR_IG] and [PWR_STDBY] fields are cleared after each warm reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							PWR_IG	PWR_STD...	Reserved						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.2.9 eDMA3 Memory Control (EDMA3_MEM_CTRL)

Offset

Register	Offset
EDMA3_MEM_CTRL	148h

Function

Configures read and write characterization mode, input and power gating for eDMA3 memory.

NOTE

Ideally, [PWR_IG] and [PWR_STDBY] fields should be cleared before warm reset if the bits were set. If this is not possible, the application programmer should make sure that [PWR_IG] and [PWR_STDBY] fields are cleared after each warm reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							PWR_IG	PWR_STD...	0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-9 —	Reserved
8 PWR_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 PWR_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief

Table continues on the next page...

Table continued from the previous page...

Field	Function
	increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.2.10 SAI3 MCLK IO Direction Control (SAI3_MCLK_CTRL)

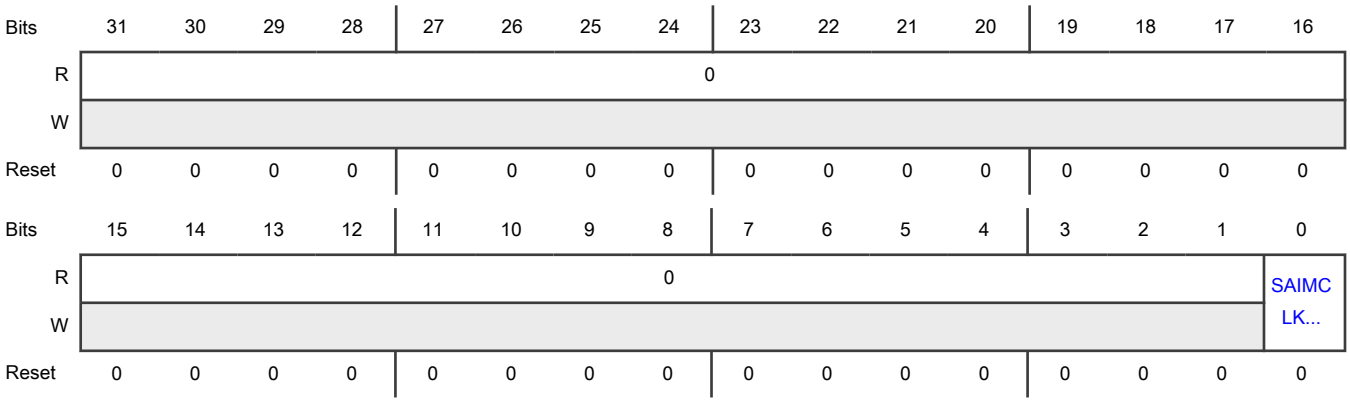
Offset

Register	Offset
SAI3_MCLK_CTRL	240h

Function

Controls MCLK input and output direction for Serial Audio Interface (SAI) 3.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 SAIMCLKDIR	SAI3 MCLK Direction Control Specifies whether MCLK is output or input signal. IOPCTL needs to select MCLK function.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Input 1b - Output

29.6.2.11 HiFi1 DSP Stall (DSPSTALL)

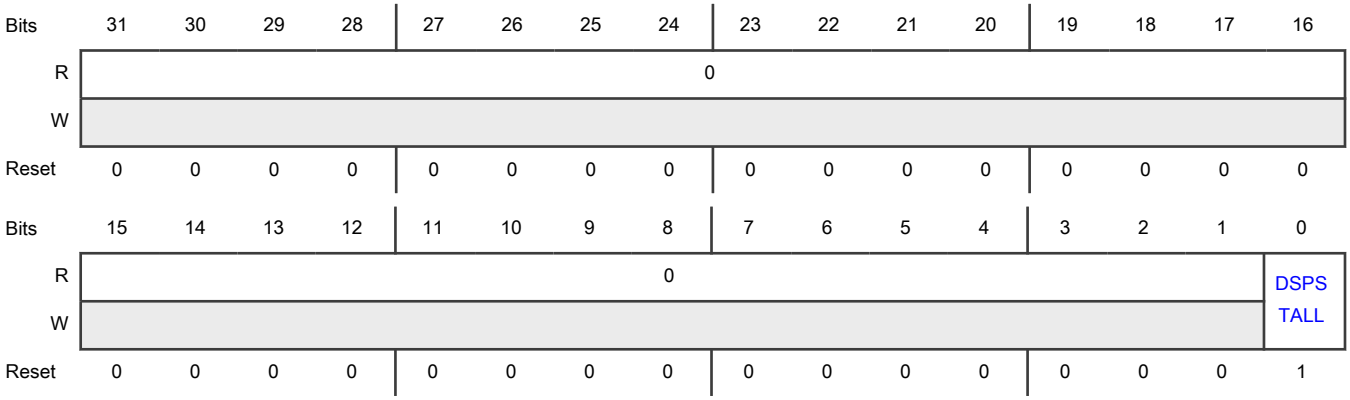
Offset

Register	Offset
DSPSTALL	300h

Function

Controls the operation of HiFi1 DSP.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 DSPSTALL	Run/Stall Control 0b - Run (normal mode) 1b - Stall mode

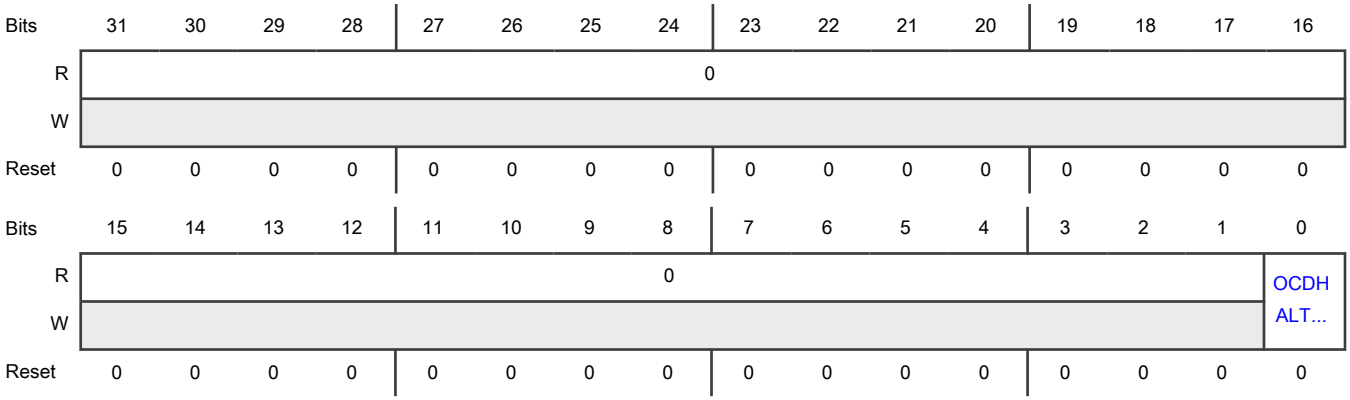
29.6.2.12 HiFi1 OCDHaltOnReset (OCDHALTONRESET)

Offset

Register	Offset
OCDHALTONRESET	304h

Function
HiFi1 enters OCD Halt mode if this signal is sampled asserted on reset.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 OCDHALTONRESET	OCDHaltOnReset 0b - Allow normal operation when HiFi1 exits reset (starts fetching and executing instructions from memory) 1b - Force HiFi1 to enter OCD Halt mode when HiFi1 exits reset

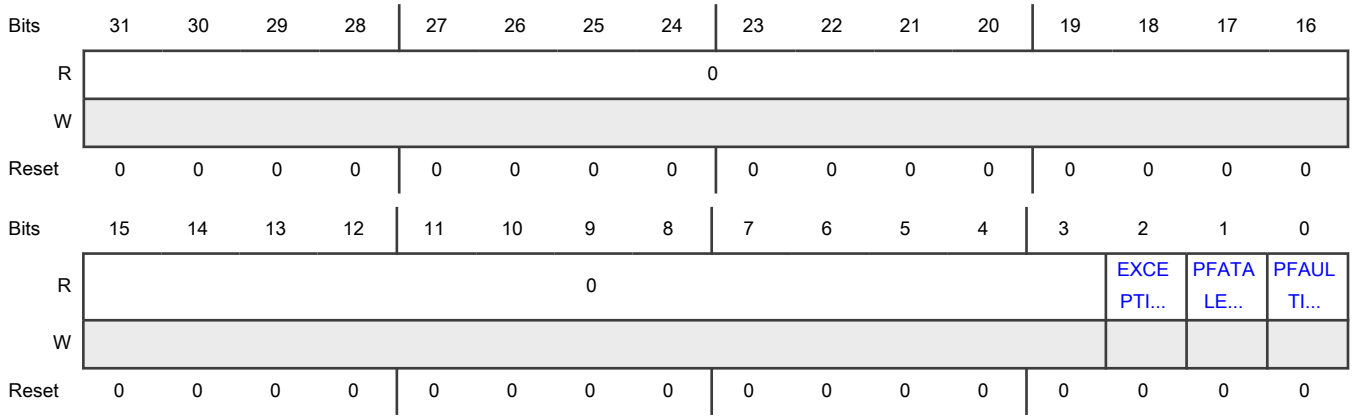
29.6.2.13 HiFi1 General Purpose Register 0 (HIFI1_GPR0)

Offset

Register	Offset
HIFI1_GPR0	308h

Function
General purpose register assigned to capture HiFi1 status signals.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 EXCEPTIONERR	DoubleExceptionError Single-cycle assertion every time double exception faults occur. Unlike [PFATALERR], this signal is not sticky. The system can use the information presented on [PFATALERR] and HIFI1_GPR1[PFAULTINFO] and decide whether to reset the HIFI1 core or the whole system. 0b - No error occurs 1b - Error occurs
1 PFATALERR	PFatalError Asserted when a fatal error condition occurs due to AXI parity or ECC errors. The PFatalError signal is sticky and is cleared by HiFi1 reset. 0b - No error occurs 1b - Error occurs
0 PFAULTINFOVLD	PFaultInfoValid Asserted when the PFaultInfo signal changes its value. The system can use its rising edge as an enabler to capture PFaultInfo. 0b - Not asserted 1b - Asserted

29.6.2.14 HiFi1 General Purpose Register 1 (HIFI1_GPR1)

Offset

Register	Offset
HIFI1_GPR1	314h

Function

General purpose register assigned to capture HiFi1 status signals.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PFAULTINFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PFAULTINFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PFAULTINFO	PFaultInfo This signal mirrors HiFi1's internal Fault Information Register and provides the source and severity of the fault. For detailed information, see "PFaultInfo and Fault Information Register (FIR)" section in HiFi1 chapter.

29.6.2.15 HiFi1 General Purpose Register 2 (HIFI1_GPR2)

Offset

Register	Offset
HIFI1_GPR2	318h

Function

General purpose register assigned to capture HiFi1 status signals.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIE_EXPSTATE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIE_EXPSTATE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TIE_EXPSTATE	GPIO32 Option TIE Output State Indicates the state of “tie_expstate” used to capture OSTIMER and to trigger EZH-V.

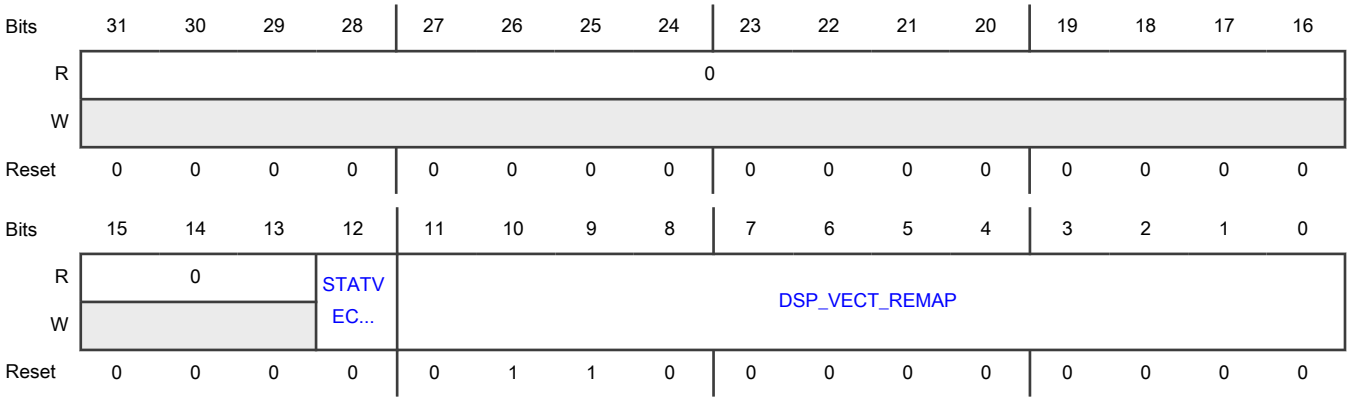
29.6.2.16 HiFi1 DSP Vector Remap (DSP_VECT_REMAP)

Offset

Register	Offset
DSP_VECT_REMAP	31Ch

Function
Configures HiFi1 DSP static vector remapping to shared SRAMs.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 STATVECSELECT	Static Vector Select Selects the static vector address on HiFi1. 0b - Selects the primary static vector address (0058_0000h) 1b - No effect
11-0	DSP Vector Remap

Table continues on the next page...

Table continued from the previous page...

Field	Function
DSP_VECT_REMAP	Remaps ARADDR[21:10] when ARADDR hits 1 KB from [STATVECSELECT] selected base address. **NOTE** Remapping the vectors will leave a "hole" of 1 KB at the default vector location which the HiFi1 DSP will never be able to access (CPU1 and all other initiators will have access though). **RESTRICTION** When [STATVECSELECT] = 0, bit 11 and bit 10 of this field will not take effect.

29.6.2.17 eDMA2 Request Enable 0 (EDMA2_EN0)

Offset

Register	Offset
EDMA2_EN0	420h

Function

Enables request register 0 for eDMA2.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LP_FLEX...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	Reserved		ACMP_EN	SDADC_F...	SDADC_F...	SDADC_F...	SDADC_F...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADC0_FI...	ADC0_FI...	CTIMER7...	CTIMER7...	CTIMER6...	CTIMER6...	CTIMER5...	CTIMER5...	PINT1_I...	PINT1_I...	PINT1_I...	PINT1_I...	XSPI2_T...	XSPI2_R...	MICFIL_...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LP_FLEXCOMM17_RX_EN	LP_FLEXCOMM17 RX FIFO Drain Enable 0b - Disables 1b - Enables
30	FLEXIO Shift Timer 7 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
FLEXIO_SHFT_TIMER7_EN	0b - Disables 1b - Enables
29 FLEXIO_SHFT_TIMER6_EN	FLEXIO Shift Timer 6 Enable 0b - Disables 1b - Enables
28 FLEXIO_SHFT_TIMER5_EN	FLEXIO Shift Timer 5 Enable 0b - Disables 1b - Enables
27 FLEXIO_SHFT_TIMER4_EN	FLEXIO Shift Timer 4 Enable 0b - Disables 1b - Enables
26 FLEXIO_SHFT_TIMER3_EN	FLEXIO Shift Timer 3 Enable 0b - Disables 1b - Enables
25 FLEXIO_SHFT_TIMER2_EN	FLEXIO Shift Timer 2 Enable 0b - Disables 1b - Enables
24 FLEXIO_SHFT_TIMER1_EN	FLEXIO Shift Timer 1 Enable 0b - Disables 1b - Enables
23 FLEXIO_SHFT_TIMER0_EN	FLEXIO Shift Timer 0 Enable 0b - Disables 1b - Enables
22-21 —	Reserved
20 ACMP_EN	ACMP0 Enable 0b - Disables 1b - Enables
19 SDADC_FIFO3_EN	SDADC FIFO3 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
18 SDADC_FIFO2_EN	SDADC FIFO2 Enable 0b - Disables 1b - Enables
17 SDADC_FIFO1_EN	SDADC FIFO1 Enable 0b - Disables 1b - Enables
16 SDADC_FIFO0_EN	SDADC FIFO0 Enable 0b - Disables 1b - Enables
15 ADC0_FIFO1_EN	ADC0 FIFO1 Enable 0b - Disables 1b - Enables
14 ADC0_FIFO0_EN	ADC0 FIFO0 Enable 0b - Disables 1b - Enables
13 CTIMER7_M1_EN	CTIMER7 M1 Enable 0b - Disables 1b - Enables
12 CTIMER7_M0_EN	CTIMER7 M0 Enable 0b - Disables 1b - Enables
11 CTIMER6_M1_EN	CTIMER6 M1 Enable 0b - Disables 1b - Enables
10 CTIMER6_M0_EN	CTIMER6 M0 Enable 0b - Disables 1b - Enables
9 CTIMER5_M1_EN	CTIMER5 M1 Enable 0b - Disables 1b - Enables
8	CTIMER5 M0 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER5_M0_EN	0b - Disables 1b - Enables
7 PINT1_IRQ3_EN	PINT1 Interrupt 3 Enable 0b - Disables 1b - Enables
6 PINT1_IRQ2_EN	PINT1 Interrupt 2 Enable 0b - Disables 1b - Enables
5 PINT1_IRQ1_EN	PINT1 Interrupt 1 Enable 0b - Disables 1b - Enables
4 PINT1_IRQ0_EN	PINT1 Interrupt 0 Enable 0b - Disables 1b - Enables
3 XSPI2_TX_EN	XSPI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
2 XSPI2_RX_EN	XSPI2 RX FIFO Drain Enable 0b - Disables 1b - Enables
1 MICFIL_EN	MICFIL Enable 0b - Disables 1b - Enables
0 —	Reserved

29.6.2.18 eDMA2 Request Enable 1 (EDMA2_EN1)

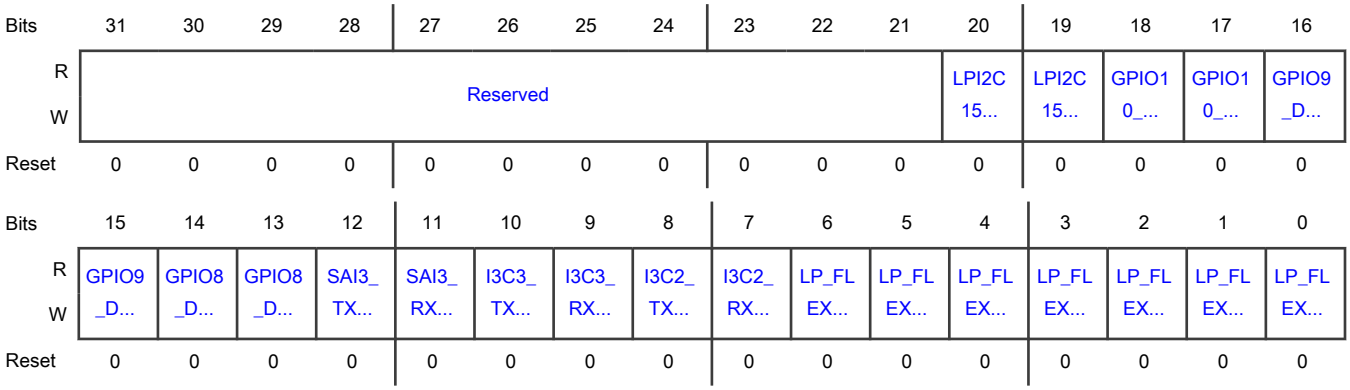
Offset

Register	Offset
EDMA2_EN1	424h

Function

Enables request register 1 for eDMA2.

Diagram



Fields

Field	Function
31-21 —	Reserved
20 LPI2C15_TX_EN	LPI2C15 TX FIFO Fill Enable 0b - Disables 1b - Enables
19 LPI2C15_RX_EN	LPI2C15 RX FIFO Drain Enable 0b - Disables 1b - Enables
18 GPIO10_DMA1_EN	GPIO10 eDMA Request 1 Enable 0b - Disables 1b - Enables
17 GPIO10_DMA0_EN	GPIO10 eDMA Request 0 Enable 0b - Disables 1b - Enables
16 GPIO9_DMA1_EN	GPIO9 eDMA Request 1 Enable 0b - Disables 1b - Enables
15 GPIO9_DMA0_EN	GPIO9 eDMA Request 0 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
14 GPIO8_DMA1_EN	GPIO8 eDMA Request 1 Enable 0b - Disables 1b - Enables
13 GPIO8_DMA0_EN	GPIO8 eDMA Request 0 Enable 0b - Disables 1b - Enables
12 SAI3_TX_EN	SAI3 TX FIFO Fill Enable 0b - Disables 1b - Enables
11 SAI3_RX_EN	SAI3 RX FIFO Drain Enable 0b - Disables 1b - Enables
10 I3C3_TX_EN	I3C3 TX FIFO Fill Enable 0b - Disables 1b - Enables
9 I3C3_RX_EN	I3C3 RX FIFO Drain Enable 0b - Disables 1b - Enables
8 I3C2_TX_EN	I3C2 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 I3C2_RX_EN	I3C2 RX FIFO Drain Enable 0b - Disables 1b - Enables
6 LP_FLEXCOMM20_TX_EN	LP_FLEXCOMM20 TX FIFO Fill Enable 0b - Disables 1b - Enables
5 LP_FLEXCOMM20_RX_EN	LP_FLEXCOMM20 RX FIFO Drain Enable 0b - Disables 1b - Enables
4	LP_FLEXCOMM19 TX FIFO Fill Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM19_TX_EN	0b - Disables 1b - Enables
3 LP_FLEXCOMM19_RX_EN	LP_FLEXCOMM19 RX FIFO Drain Enable 0b - Disables 1b - Enables
2 LP_FLEXCOMM18_TX_EN	LP_FLEXCOMM18 TX FIFO Fill Enable 0b - Disables 1b - Enables
1 LP_FLEXCOMM18_RX_EN	LP_FLEXCOMM18 RX FIFO Drain Enable 0b - Disables 1b - Enables
0 LP_FLEXCOMM17_TX_EN	LP_FLEXCOMM17 TX FIFO Fill Enable 0b - Disables 1b - Enables

29.6.2.19 eDMA3 Request Enable 0 (EDMA3_EN0)

Offset

Register	Offset
EDMA3_EN0	430h

Function

Enables request register 0 for eDMA3.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LP_FLEX...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	FLEXIO_...	Reserved		ACMP_EN	SDADC_F...	SDADC_F...	SDADC_F...	SDADC_F...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADC0_FI...	ADC0_FI...	CTIMER7...	CTIMER7...	CTIMER6...	CTIMER6...	CTIMER5...	CTIMER5...	PINT1_I...	PINT1_I...	PINT1_I...	PINT1_I...	XSPI2_T...	XSPI2_R...	MICFIL_...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LP_FLEXCOMM17_RX_EN	LP_FLEXCOMM17 RX FIFO Drain Enable 0b - Disables 1b - Enables
30 FLEXIO_SHFT_TIMER7_EN	FLEXIO Shift Timer 7 Enable 0b - Disables 1b - Enables
29 FLEXIO_SHFT_TIMER6_EN	FLEXIO Shift Timer 6 Enable 0b - Disables 1b - Enables
28 FLEXIO_SHFT_TIMER5_EN	FLEXIO Shift Timer 5 Enable 0b - Disables 1b - Enables
27 FLEXIO_SHFT_TIMER4_EN	FLEXIO Shift Timer 4 Enable 0b - Disables 1b - Enables
26 FLEXIO_SHFT_TIMER3_EN	FLEXIO Shift Timer 3 Enable 0b - Disables 1b - Enables
25 FLEXIO_SHFT_TIMER2_EN	FLEXIO Shift Timer 2 Enable 0b - Disables 1b - Enables
24 FLEXIO_SHFT_TIMER1_EN	FLEXIO Shift Timer 1 Enable 0b - Disables 1b - Enables
23 FLEXIO_SHFT_TIMER0_EN	FLEXIO Shift Timer 0 Enable 0b - Disables 1b - Enables
22-21 —	Reserved
20 ACMP_EN	ACMP0 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
19 SDADC_FIFO3_EN	SDADC FIFO3 Enable 0b - Disables 1b - Enables
18 SDADC_FIFO2_EN	SDADC FIFO2 Enable 0b - Disables 1b - Enables
17 SDADC_FIFO1_EN	SDADC FIFO1 Enable 0b - Disables 1b - Enables
16 SDADC_FIFO0_EN	SDADC FIFO0 Enable 0b - Disables 1b - Enables
15 ADC0_FIFO1_EN	ADC0 FIFO1 Enable 0b - Disables 1b - Enables
14 ADC0_FIFO0_EN	ADC0 FIFO0 Enable 0b - Disables 1b - Enables
13 CTIMER7_M1_EN	CTIMER7 M1 Enable 0b - Disables 1b - Enables
12 CTIMER7_M0_EN	CTIMER7 M0 Enable 0b - Disables 1b - Enables
11 CTIMER6_M1_EN	CTIMER6 M1 Enable 0b - Disables 1b - Enables
10 CTIMER6_M0_EN	CTIMER6 M0 Enable 0b - Disables 1b - Enables
9	CTIMER5 M1 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER5_M1_EN	0b - Disables 1b - Enables
8 CTIMER5_M0_EN	CTIMER5 M0 Enable 0b - Disables 1b - Enables
7 PINT1_IRQ3_EN	PINT1 Interrupt 3 Enable 0b - Disables 1b - Enables
6 PINT1_IRQ2_EN	PINT1 Interrupt 2 Enable 0b - Disables 1b - Enables
5 PINT1_IRQ1_EN	PINT1 Interrupt 1 Enable 0b - Disables 1b - Enables
4 PINT1_IRQ0_EN	PINT1 Interrupt 0 Enable 0b - Disables 1b - Enables
3 XSPI2_TX_EN	XSPI2 TX FIFO Fill Enable 0b - Disables 1b - Enables
2 XSPI2_RX_EN	XSPI2 RX FIFO Drain Enable 0b - Disables 1b - Enables
1 MICFIL_EN	MICFIL Enable 0b - Disables 1b - Enables
0 —	Reserved

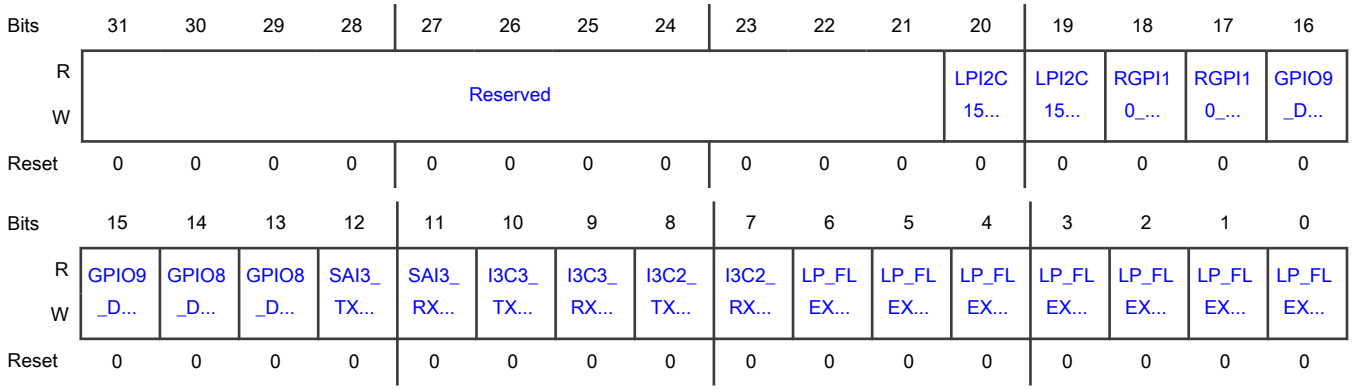
29.6.2.20 eDMA3 Request Enable 1 (EDMA3_EN1)

Offset

Register	Offset
EDMA3_EN1	434h

Function
Enables request register 1 for eDMA3.

Diagram



Fields

Field	Function
31-21 —	Reserved
20 LPI2C15_TX_EN	LPI2C15 TX FIFO Fill Enable 0b - Disables 1b - Enables
19 LPI2C15_RX_EN	LPI2C15 RX FIFO Drain Enable 0b - Disables 1b - Enables
18 RGPI10_DMA1_EN	GPIO10 eDMA Request 1 Enable 0b - Disables 1b - Enables
17 RGPI10_DMA0_EN	GPIO10 eDMA Request 0 Enable 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 GPIO9_DMA1_EN	GPIO9 eDMA Request 1 Enable 0b - Disables 1b - Enables
15 GPIO9_DMA0_EN	GPIO9 eDMA Request 0 Enable 0b - Disables 1b - Enables
14 GPIO8_DMA1_EN	GPIO8 eDMA Request 1 Enable 0b - Disables 1b - Enables
13 GPIO8_DMA0_EN	GPIO8 eDMA Request 0 Enable 0b - Disables 1b - Enables
12 SAI3_TX_EN	SAI3 TX FIFO Fill Enable 0b - Disables 1b - Enables
11 SAI3_RX_EN	SAI3 RX FIFO Drain Enable 0b - Disables 1b - Enables
10 I3C3_TX_EN	I3C3 TX FIFO Fill Enable 0b - Disables 1b - Enables
9 I3C3_RX_EN	I3C3 RX FIFO Drain Enable 0b - Disables 1b - Enables
8 I3C2_TX_EN	I3C2 TX FIFO Fill Enable 0b - Disables 1b - Enables
7 I3C2_RX_EN	I3C2 RX FIFO Drain Enable 0b - Disables 1b - Enables
6	LP_FLEXCOMM20 TX FIFO Fill Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM20_TX_EN	0b - Disables 1b - Enables
5 LP_FLEXCOMM20_RX_EN	LP_FLEXCOMM20 RX FIFO Drain Enable 0b - Disables 1b - Enables
4 LP_FLEXCOMM19_TX_EN	LP_FLEXCOMM19 TX FIFO Fill Enable 0b - Disables 1b - Enables
3 LP_FLEXCOMM19_RX_EN	LP_FLEXCOMM19 RX FIFO Drain Enable 0b - Disables 1b - Enables
2 LP_FLEXCOMM18_TX_EN	LP_FLEXCOMM18 TX FIFO Fill Enable 0b - Disables 1b - Enables
1 LP_FLEXCOMM18_RX_EN	LP_FLEXCOMM18 RX FIFO Drain Enable 0b - Disables 1b - Enables
0 LP_FLEXCOMM17_TX_EN	LP_FLEXCOMM17 TX FIFO Fill Enable 0b - Disables 1b - Enables

29.6.2.21 I3C Asynchronous Wake-up Control (I3C_ASYNC_WAKEUP_CTRL)

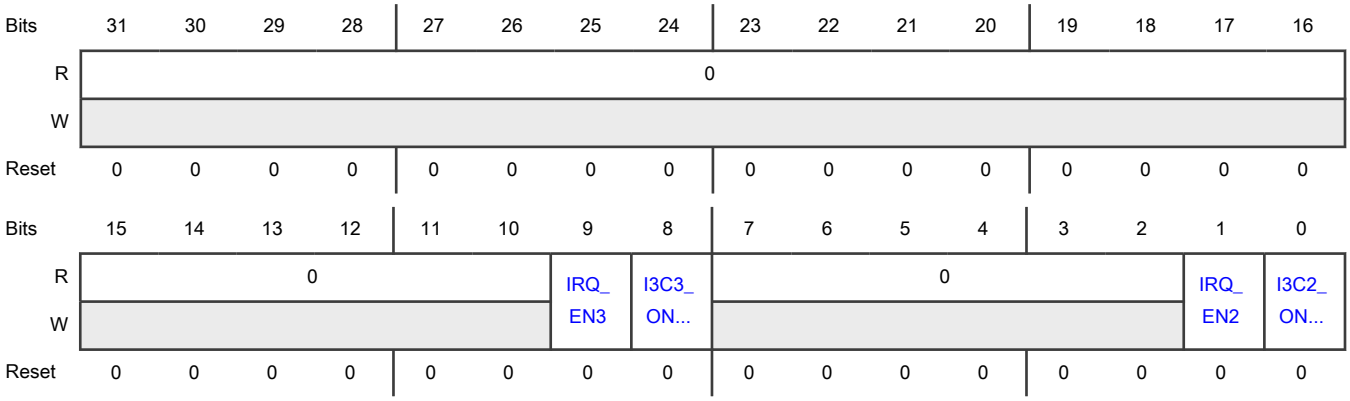
Offset

Register	Offset
I3C_ASYNC_WAKEUP_CTRL	628h

Function

Controls Improved Inter-Integrated Circuit 2-3 (I3C2 and I3C3) asynchronous wake-up for I3C2 and I3C3 in VDD1_SENSE domain.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 IRQ_EN3	I3C3 Controller Mode Asynchronous Wake-up Interrupt Enable If enabled, the interrupt happens when I3C3 is in controller mode, FCLK is stopped and SDA is driven low by target. 0b - Interrupt disabled 1b - Interrupt enabled
8 I3C3_ON_CHIP_STRONG_PULL_DIS	I3C3 On-Chip Strong Pull Disable 0b - Enables 1b - Disables
7-2 —	Reserved
1 IRQ_EN2	I3C2 Controller Mode Asynchronous Wake-up Interrupt Enable If enabled, the interrupt happens when I3C2 is in controller mode, FCLK is stopped and SDA is driven low by target. 0b - Interrupt disabled 1b - Interrupt enabled
0 I3C2_ON_CHIP_STRONG_PULL_DIS	I3C2 On-Chip Strong Pull Disable 0b - Enables 1b - Disables

29.6.3 SYSCON2 register descriptions

29.6.3.1 SYSCON2 memory map

SYSCON2 base address: 4006_6000h

Offset	Register	Width (In bits)	Access	Reset value
FE0h - FE4h	General Purpose Register n (GPR0 - GPR1)	32	RW	0000_0000h

29.6.3.2 General Purpose Register n (GPR0 - GPR1)

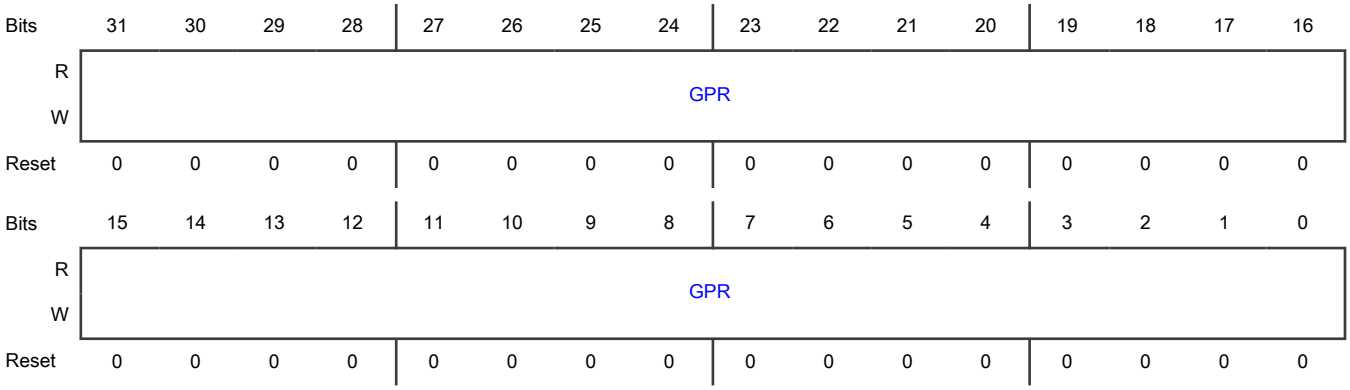
Offset

Register	Offset
GPR0	FE0h
GPR1	FE4h

Function

Available for the software to store data. The register content is retained when the chip enters Full Deep Sleep Retention (FDSR) mode.

Diagram



Fields

Field	Function
31-0 GPR	General Purpose Register Bits

29.6.4 SYSCON3 register descriptions

29.6.4.1 SYSCON3 memory map

SYSCON3 base address: 4006_2000h

Offset	Register	Width (In bits)	Access	Reset value
80h	Receive Event Pulse Generator (RXEVPULSEGEN)	32	W	0000_0000h
8Ch	CPU Status (CPU_STATUS)	32	RW	0000_0001h
98h	CPU1 Secure Vector Table Offset (CPU1_SVTOR)	32	RW	0020_B000h
9Ch	CPU1 Non-Secure Vector Table Offset (CPU1_NSVTOR)	32	RW	0000_B000h
C4h	Silicon Revision ID (SILICONREV_ID)	32	R	000B_0000h
114h	Automatic Clock Gate Override (AUTOCLKGATEOVERRIDE0)	32	RW	0000_0FFFh
118h	SRAM Clock Gating Control (SRAM_CLKGATE_CTRL)	32	RW	0000_0FFFh
280h	VDD1_SENSE Auto Gating Enable (SENSE_AUTOGATE_EN)	32	RW	0000_0000h
514h	TEMPDETECT0 Control (TEMPDETECT0_CTRL)	32	RW	0000_0000h
518h	TEMPDETECT1 Control (TEMPDETECT1_CTRL)	32	RW	0000_0000h
51Ch	TEMPDETECT Interrupts Output (TEMPDETECT_FLAGS)	32	R	0000_0000h
600h	TEMPDETECT Interrupts Enable (TEMPDETECT_INT_EN)	32	RW	0000_0000h
B60h	Gray to Binary Converter - Gray Code [31:0] (GRAY_CODE_LSB)	32	RW	0000_0000h
B64h	Gray to Binary Converter - Gray Code [63:32] (GRAY_CODE_MSB)	32	RW	0000_0000h
B68h	Gray to Binary Converter - Binary Code [31:0] (BINARY_CODE_LSB)	32	R	0000_0000h
B6Ch	Gray to Binary Converter - Binary Code [63:32] (BINARY_CODE_MSB)	32	R	0000_0000h
FE0h - FFCh	Data Register n (DATA_REG0 - DATA_REG7)	32	R	See section

29.6.4.2 Receive Event Pulse Generator (RXEVPULSEGEN)

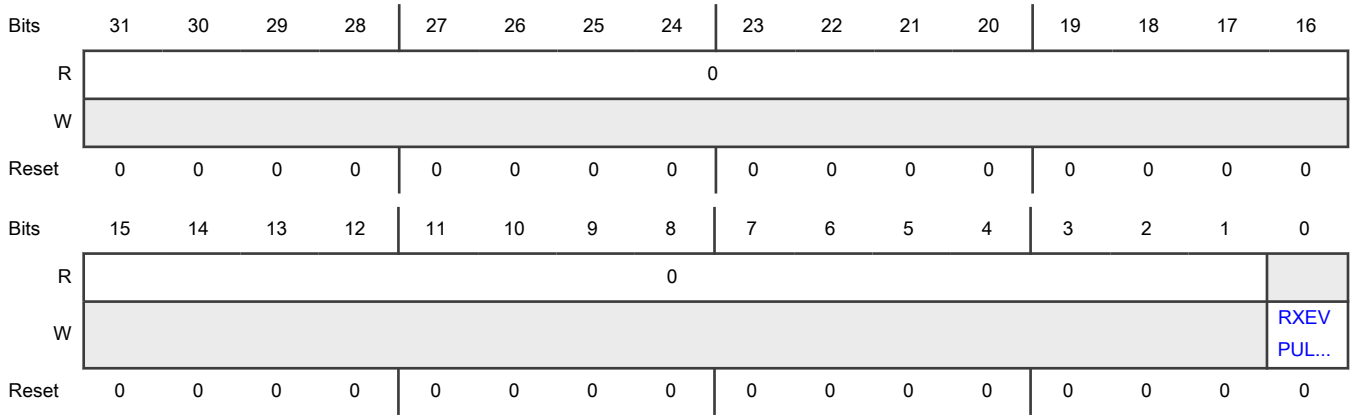
Offset

Register	Offset
RXEVPULSEGEN	80h

Function

Allows generating a pulse on CPU1 receive event input that will wake it up if it is stopped in a wait for event (WFE) instruction.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 RXEVPULSEGEN	Receive Event Pulse Generator Writing a 1 to this register will create a logic 1 pulse for one APB bus clock when set by HiFi1 and HiFi4. It is automatically cleared. 0b - No effect 1b - Pulse receive event high for one cycle

29.6.4.3 CPU Status (CPU_STATUS)

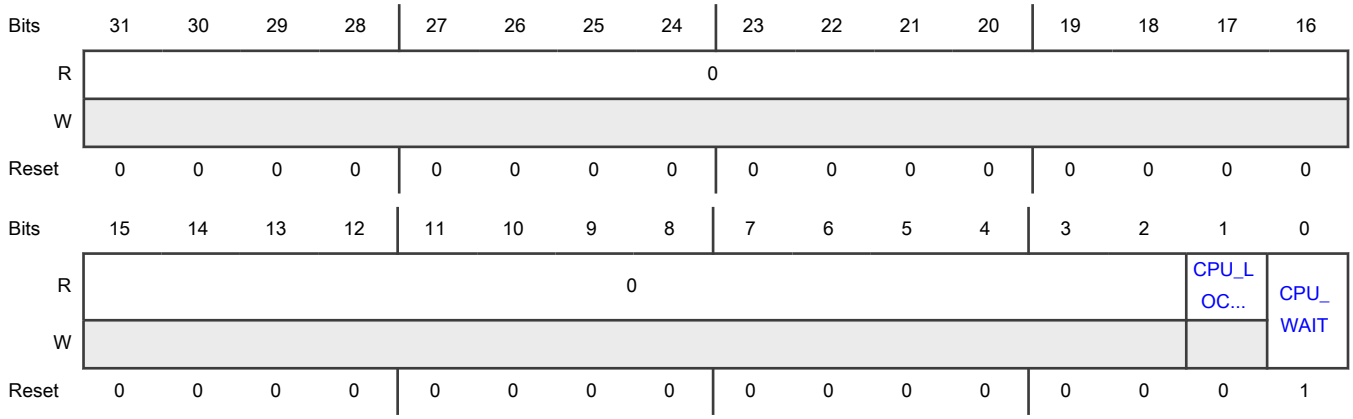
Offset

Register	Offset
CPU_STATUS	8Ch

Function

Indicates CPU1 status.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 CPU_LOCKUP	CPU1 Lockup State Indicates whether CPU1 is in lockup state. 0b - Not in lockup 1b - In lockup
0 CPU_WAIT	CPU1 Stall Stalls CPU1 out of reset. 0b - No stall 1b - Stall

29.6.4.4 CPU1 Secure Vector Table Offset (CPU1_SVTOR)

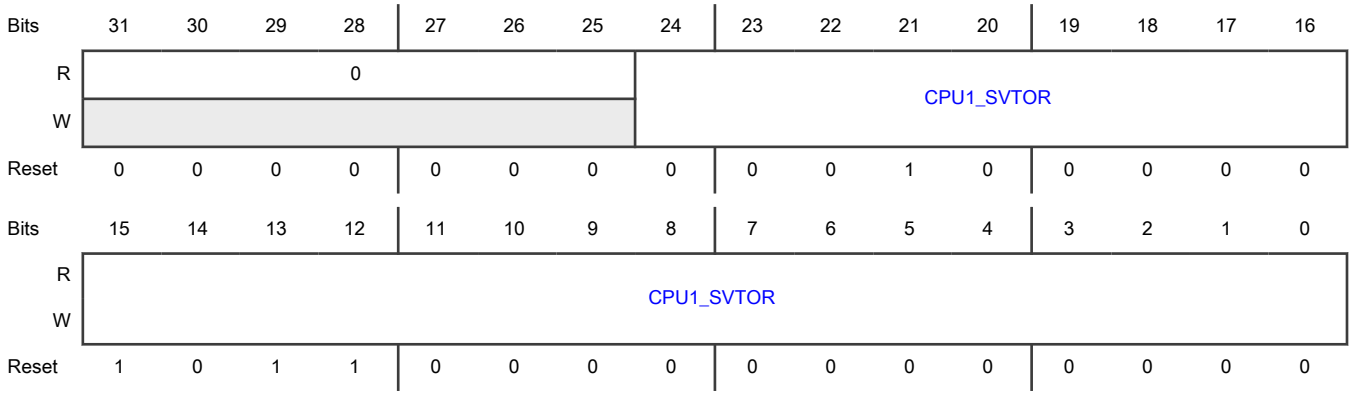
Offset

Register	Offset
CPU1_SVTOR	98h

Function

Indicates the offset of the secure vector table base address from the shared memory address.

Diagram



Fields

Field	Function
31-25 —	Reserved
24-0 CPU1_SVTOR	Secure Vector Table Offset Specifies the secure vector table offset register out of reset. This register controls INIT_SVTOR[31:7] of CPU1. For example, if secure vector is 32'h1058_0000, the register value will be 25'h020B000.

29.6.4.5 CPU1 Non-Secure Vector Table Offset (CPU1_NSVTOR)

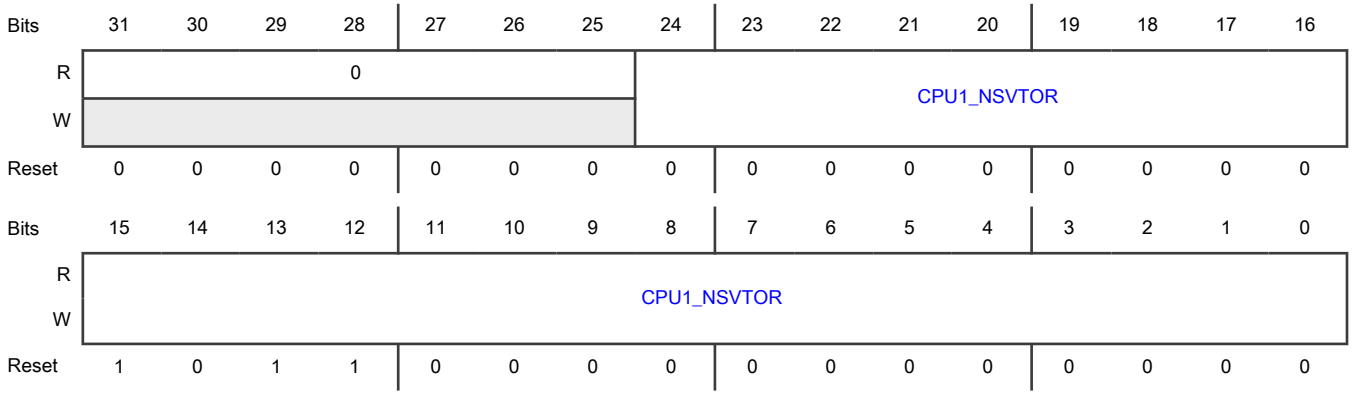
Offset

Register	Offset
CPU1_NSVTOR	9Ch

Function

Indicates the offset of the non-secure vector table base address from memory address.

Diagram



Fields

Field	Function
31-25 —	Reserved
24-0 CPU1_NSVTOR	Non-Secure Vector Table Offset Specifies the non-secure vector table offset register out of reset. This register controls INIT_NSVTOR[31:7] of CPU1.

29.6.4.6 Silicon Revision ID (SILICONREV_ID)

Offset

Register	Offset
SILICONREV_ID	C4h

Function

On the first silicon, the minor revision is 0h. Any subsequent metal mask change or update should result in the minor revision being upgraded to the next level.

On the first silicon, the major revision is Ah. Any subsequent full mask change or update should result in the major revision being upgraded to the next level.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												MAJOR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												MINOR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
19-16 MAJOR	Silicon Revision Major Tag Namely: A, B, C, etc.
15-4 —	Reserved
3-0 MINOR	Silicon Revision Minor Tag Namely: 0, 1, 2, 3, etc.

29.6.4.7 Automatic Clock Gate Override (AUTOCLKGATEOVERRIDE0)

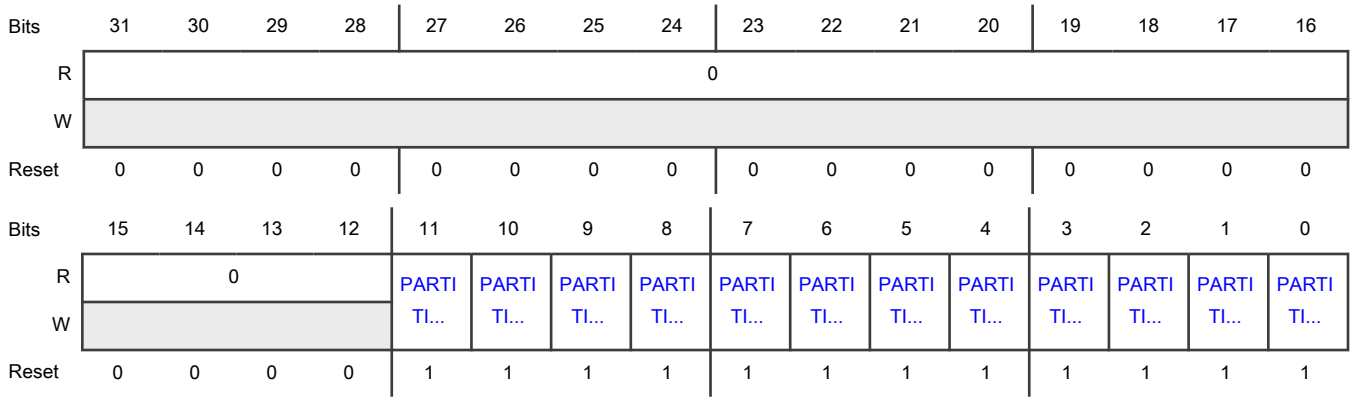
Offset

Register	Offset
AUTOCLKGATEOVERRIDE0	114h

Function

This register allows selectively disabling automatic clock gating for SRAM partitions. By default, automatic clock gating turns off clocks to each internal SRAM after 5 bus clocks with no activity. This saves power when the SRAMs are not used for a period of time. When turned off due to automatic clock gating, there is a 1- clock delay for the next access to an SRAM.

Diagram



Fields

Field	Function
31-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-0 PARTITIONn	Partition n Controls clock gating for Partition 18 (bit 0) to Partition 29 (bit 11). 0b - Enables clock gating 1b - Continuous clocking

29.6.4.8 SRAM Clock Gating Control (SRAM_CLKGATE_CTRL)

Offset

Register	Offset
SRAM_CLKGATE_CTRL	118h

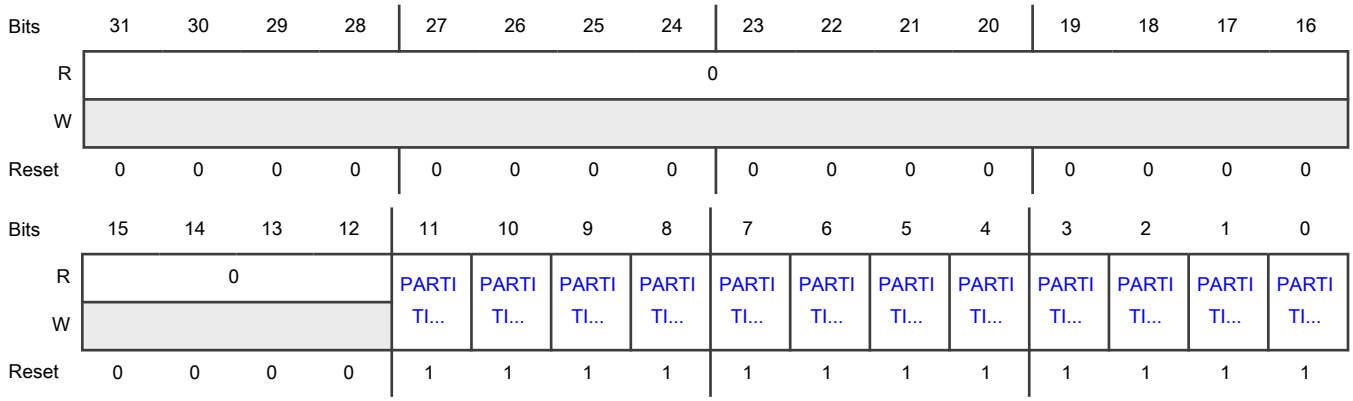
Function

If cleared, the corresponding SRAM partition input clock driven by SENSE_MAIN_CLK will be gated off. This provides means to save power if the corresponding partition is not planned to be used for a significant amount of time. To get the power saving, the Sense domain CPU (CPU1) needs to do a dummy read to the partition to switch the partition clock to SENSE_MAIN_CLK, and then clear this bit. While the bit for an associated partition is cleared, no bus initiators will be able to access that partition unless the bit is set again.

NOTE

When gated, any core will halt when accessing the partition, until CPU1 turns on the partition clock.

Diagram



Fields

Field	Function
31-12	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
11-0 PARTITIONn	Partition n Gates off the bus clock input in VDD1_SENSE domain for Partition 18 (bit 0) to Partition 29 (bit 11). 0b - Bus clock input gated 1b - No effects

29.6.4.9 VDD1_SENSE Auto Gating Enable (SENSE_AUTOGATE_EN)

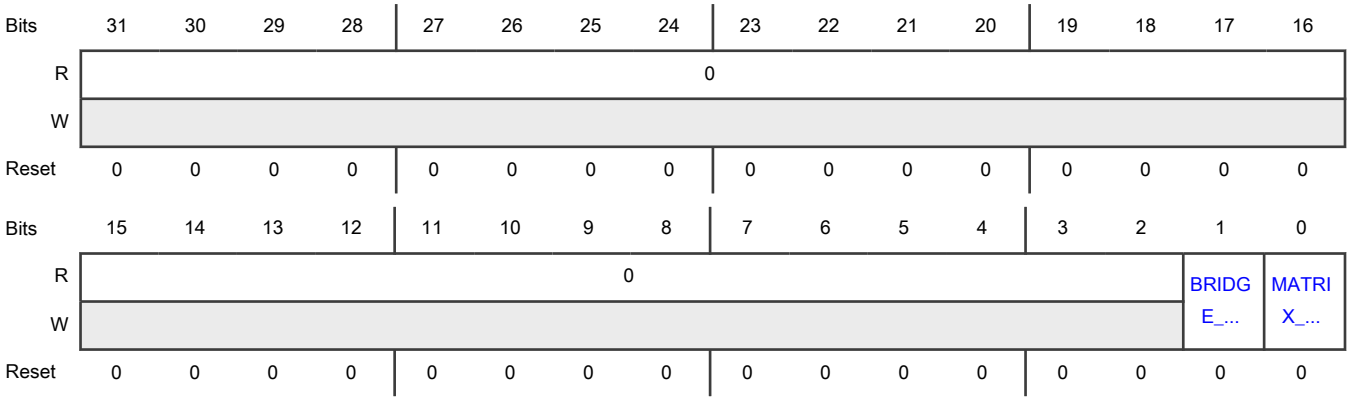
Offset

Register	Offset
SENSE_AUTOGATE_EN	280h

Function

Controls clock gating for S-bus, AHB2APB bridge, and the AIPS and AXBS in VDD1_SENSE domain. This register allows selectively enabling automatic clock gating for the modules. By default, automatic clock gating turns off clocks to the modules after 2 bus clocks with no bus transaction activity. This saves power when the corresponding modules are not used for a period of time.

Diagram



Fields

Field	Function
31-2 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 BRIDGE_EN	Automatic Clock Gating Enable for AHB to APB Bridge, AIPS and AXBS Controls clock gating for AHB2APB bridge, and the AIPS and AXBS in VDD1_SENSE domain. You can set it to disable the module's clock in order to save power when the module is not working. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating
0 MATRIX_EN	Automatic Clock Gating Enable for S-Bus Controls clock gating for S-bus. You can set it to disable the module's clock in order to save power when the module is not working. **NOTE** Once CPU1 enters Deep Sleep when accessing RAM arbiter 1, if any other initiator needs to access RAM arbiter 1, you must disable this bit before CPU1 enters Deep Sleep. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating

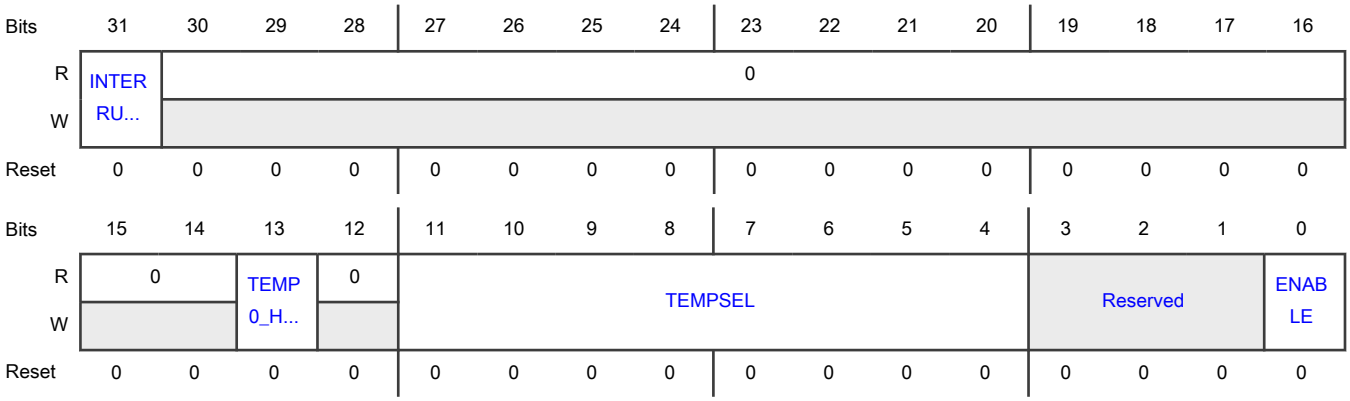
29.6.4.10 TEMPDETECT0 Control (TEMPDETECT0_CTRL)

Offset

Register	Offset
TEMPDETECT0_CTRL	514h

Function
Controls TEMPDETECT0.

Diagram



Fields

Field	Function
31 INTERRUPT_CLEAR	TEMPDETECT0 Interrupt Clear Writing 1 to this register will clear the corresponding bit in TEMPDETECT_FLAGS. 0b - Not clear 1b - Clears
30-14 —	Reserved
13 TEMP0_HIGH_DETECT	TEMPDETECT0 High Or Low Detection Controls if TEMPDECT0 will trigger on a temperature above or below the set threshold. 0b - Below 1b - Above
12 —	Reserved
11-4 TEMPSEL	TEMPDETECT0 Temperature Select For value settings, see Temperature threshold. 0000_0000b-1111_1111b - See Table 248.
3-1 —	Reserved
0 ENABLE	TEMPDETECT0 Enable 0b - Disables 1b - Enables

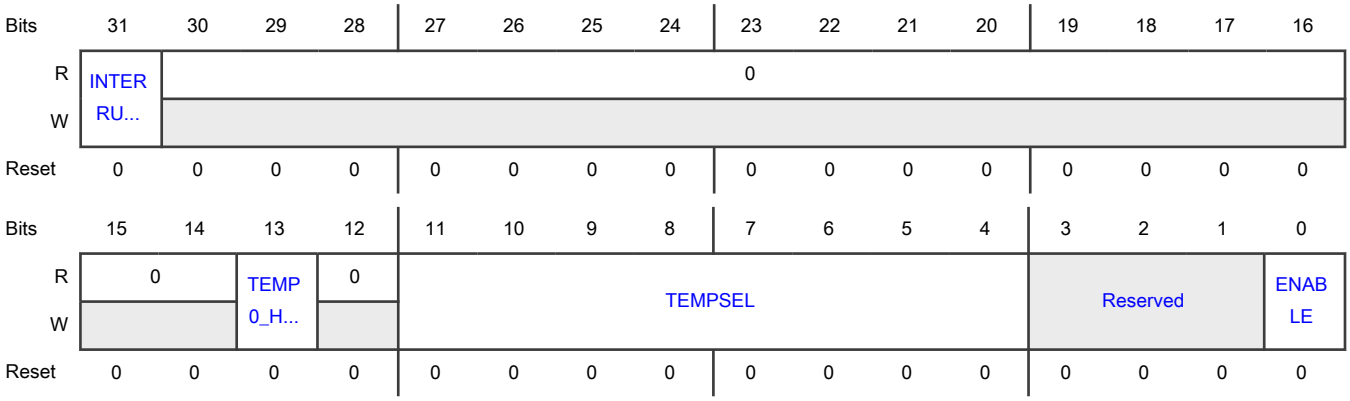
29.6.4.11 TEMPDETECT1 Control (TEMPDETECT1_CTRL)

Offset

Register	Offset
TEMPDETECT1_CTRL	518h

Function
Controls TEMPDETECT1.

Diagram



Fields

Field	Function
31 INTERRUPT_CLEAR	TEMPDETECT1 Interrupt Clear Writing 1 to this register will clear the corresponding bit in TEMPDETECT_FLAGS. 0b - Not clear 1b - Clears
30-14 —	Reserved
13 TEMP0_HIGH_DETECT	TEMPDETECT1 High Or Low Detection Controls if TEMPDECT1 will trigger on a temperature above or below the set threshold. 0b - Below 1b - Above
12 —	Reserved
11-4 TEMPSEL	TEMPDETECT1 Temperature Select For value settings, see Temperature threshold. 0000_0000b-1111_1111b - See Table 248.
3-1 —	Reserved
0 ENABLE	TEMPDETECT1 Enable 0b - Disables 1b - Enables

29.6.4.12 TEMPDETECT Interrupts Output (TEMPDETECT_FLAGS)

Offset

Register	Offset
TEMPDETECT_FLAGS	51Ch

Function

Indicates interrupts for TEMPDETECT0 and TEMPDETECT1. Use TEMPDETECTn_CTRL[INTERRUPT_CLEAR] to clear these interrupt flags.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														TEMPDET...	TEMPDET...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1 TEMPDETECT1	TEMPDETECT1 Interrupt 0b - No Interrupt 1b - Interrupt
0 TEMPDETECT0	TEMPDETECT0 Interrupt 0b - No Interrupt 1b - Interrupt

29.6.4.13 TEMPDETECT Interrupts Enable (TEMPDETECT_INT_EN)

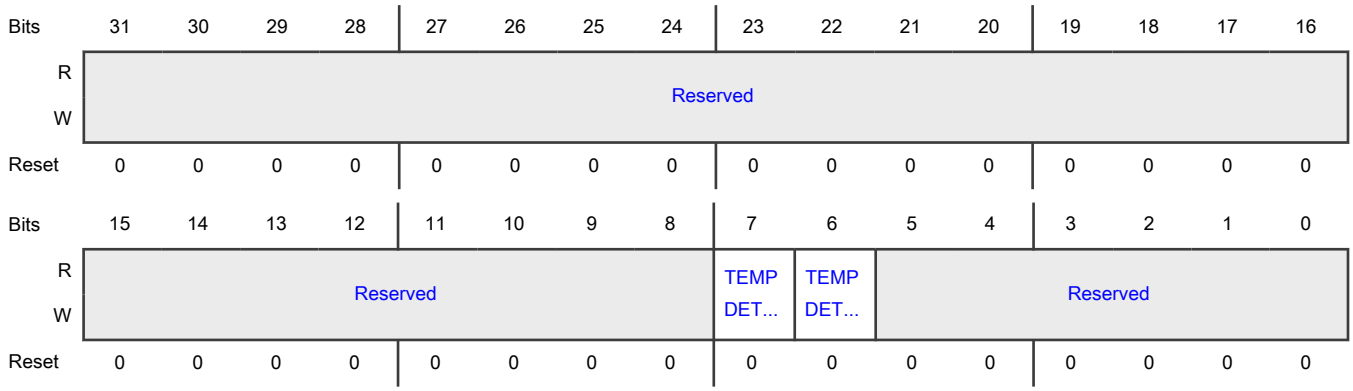
Offset

Register	Offset
TEMPDETECT_INT_EN	600h

Function

When TEMPDETECT is enabled, the overflag output signal of the analog module is pulled up immediately because of the strong PMOS, and it takes 200 µs for the final valid output to be settled. Therefore, software must set the interrupt enable bits until 200 µs after TEMPDETECT is enabled.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 TEMPDETECT1_INT_EN	TEMPDETECT1 Interrupt Enable **NOTE** When TEMPDETECT1 is active and you wish to alter the temperature threshold selection, you should first disable this bit before making any changes to the threshold. After changing the threshold, wait for 200 µs before re-enabling this bit. 0b - Disables 1b - Enables
6 TEMPDETECT0_INT_EN	TEMPDETECT0 Interrupt Enable **NOTE** When TEMPDETECT0 is active and you wish to alter the temperature threshold selection, you should first disable this bit before making any changes to the threshold. After changing the threshold, wait for 200 µs before re-enabling this bit. 0b - Disables 1b - Enables
5-0 —	Reserved

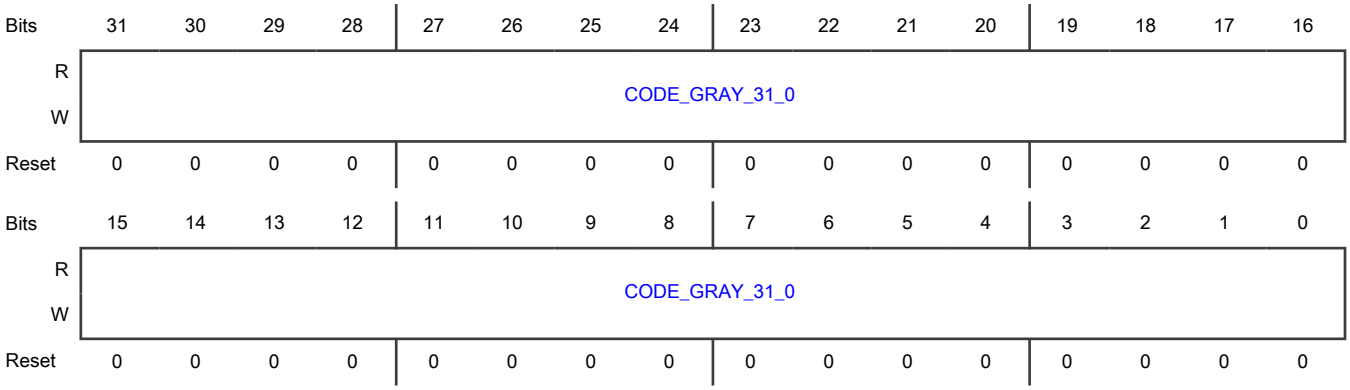
29.6.4.14 Gray to Binary Converter - Gray Code [31:0] (GRAY_CODE_LSB)

Offset

Register	Offset
GRAY_CODE_LSB	B60h

Function
Contains the least-significant portion of the Gray code to be converted back to binary. CPU1 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_GRAY_31_0	Gray Code [31:0] Indicates the least-significant 32 bits of the 64-bit Gray code to be converted.

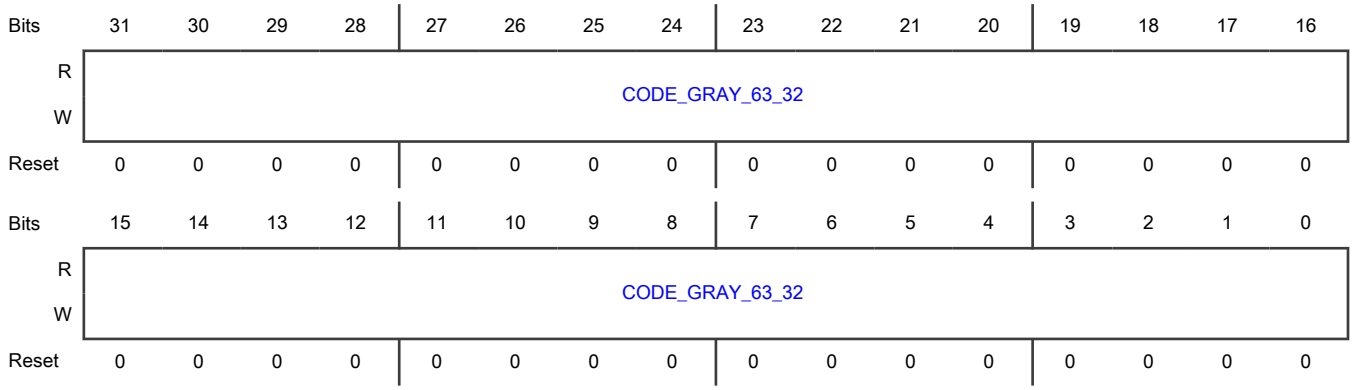
29.6.4.15 Gray to Binary Converter - Gray Code [63:32] (GRAY_CODE_MSB)

Offset

Register	Offset
GRAY_CODE_MSB	B64h

Function
Contains the most-significant portion of the Gray code to be converted back to binary. CPU1 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_GRAY_63_32	Gray Code [63:32] Indicates the most-significant 32 bits of the 64-bit Gray code to be converted.

29.6.4.16 Gray to Binary Converter - Binary Code [31:0] (BINARY_CODE_LSB)

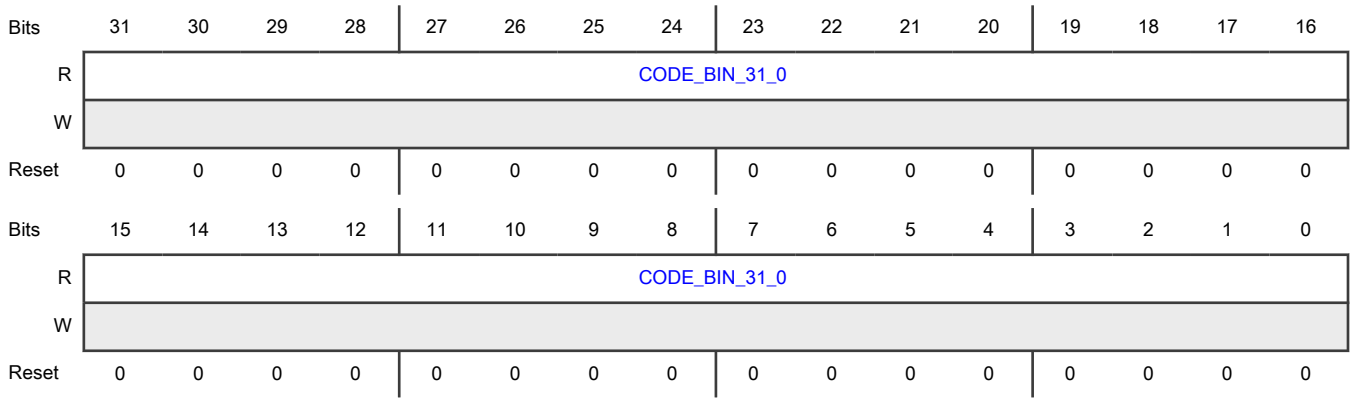
Offset

Register	Offset
BINARY_CODE_LSB	B68h

Function

Contains the least-significant portion of the code converted from Gray to binary coding. CPU1 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_BIN_31_0	Binary Code [31:0] Indicates the least-significant 32 bits of the 64-bit converted code.

29.6.4.17 Gray to Binary Converter - Binary Code [63:32] (BINARY_CODE_MSB)

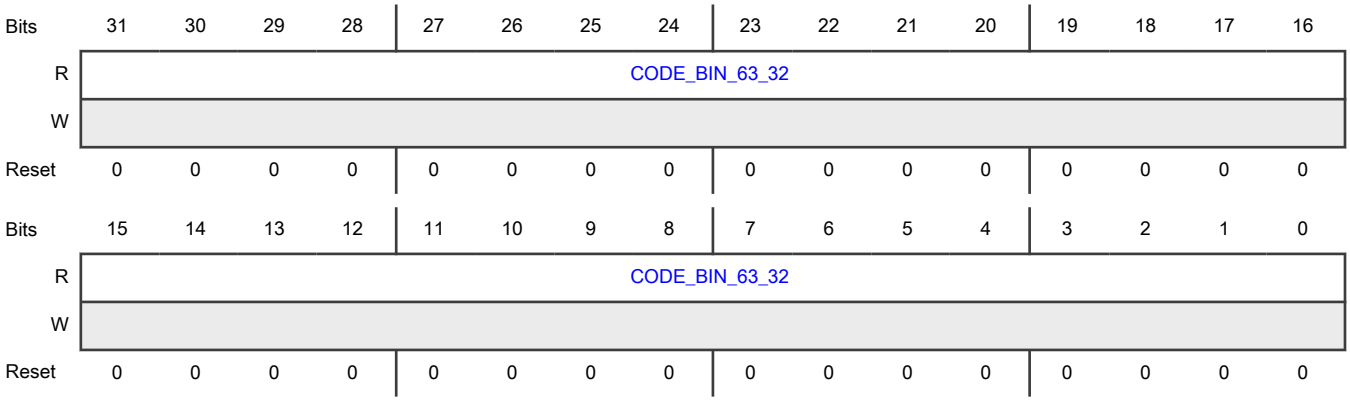
Offset

Register	Offset
BINARY_CODE_MSB	B6Ch

Function

Contains the most-significant portion of the code converted from Gray to binary coding. CPU1 can use it for timestamping operation using OSTIMER.

Diagram



Fields

Field	Function
31-0 CODE_BIN_63_32	Binary Code [63:32] Indicates the most-significant 32 bits of the 64-bit converted code.

29.6.4.18 Data Register n (DATA_REG0 - DATA_REG7)

Offset

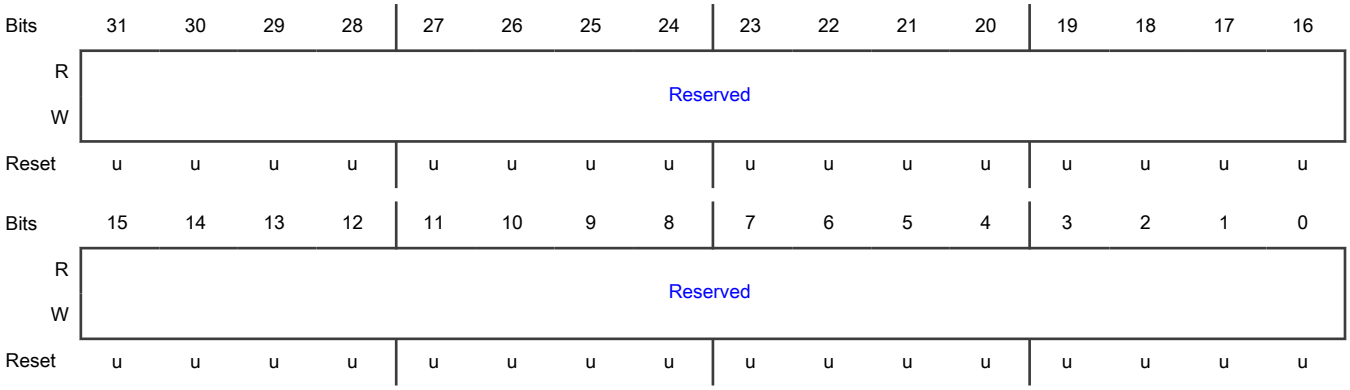
For n = 0 to 7:

Register	Offset
DATA_REGn	FE0h + (n × 4h)

Function

Preserves data on warm reset.

Diagram



Fields

Field	Function
31-0 —	Reserved

29.6.5 SYSCON4 register descriptions

29.6.5.1 SYSCON4 memory map

SYSCON4 base address: 400A_2000h

Offset	Register	Width (In bits)	Access	Reset value
28h	MMU2 Memory Control (MMU2_MEM_CTRL)	32	RW	0000_0000h
2Ch	XSPI2 Memory Control (XSPI2_MEM_CTRL)	32	RW	0000_0000h
30h	EZH-V Memory Control (EZHV_MEM_CTRL)	32	RW	0000_0000h
34h	LCDIF Memory Control (LCDIF_MEM_CTRL)	32	RW	0000_0000h
3Ch	JPEGDEC Memory Control (JPEGDEC_MEM_CTRL)	32	RW	0000_0000h
40h	PNGDEC Memory Control (PNGDEC_MEM_CTRL)	32	RW	0000_0000h
48h	MIPI Memory Control (MIPI_MEM_CTRL)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4Ch	VGPU Memory Control (VGPU_MEM_CTRL)	32	RW	0000_0000h
D0h	EZH-V Reset Base (EZHV_RSTBASE)	32	RW	0000_0000h
D4h	EZH-V MTVEC Reset Value (EZHV_RSTMTVEC)	32	RW	0000_0000h
D8h	EZH-V STVEC Reset Value (EZHV_RSTSTVEC)	32	RW	0000_0000h
DCh	EZH-V Interrupt Vector Offset (EZHV_VOFFSET_T)	32	RW	0000_0000h
E0h	EZH-V To Arm Interrupt Enable (EZHV2ARM_INT_EN)	32	RW	0000_0000h
E4h	EZH-V Interrupt Channel (EZHV2ARM_INT_CHAN)	32	RW	0000_0000h
E8h	Arm To EZH-V Interrupt Control (ARM2EZHV_INT_CTRL)	32	RW	0000_0000h
130h	MIPI_DSI Control (MIPI_DSI_CTRL)	32	RW	0000_0000h
280h	VDD2_MEDIA Auto Gating Enable (MEDIA_AUTOGATE_EN)	32	RW	0000_0000h
510h	USB0 Memory Control (USB0_MEM_CTRL)	32	RW	0000_0000h
514h	USB1 Memory Control (USB1_MEM_CTRL)	32	RW	0000_0000h
518h	uSDHC0 Memory Control (USDHC0_MEM_CTRL)	32	RW	0000_0000h
51Ch	uSDHC1 Memory Control (USDHC1_MEM_CTRL)	32	RW	0000_0000h
624h	PIO4 Configuration (GPIO_BANK4_CFG)	32	RW	0000_0000h
628h	PIO5 Configuration (GPIO_BANK5_CFG)	32	RW	0000_0000h
62Ch	PIO6 Configuration (GPIO_BANK6_CFG)	32	RW	0000_0000h
650h	USBPHY0 CLK Active (USBPHY0_CLK_ACTIVE)	32	RW	0000_0000h

29.6.5.2 MMU2 Memory Control (MMU2_MEM_CTRL)

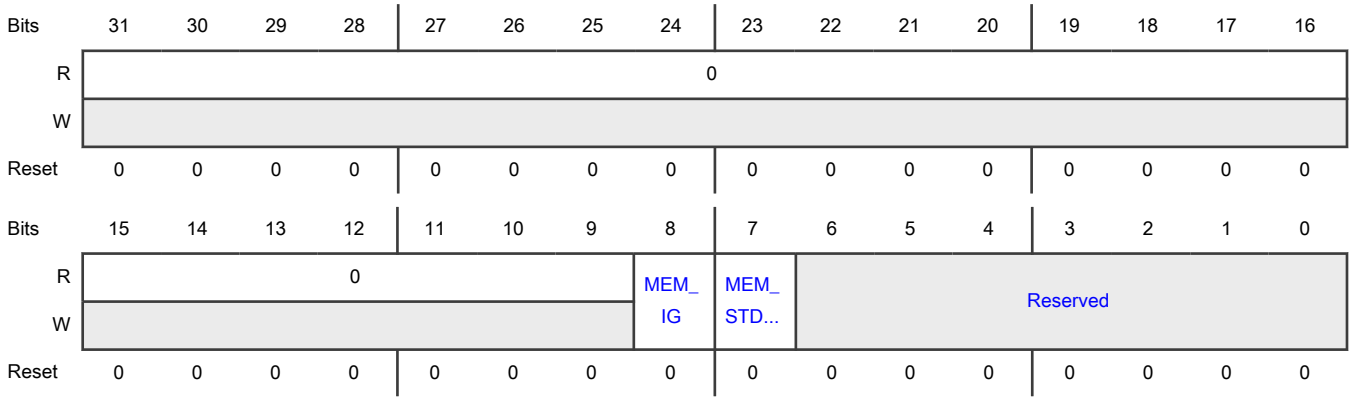
Offset

Register	Offset
MMU2_MEM_CTRL	28h

Function

Configures input and power gating for MMU2 memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.3 XSPI2 Memory Control (XSPI2_MEM_CTRL)

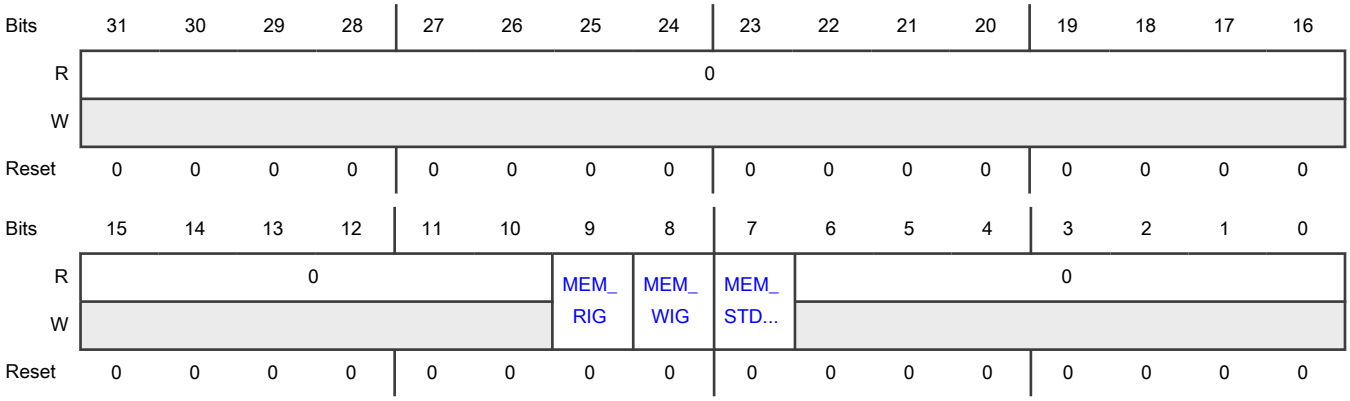
Offset

Register	Offset
XSPI2_MEM_CTRL	2Ch

Function

Configures read and write input gating and power gating for XSPI2 memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
6-0 —	Reserved

29.6.5.4 EZH-V Memory Control (EZHV_MEM_CTRL)

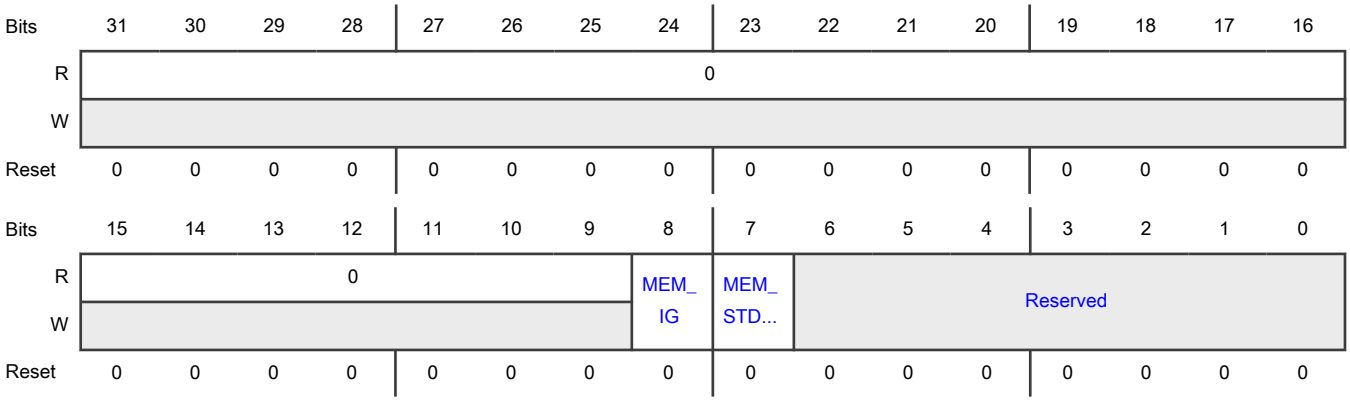
Offset

Register	Offset
EZHV_MEM_CTRL	30h

Function

Configures input and power gating for EZH-V memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
7	Memory Standby Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_STDBY	When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.5 LCDIF Memory Control (LCDIF_MEM_CTRL)

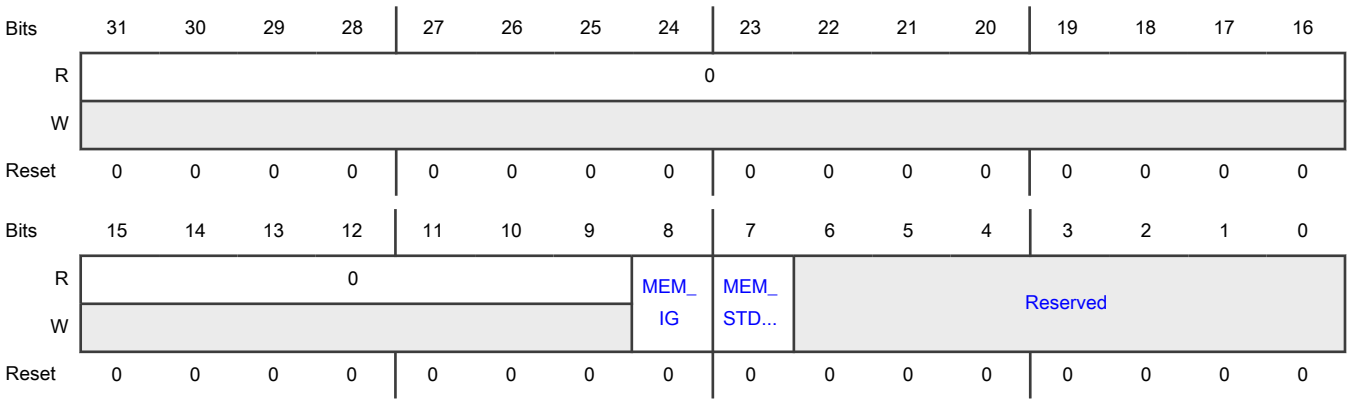
Offset

Register	Offset
LCDIF_MEM_CTRL	34h

Function

Configures input and power gating for LCDIF memory.

Diagram



Fields

Field	Function
31-9 —	Reserved
8	Memory Input Gating

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_IG	When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. **NOTE** When this bit is set, only memory inputs are gated and outputs still have data. For LCDIF, DONE flag might be observed. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.6 JPEGDEC Memory Control (JPEGDEC_MEM_CTRL)

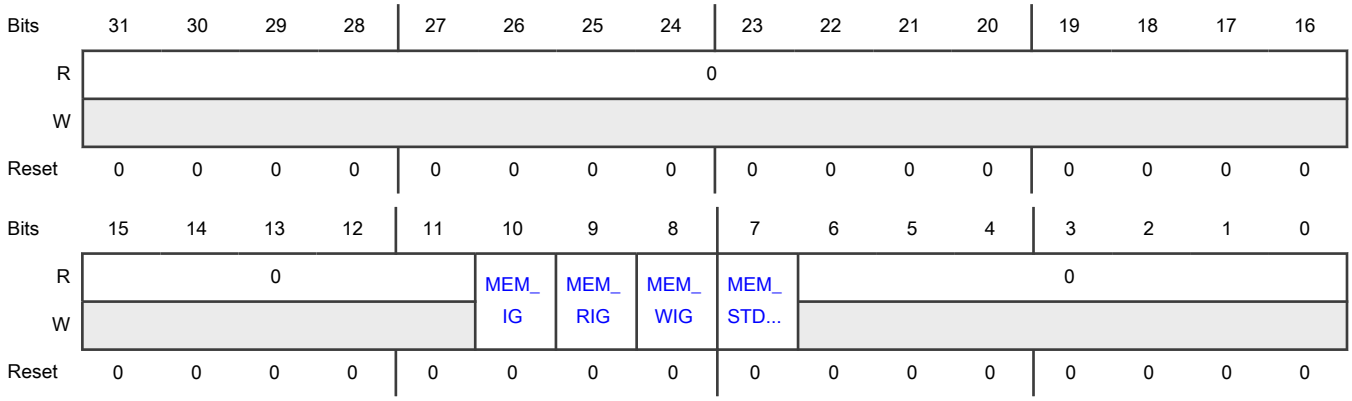
Offset

Register	Offset
JPEGDEC_MEM_CTRL	3Ch

Function

Configures read and write input gating and power gating for JPEGDEC memory.

Diagram



Fields

Field	Function
31-11 —	Reserved
10 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. **NOTE** When this bit is set, only memory inputs are gated and outputs still have data. For JPEGDEC, DONE flag might be observed. 0b - Inactive 1b - Active
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7	Memory Standby Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_STDBY	When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.7 PNGDEC Memory Control (PNGDEC_MEM_CTRL)

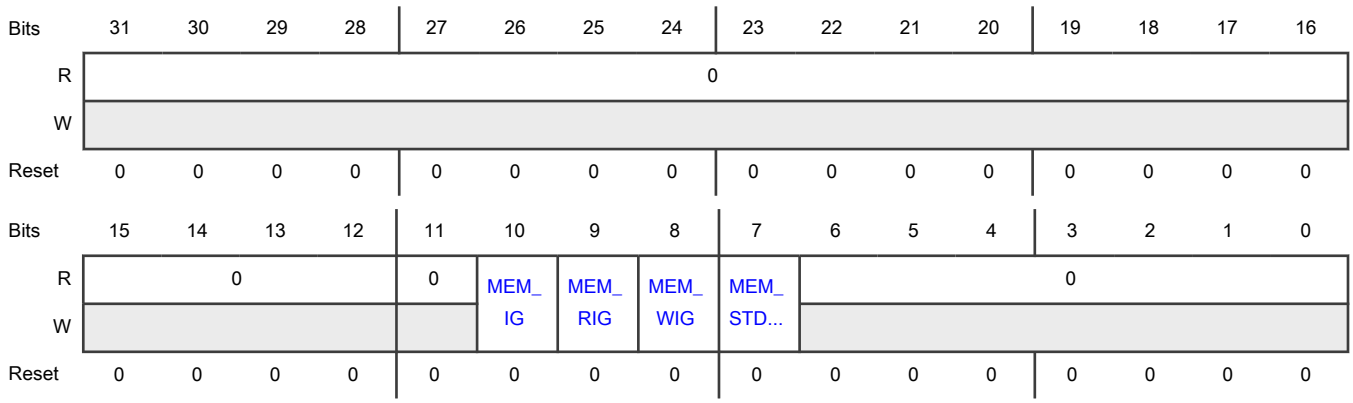
Offset

Register	Offset
PNGDEC_MEM_CTRL	40h

Function

Configures input and power gating for PNGDEC memory.

Diagram



Fields

Field	Function
31-12 —	Reserved
11	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
10 MEM_IG	Memory Input Gating When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. **NOTE** If this bit is set, it gates the memory (ROM) input. If you want the PNGDEC module to receive the ROM input again, you must clear this bit, and then reset PNGDEC by using RSTCTL4.PRSTCTL0[PNGDEC]. 0b - Inactive 1b - Active
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.8 MIPI Memory Control (MIPI_MEM_CTRL)

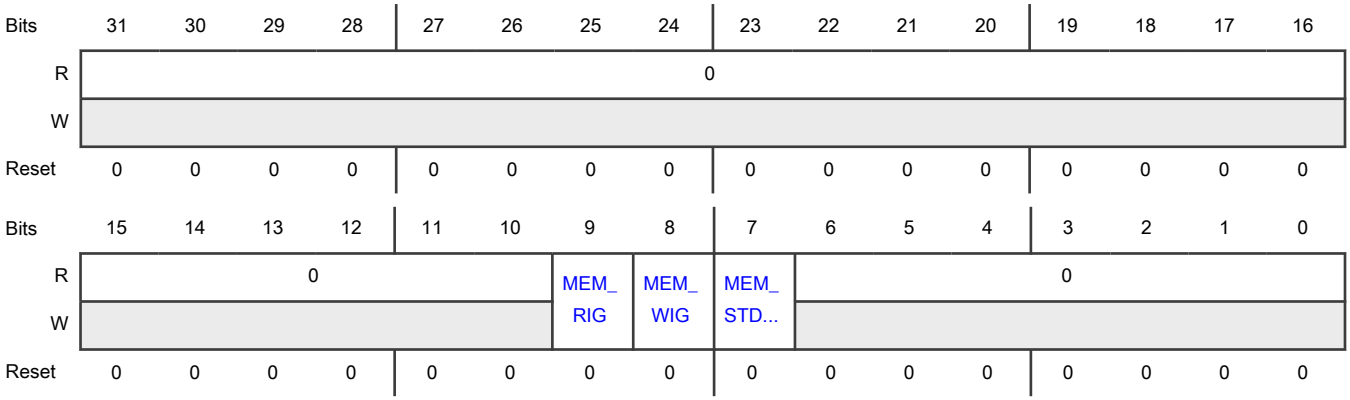
Offset

Register	Offset
MIPI_MEM_CTRL	48h

Function

Controls Mobile Industry Processor Interface (MIPI) memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7	Memory Standby Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_STDBY	When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.9 VGPU Memory Control (VGPU_MEM_CTRL)

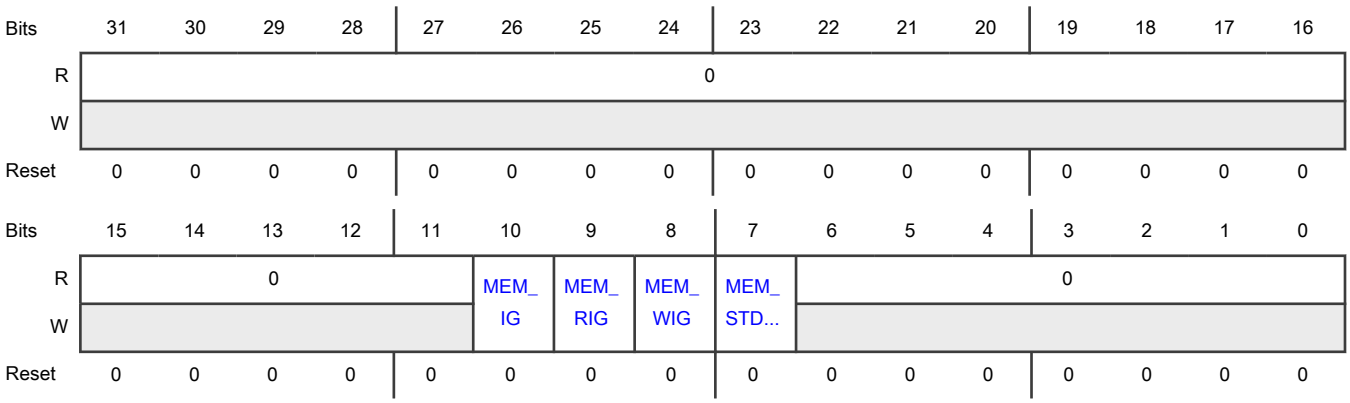
Offset

Register	Offset
VGPU_MEM_CTRL	4Ch

Function

Configures read and write input gating and power gating for VGPU memory.

Diagram



Fields

Field	Function
31-11 —	Reserved
10	Memory Input Gating

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_IG	When active, gates the selected memory inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads or writes are not needed for short periods of time. 0b - Inactive 1b - Active
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.10 EZH-V Reset Base (EZHV_RSTBASE)

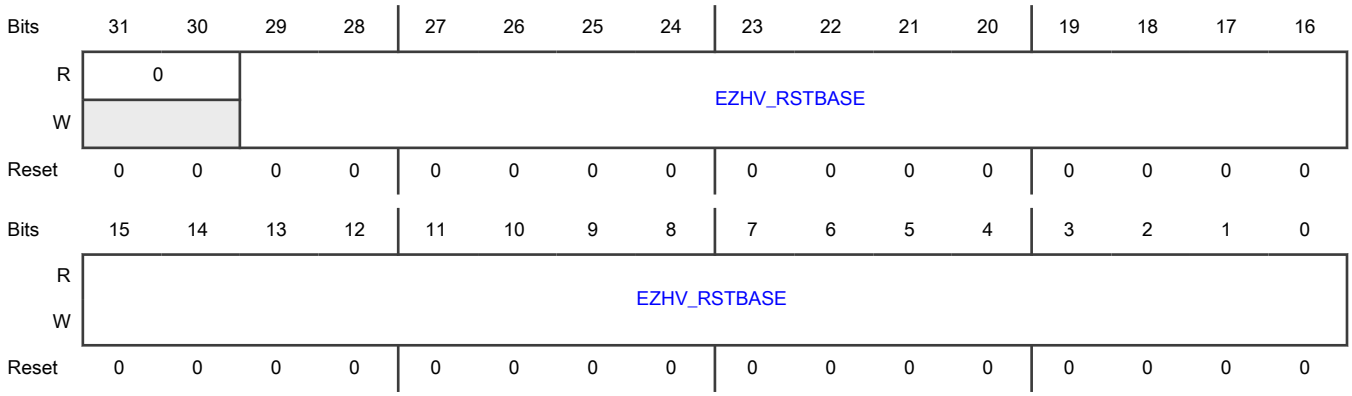
Offset

Register	Offset
EZHV_RSTBASE	D0h

Function

Allows system integrators to be able to specify and relocate the base address of the reset exception handler. These inputs are used to form the upper 30 bits of the instruction access following negation of reset which is used to fetch the initial instruction of the reset exception handler.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-0 EZHV_RSTBASE	Reset Exception Handler Base Address These bits should be driven to a value corresponding to the desired boot memory device in the system. Note the byte order: PW_DATA[29:0] is for EZHV_RSTBASE[0:29].

29.6.5.11 EZH-V MTVEC Reset Value (EZHV_RSTMTVEC)

Offset

Register	Offset
EZHV_RSTMTVEC	D4h

Function

Allows system integrators to be able to specify and relocate the base address of the exception vector table at reset. These inputs are used to initialize MTVEC on reset to a value appropriate for the system in order to support certain exceptions which could occur prior to software being able to configure the MTVEC register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EZHV_RSTMTVEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EZHV_RSTMTVEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 EZHV_RSTMTVEC	Reset Value of CSRs MTVEC These bits should be driven to a value corresponding to the desired initial exception vector table base address. Note the byte order: PW_DATA[31:0] is for EZHV_RSTBASE[0:31].

29.6.5.12 EZH-V STVEC Reset Value (EZHV_RSTSTVEC)

Offset

Register	Offset
EZHV_RSTSTVEC	D8h

Function

Allows system integrators to be able to specify and relocate the base address of the exception vector table at reset. These inputs are used to initialize STVEC on reset to a value appropriate for the system in order to support certain exceptions which could occur prior to software being able to configure the STVEC register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EZHV_RSTSTVEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EZHV_RSTSTVEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 EZHV_RSTSTVEC	Reset Value of CSRs STVEC These bits should be driven to a value corresponding to the desired initial exception vector table base address. Note the byte order: PW_DATA[31:0] is for EZHV_RSTBASE[0:31].

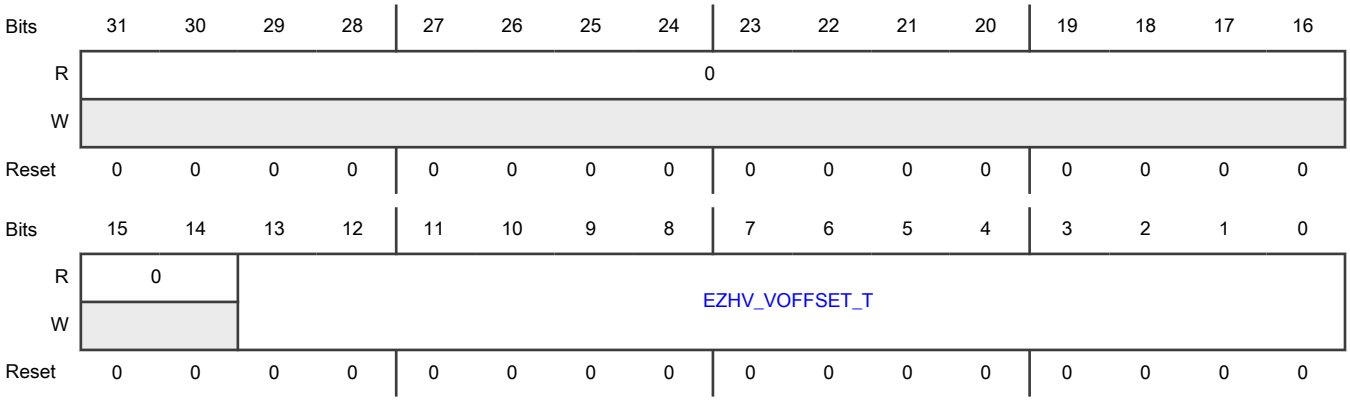
29.6.5.13 EZH-V Interrupt Vector Offset (EZHV_VOFFSET_T)

Offset

Register	Offset
EZHV_VOFFSET_T	DCh

Function
Provides a vector offset to be used when exception processing begins for an incoming interrupt request.

Diagram



Fields

Field	Function
31-14 —	Reserved
13-0 EZHV_VOFFSET_T	Interrupt Vector Offset for Vectored Interrupts Note the byte order: PW_DATA[13:0] is for EZHV_VOFFSET_T[0:13].

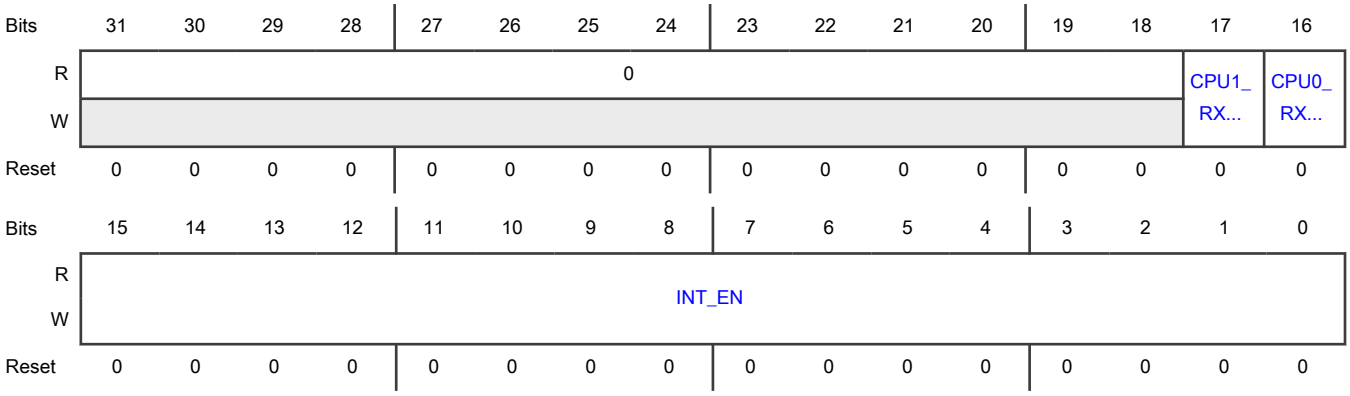
29.6.5.14 EZH-V To Arm Interrupt Enable (EZHV2ARM_INT_EN)

Offset

Register	Offset
EZHV2ARM_INT_EN	E0h

Function
Controls EZH-V to Arm interrupts.

Diagram



Fields

Field	Function
31-18 —	Reserved
17 CPU1_RXEV_EN	CPU1 RXEV Enable Generates EZH-V event out to CPU1 receive event (RXEV). 0b - Disables 1b - Enables
16 CPU0_RXEV_EN	CPU0 RXEV Enable Generates EZH-V event out to CPU0 receive event (RXEV). 0b - Disables 1b - Enables
15-0 INT_EN	Interrupt Enable Enables the EZH-V interrupt channel(s) to Cortex-M33 by writing 1 to the corresponding bit(s).

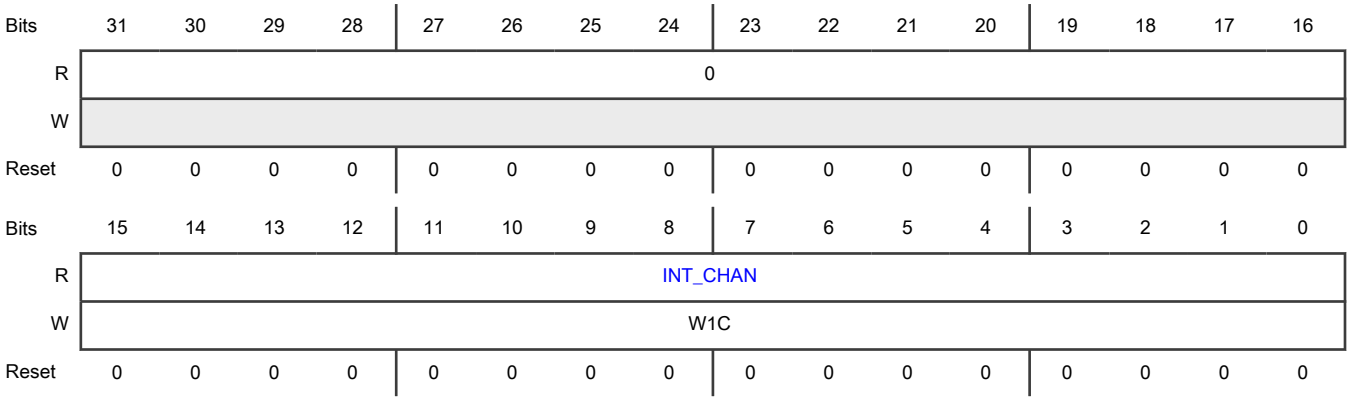
29.6.5.15 EZH-V Interrupt Channel (EZHV2ARM_INT_CHAN)

Offset

Register	Offset
EZHV2ARM_INT_CHAN	E4h

Function
Controls clearing of EZH-V interrupt outputs. Each interrupt outputs can be cleared by writing 1 to the corresponding bit.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 INT_CHAN	EZH-V Interrupt Outputs Write 1 to clear for each bit.

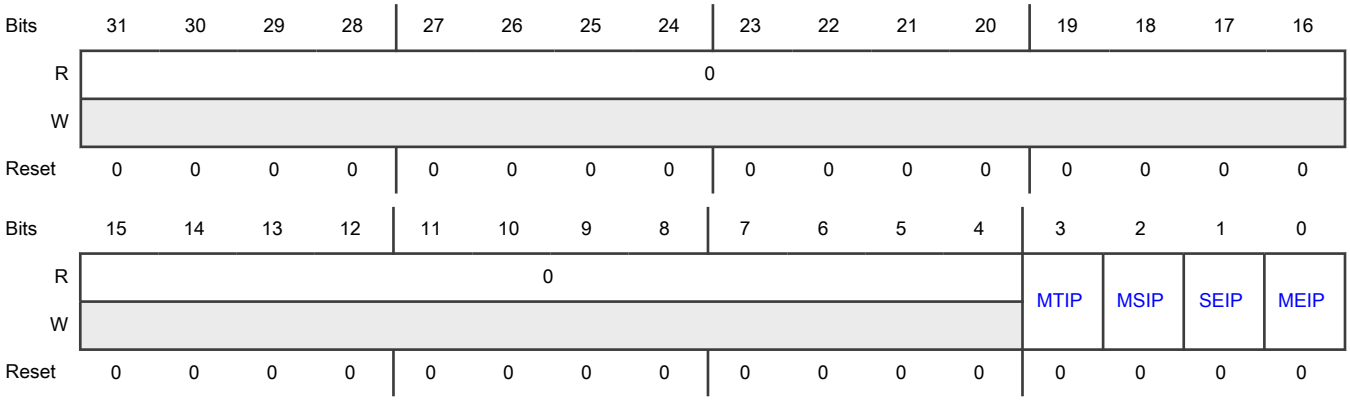
29.6.5.16 Arm To EZH-V Interrupt Control (ARM2EZHV_INT_CTRL)

Offset

Register	Offset
ARM2EZHV_INT_CTRL	E8h

Function
Controls Arm core to EZH-V interrupts. CPU0 or CPU1 writes 1 to trigger INPUTMUX for relevant interrupts.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 MTIP	Machine Timer Input Interrupt Request 0b - Disables 1b - Enables
2 MSIP	Machine Software Input Interrupt Request 0b - Disables 1b - Enables
1 SEIP	Supervisor External Input Interrupt Request 0b - Disables 1b - Enables
0 MEIP	Machine External Input Interrupt Request 0b - Disables 1b - Enables

29.6.5.17 MIPI_DSI Control (MIPI_DSI_CTRL)

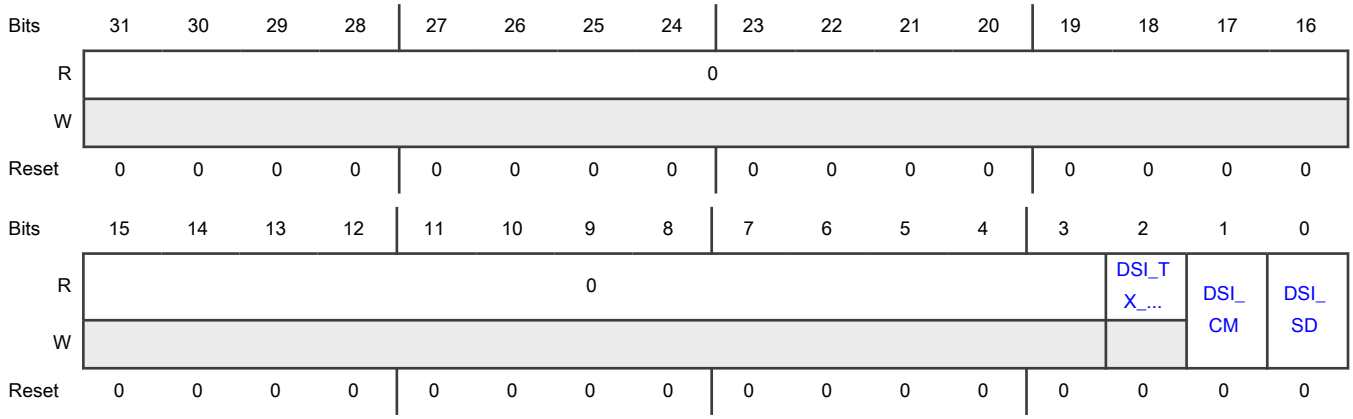
Offset

Register	Offset
MIPI_DSI_CTRL	130h

Function

Controls DSI shutdown, color mode, and transmit status.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 DSI_TX_ACTIVE	DSI Transmit Active Asserted when the MIPI_DSI is actively transmitting data or when it has accepted a request from the user but has not yet started transmitting.
1 DSI_CM	DSI Color Mode Control Color mode control for Type-4 display only. This feature is used to configure the display on Normal mode or Low color mode. When bit is configured to low-color mode, display works in 8-bit mode and display power is reduced. **NOTE** Both signals [DSI_SD] and [DSI_CM] are synchronized with the DPI clock (dpi_pclk) coming from display controller. Turn on the DPI mode of LCDIF to have [DSI_SD] and [DSI_CM] signals being propagated to DSI interface. 0b - Normal (full color) mode 1b - Low color mode (8-bit)
0 DSI_SD	DSI Shutdown Control Shutdown control for Type-4 display only. By specifying whether to send shutdown command to the Type-4 display, this feature triggers the display module to shut down the streaming video interface to reduce power consumption. **NOTE** Both signals [DSI_SD] and [DSI_CM] are synchronized with the DPI clock (dpi_pclk) coming from display controller. Turn on the DPI mode of LCDIF to have [DSI_SD] and [DSI_CM] signals being propagated to DSI interface. 0b - Do not send 1b - Send

29.6.5.18 VDD2_MEDIA Auto Gating Enable (MEDIA_AUTOGATE_EN)

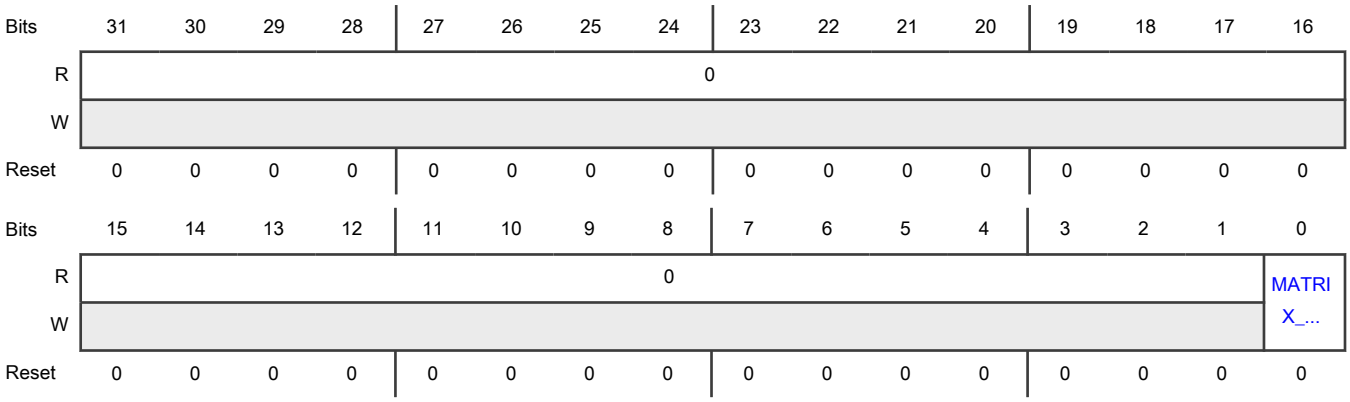
Offset

Register	Offset
MEDIA_AUTOGATE_EN	280h

Function

Controls clock gating for AHB bus matrix, AHB to APB bridge, and the AIPS and AXBS in VDD2_MEDIA domain. This register allows selectively enabling automatic clock gating for the modules. By default, automatic clock gating turns off clocks to the modules after 2 bus clocks with no bus transaction activity. This saves power when the corresponding modules are not used for a period of time.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 MATRIX_EN	VDD2_MEDIA Automatic Clock Gating Enable Controls clock gating for AHB bus matrix, AHB to APB bridge, and the AIPS and AXBS in VDD2_MEDIA domain. You can set it to disable the module's clock in order to save power when the module is not working. 0b - Disables clock gating (continuous clock) 1b - Enables clock gating

29.6.5.19 USB0 Memory Control (USB0_MEM_CTRL)

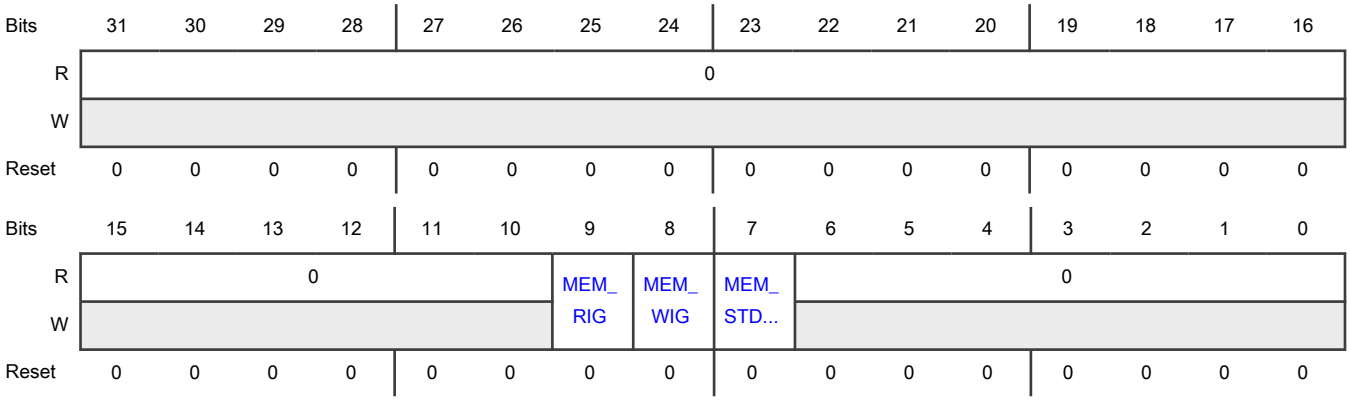
Offset

Register	Offset
USB0_MEM_CTRL	510h

Function

Configures read and write input gating and power gating for USB0 memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
6-0 —	Reserved

29.6.5.20 USB1 Memory Control (USB1_MEM_CTRL)

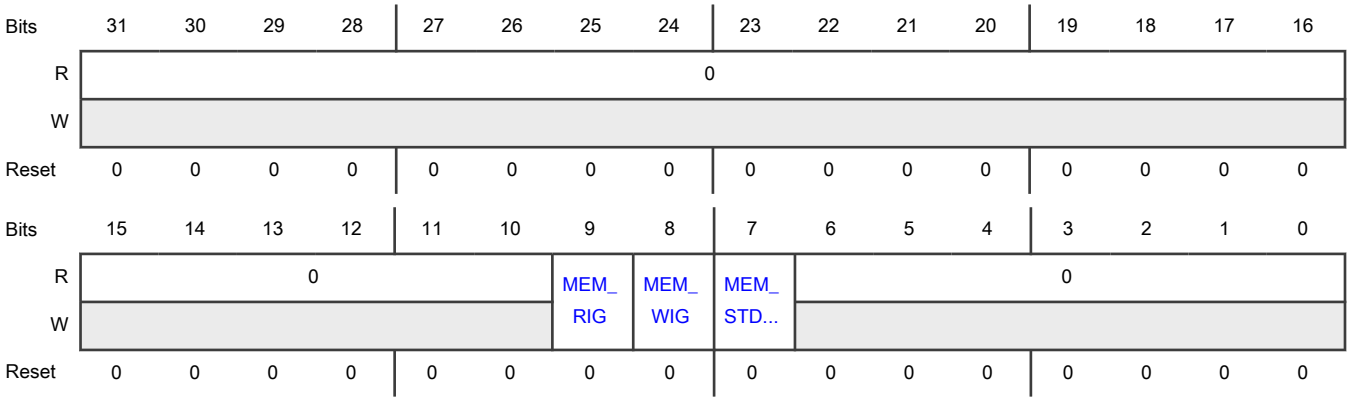
Offset

Register	Offset
USB1_MEM_CTRL	514h

Function

Configures read and write input gating and power gating for USB1 memory.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8	Memory Write Input Gating

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEM_WIG	When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.21 uSDHC0 Memory Control (USDHC0_MEM_CTRL)

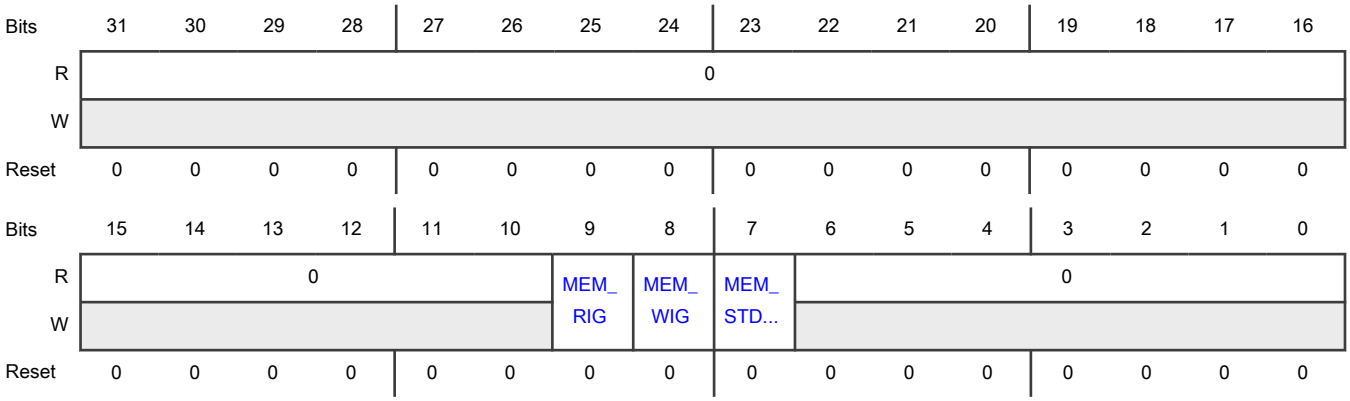
Offset

Register	Offset
USDHC0_MEM_CTRL	518h

Function

Controls the memory of the ultra Secured Digital Host Controller (uSDHC) 0.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

29.6.5.22 uSDHC1 Memory Control (USDHC1_MEM_CTRL)

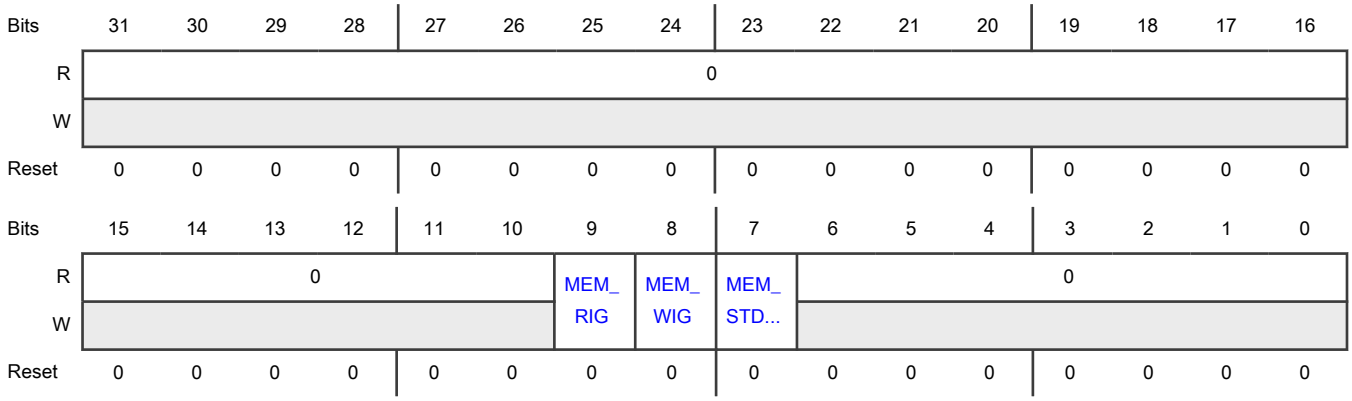
Offset

Register	Offset
USDHC1_MEM_CTRL	51Ch

Function

Controls the memory of the ultra Secured Digital Host Controller (uSDHC) 0.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 MEM_RIG	Memory Read Input Gating When active, gates the selected memory read inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if reads are not needed for short periods of time. 0b - Inactive 1b - Active
8 MEM_WIG	Memory Write Input Gating When active, gates the selected memory write inputs to reduce active power associated with input toggles. This bit can be set to reduce RAM power consumption if writes are not needed for short periods of time. 0b - Inactive 1b - Active
7 MEM_STDBY	Memory Standby Mode When active, puts the memory into standby mode. This bit can be set to reduce RAM power consumption if the RAM is never used or will not be used for long periods of time. There is a brief increase in RAM power created by the shutdown sequence caused by a transition from regular mode to standby mode. Therefore, standby mode should only be used when the RAM array will not be accessed for long periods of time. 0b - Inactive 1b - Active
6-0 —	Reserved

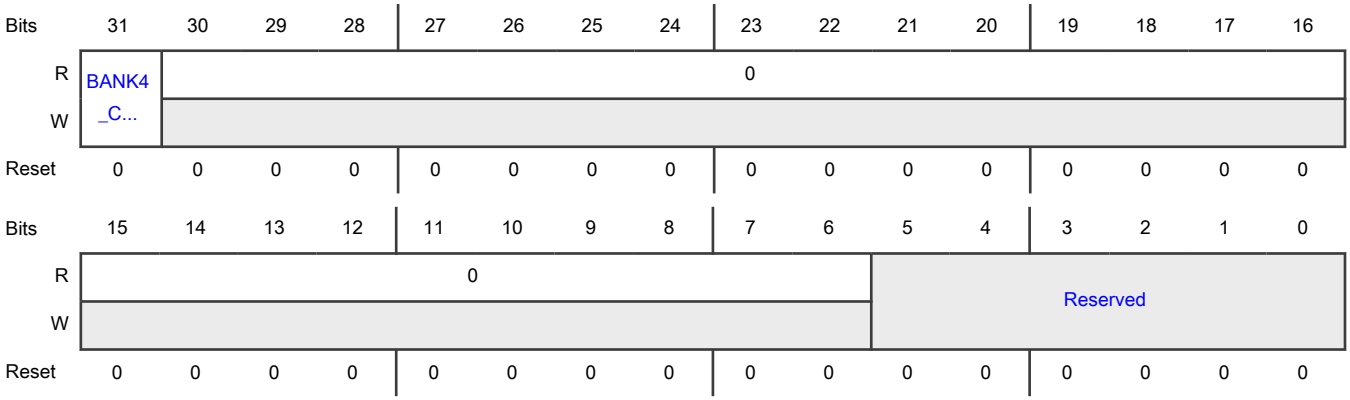
29.6.5.23 PIO4 Configuration (GPIO_BANK4_CFG)

Offset

Register	Offset
GPIO_BANK4_CFG	624h

Function
Configures PIO4 (GPIO bank 4).

Diagram



Fields

Field	Function
31 BANK4_CFG	Transmit Driver Size Configuration Used to adjust Rout at different VDDIO_4 supply levels, for 1.8 V VDDIO_4 it should be set to low, and for 1.2 V VDDIO_4 it should be set to high. When this field is high, transmit transistor size is bigger than being set to low. 0b - 1.8 V 1b - 1.2 V
30-6 —	Reserved
5-0 —	Reserved

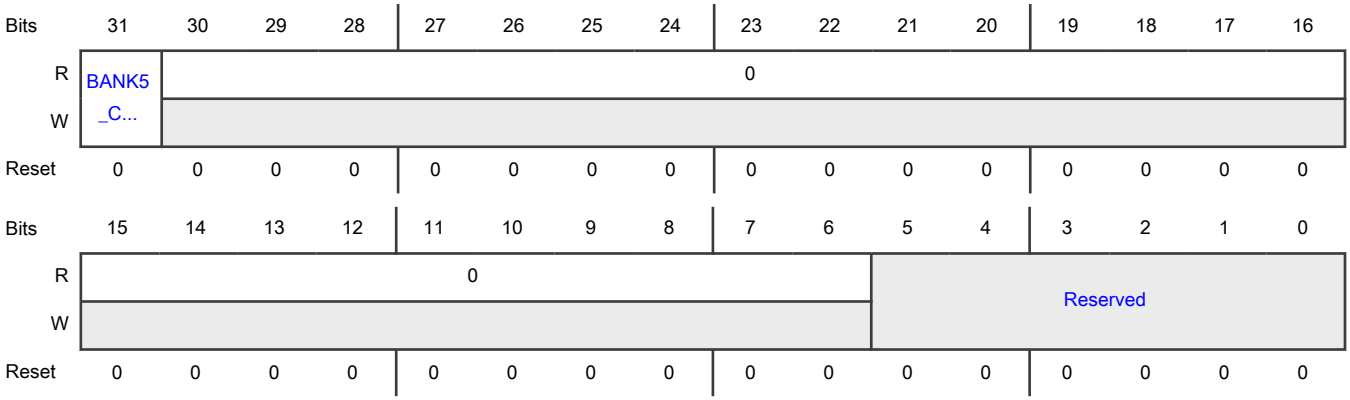
29.6.5.24 PIO5 Configuration (GPIO_BANK5_CFG)

Offset

Register	Offset
GPIO_BANK5_CFG	628h

Function
Configures PIO5 (GPIO bank 5).

Diagram



Fields

Field	Function
31 BANK5_CFG	Transmit Driver Size Configuration Used to adjust Rout at different VDDIO_5 supply levels, for 1.8 V VDDIO_5 it should be set to low, and for 1.2 V VDDIO_5 it should be set to high. When this field is high, transmit transistor size is bigger than being set to low. 0b - 1.8 V 1b - 1.2 V
30-6 —	Reserved
5-0 —	Reserved

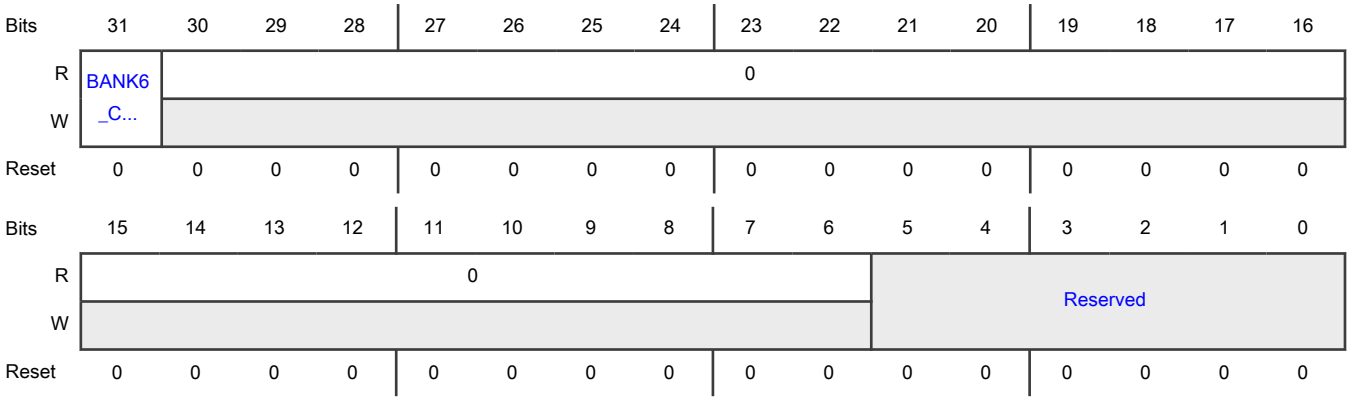
29.6.5.25 PIO6 Configuration (GPIO_BANK6_CFG)

Offset

Register	Offset
GPIO_BANK6_CFG	62Ch

Function
Configures PIO6 (GPIO bank 6).

Diagram



Fields

Field	Function
31 BANK6_CFG	Transmit Driver Size Configuration Used to adjust Rout at different VDDIO_6 supply levels, for 1.8 V VDDIO_6 it should be set to low, and for 1.2 V VDDIO_6 it should be set to high. When this field is high, transmit transistor size is bigger than being set to low. 0b - 1.8 V 1b - 1.2 V
30-6 —	Reserved
5-0 —	Reserved

29.6.5.26 USBPHY0 CLK Active (USBPHY0_CLK_ACTIVE)

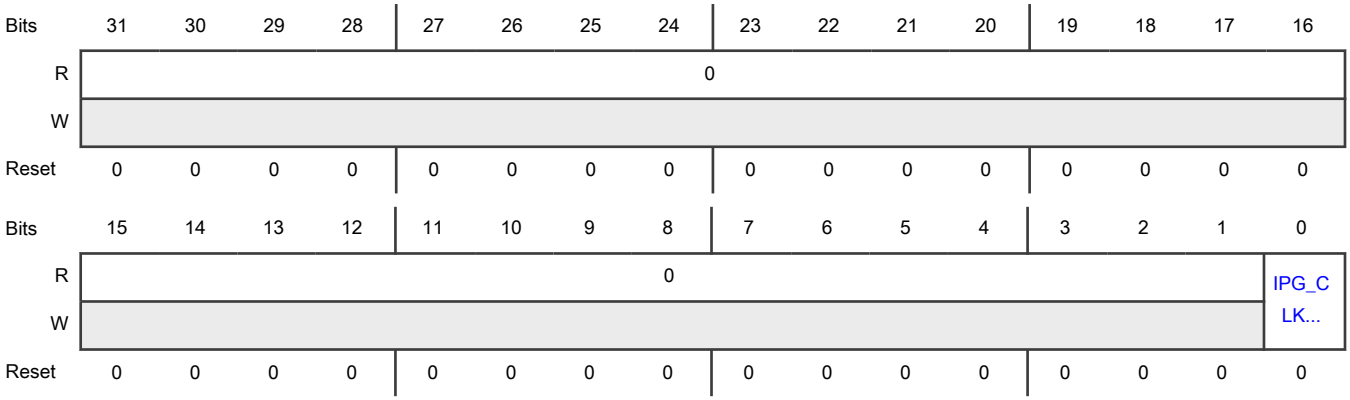
Offset

Register	Offset
USBPHY0_CLK_ACTIVE	650h

Function

Controls IPG_CLK for USBPHY0.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 IPG_CLK_ACTIVE	IPG_CLK Active Controls whether IPG_CLK in USBPHY0 is active. 0b - IPG_CLK is inactive 1b - IPG_CLK is active

Chapter 30
Sleep Controller (SLEEPCON)

30.1 Chip-specific SLEEPCON information

Table 253. Reference links to related information

Topic	Related module	Reference
Full description	SLEEPCON	SLEEPCON
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller

NOTE

The SLEEPCON module includes some security functionality that is not described here. Refer to the i.MX RT700 Security Reference Manual for more details.

30.1.1 Module instances

This device has two instances of the SLEEPCON module: SLEEPCON0 and SLEEPCON1.

Table 254. SLEEPCON instances

Features	SLEEPCON0	SLEEPCON1
Located in	VDD2_COM	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Controls modules in	VDD2_COMP, VDD2_MEDIA, VDDN_MEDIA, VDDN_COM, VDD1P8_AON, VDD2_COM, VDD1_SENSE (shared)	VDD2_MEDIA, VDDN_MEDIA, VDDN_COM, VDD1P8_AON, VDD2_COM, VDD1_SENSE
pclk clock source	COMPUTE_MAIN_CLK	SENSE_MAIN_CLK
pclk_regbank clock source	COMPUTE_MAIN_CLK	SENSE_MAIN_CLK

30.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

30.2 Overview

SLEEPCON is a programmable low-power control module within the chip's overall power management and control system. Working with PMC and CLKCTL, SLEEPCON helps coordinate power mode transitions, especially the low-power transitions for individual modules.

30.2.1 Block diagram

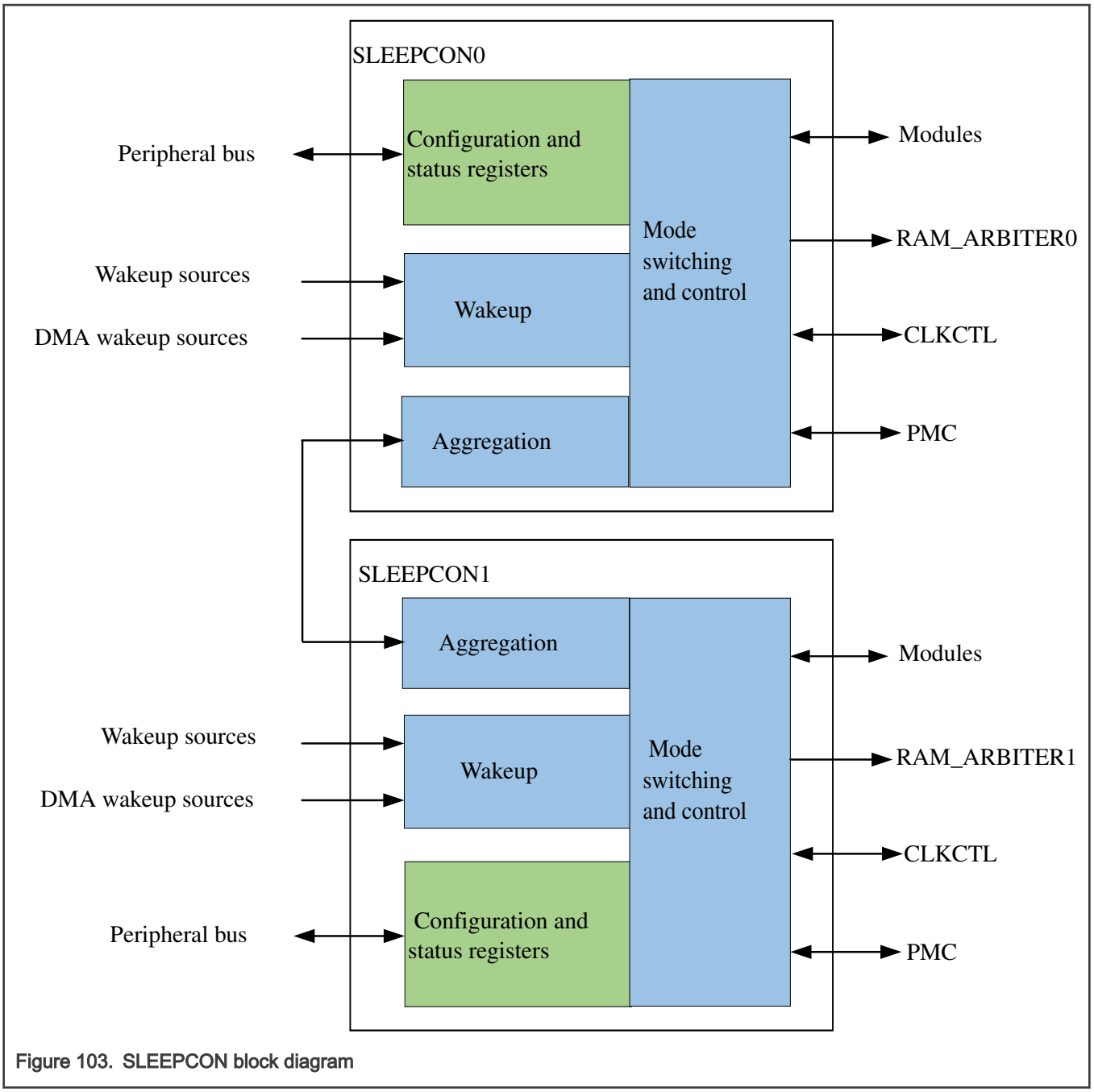


Figure 103. SLEEPCON block diagram

30.2.2 Features

SLEEPCON features include:

- Enables wakeup interrupts for on-chip modules
- Performs the low-power request handshake for individual modules to ensure safe transitioning to a low-power state
- Enables the DMA wakeup mechanism for supported modules
- Triggers CLKCTL to shut off associated clocks to modules, system components and memory
- Enables the low-power sequence interaction with PMC

30.3 Functional description

SLEEPCON provides for the following functions within the power management and control system:

- Requesting low-power for modules
- DMA wakeup

30.3.1 Aggregation for shared modules

Table 255 lists the shared modules that can be controlled by both SLEEPCONs. These modules are mainly clock sources and main clocks of different domains. These bits in RUNCFG and SLEEPCFG registers can power down the corresponding modules or gate the clock. SLEEPCON0 and SLEEPCON1 control them together. These bits are not relevant to the low-power requests. A shared module is powered down only when the related bit is configured as power down in both SLEEPCONs.

Table 255. Shared modules

Shared modules	Control bit in RUNCFG/SLEEPCFG
ADC0	ADC_PD
Audio PLL	AUDPLLLDO_PD
	AUDPLLANA_PD
Main PLL	PLLLDO_PD
	PLLANA_PD
LPOSC 1 MHz	LPOSC_PD
FRO2	FRO2_PD FRO2_GATE
Xtal Oscillator	XTAL_PD
Media domain main clock	MEDIA_MAINCLK_SHUTOFF
VDDN_COM domain main clock	COMN_MAINCLK_SHUTOFF
RAM Arbiter1 clock	RAM1_CLK_SHUTOFF
RAM Arbiter0 clock	RAM0_CLK_SHUTOFF
VDD1_SENSE domain main clock-shared modules	SENSES_MAINCLK_SHUTOFF
VDD1_SENSE domain main clock private modules	SENSEP_MAINCLK_SHUTOFF

30.3.2 Requesting low-power for modules

For modules that do not need to remain active, software can first execute a low-power request-and-acknowledge handshake to check for module activity before gating its clock in active mode/low-power modes. The handshake ensures modules have safely completed any current transactions before gating their clocks.

The general handshake with a module is as follows:

1. Make the request by configuring the relevant low-power control register (xxx_CCTRLn and xxx_TCTRLn).
2. Poll for acknowledgement by reading the corresponding low-power status register (xxx_CSTATn and xxx_TSTATn). If the value is 1b, go to the next step.

NOTE

Most modules can send back low-power accept signals to SLEEPCON, except the following modules that are in reset status or their related peripheral clocks are off:

- eDMA0-3
- XSPI2
- ADC0
- EZH-V
- VGPU

3. Gate the module's clock by configuring the relevant PSCCTLn register field in the CLKCTL.

Factors affecting the specific details of the overall handshake sequence are:

- Whether the module can initiate bus transactions: See Order of module requests
- Whether both CPUs can access the module: See Private versus shared modules

NOTE

For some peripherals like LP_FLEXCOMM, LPSPI and FLEXIO, low-power requests can be made for different logic parts separately. One part of the logic of these peripheral is driven by the bus clock, and the other part is driven by its function clock. Peripheral that is not being used in Deep Sleep mode should complete the low-power handshake process and detect an accept or acknowledge state. ADC0, ACMP0, MICFILF, GPIO, SAI and I3C low-power handshaking is mandatory to enable them to generate a wake-up signal in Deep Sleep mode. Peripherals with separated bus and function clock handshaking mechanisms such as LP_FLEXCOMM, LPSPI, LPI2C15, FLEXIO, the bus portion of the handshaking is mandatory, otherwise the module will not be able to generate a wake-up signal in Deep Sleep mode. Low-power handshaking process is also needed for DSR/Deep Power Down/Full Deep Power down modes.

30.3.2.1 Order of module requests

To ensure coherency of operations and data flow, the order in which modules should be clock-gated is as follows:

Table 256. Order of module requests

Step	Modules that can...	Use these request registers...	And these status registers...
1	Initiate bus transactions (bus controllers)	SHA_MED_CCTRL0 and PRIVATE_CCTRL0	SHA_MED_CSTAT0 and PRIVATE_CSTAT0
2	Respond to bus transactions (bus targets)[1]	SHA_MED_TCTRL0, SHA_SEN_TCTRL0, and PRIVATE_TCTRLn	SHA_MEDSEN_TSTAT0 and PRIVATE_TSTATn

1. If a module has both bus logic and function logic, make the bus low-power request first, followed by the function low-power request.

30.3.2.2 Private versus shared modules

Modules can be categorized as private or shared:

- Private modules can be controlled by one and only one of the CPUs.
- Shared modules can be controlled by both CPUs. Shared modules require arbitration because both CPUs could attempt to affect their power state. See Arbitrating low-power requests for shared modules. A shared module powers down only after both SLEEPCON0 and SLEEPCON1 require it to power down. The shared module is on if either or both SLEEPCON instances need it on.

  The list below is the shared modules for low power requests:

— NIC_MEDIA1
— NIC_MEDIA0
— FLEXIO
— LPSPI16
— LPSPI14
— XSPI2
— EZH-V
— VGPU
— RTC
— LPI2C15
— MICFIL
— ACMP0
— ADC0

30.3.2.3 Arbitrating low-power requests for shared modules

For shared modules jointly controlled by SLEEPCON0 and SLEEPCON1, a hardware arbiter determines whether a low-power request can be passed along to a target module. The arbiter drives the module to enter its low-power state only when both SLEEPCON instances make a low-power request. Otherwise, as long as at least one instance needs the module to keep running, the module does not enter the low-power state.

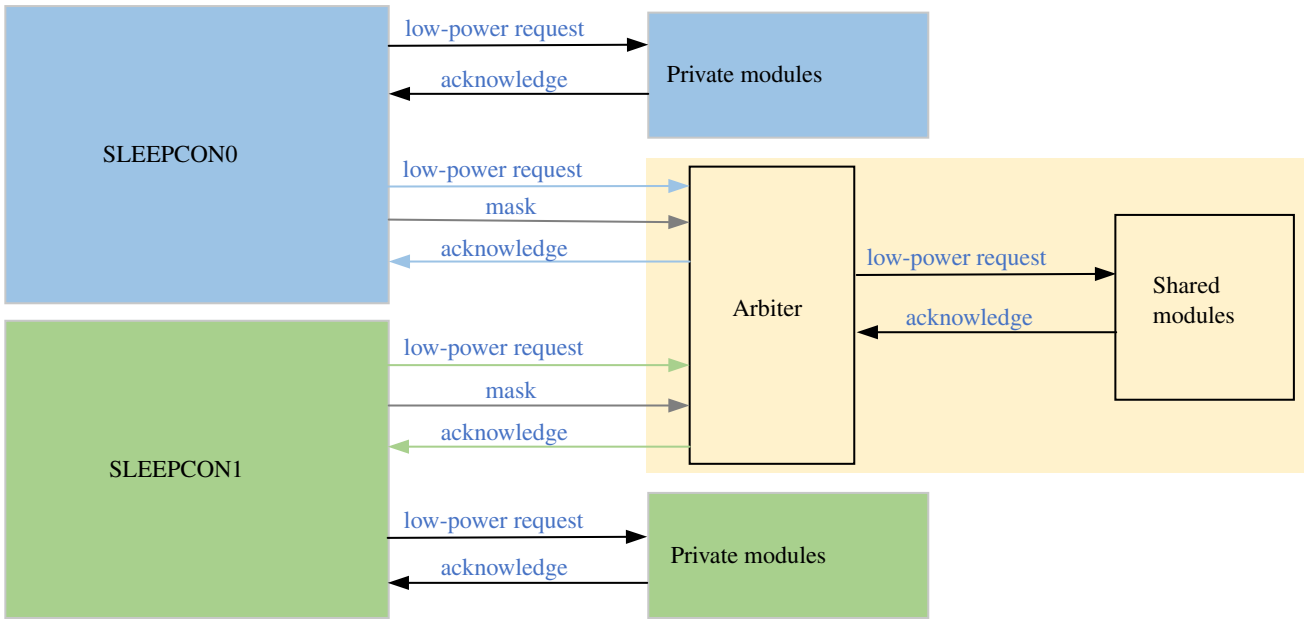


Figure 104. SLEEPCON low-power arbiter

30.3.2.4 Masking shared modules from arbitration

If software running on CPU0 or CPU1 does not require a shared module for an application, the software can remove itself from arbitration consideration by setting the module's mask bit in SHARED_MASK0 in the corresponding SLEEPCON instance. In this case, the other SLEEPCON instance has exclusive control over low-power requests to the module.

NOTE

A shared module's mask bit in SHARED_MASK0 must not be configured as 0b in both SLEEPCONs.

30.3.3 Clocking

Table 257. SLEEPCON clocks

Type of clock	Description
Module clock (pclk)	Functions the SLEEPCON module.
Module register clock (pclk_regbank)	Controls the access to the SLEEPCON registers.

See the chip-specific information for more details.

30.3.4 DMA wakeup

DMA wakeup is available for some modules in the VDD2_COMP, VDD1_SENSE and VDD2_MEDIA domains. It is an automatic hardware mechanism that handles DMA transactions for selected modules without core processing. Software can configure individual DMA channels to generate a wakeup when they receive a DMA request. The DMA wakeup triggers an exit from Deep Sleep mode except the core, which remains sleeping.

In order to use DMA wakeup, the modules which support this feature must complete the low-power handshake before entering the Deep Sleep mode.

- For DMA wakeup by eDMA0 or eDMA1: SAI0-2, LP_FLEXCOMM0-13, LPSPI14, LPSPI16, FLEXIO, eDMA0 and eDMA1 need to complete low-power handshake process.
- For DMA wakeup by eDMA2 or eDMA3: SAI3, LP_FLEXCOMM17-20, LPSPI14, LPSPI16, FLEXIO, eDMA2 and eDMA3 need to complete low-power handshake process.

If a module has seperate bus and function low-power handshake, the bus low-power handshake is needed to complete.

See Temporary wakeup for DMA for procedures.

An interrupt that occurs during a DMA wakeup causes an exit from the deep sleep mode without affecting the DMA transfer.

30.4 Memory map and register definition

This section includes the SLEEPCON0 and SLEEPCON1 module memory maps and detailed descriptions of all registers.

30.4.1 SLEEPCON0 register descriptions

30.4.1.1 SLEEPCON0 memory map

SLEEPCON0 base address: 4000_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Sleep Configuration (SLEEPCFG)	32	RW	A000_F502h
10h	Run Configuration (RUNCFG)	32	RW	A000_F502h
20h	RUNCFG Set (RUNCFG_SET)	32	W	0000_0000h
30h	RUNCFG Clear (RUNCFG_CLR)	32	W	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
60h	Wakeup Enable 0 (WAKEUPEN0)	32	RW	0000_0000h
64h	Wakeup Enable 1 (WAKEUPEN1)	32	RW	0000_0000h
68h	Wakeup Enable 2 (WAKEUPEN2)	32	RW	0000_0000h
6Ch	Wakeup Enable 3 (WAKEUPEN3)	32	RW	0000_0000h
70h	Wakeup Enable 4 (WAKEUPEN4)	32	RW	0000_0000h
A0h	Wakeup Enable 0 Set (WAKEUPEN0_SET)	32	RW	0000_0000h
A4h	Wakeup Enable 1 Set (WAKEUPEN1_SET)	32	RW	0000_0000h
A8h	Wakeup Enable 2 Set (WAKEUPEN2_SET)	32	RW	0000_0000h
ACh	Wakeup Enable 3 Set (WAKEUPEN3_SET)	32	RW	0000_0000h
B0h	Wakeup Enable 4 Set (WAKEUPEN4_SET)	32	RW	0000_0000h
100h	Wakeup Enable 0 Clear (WAKEUPEN0_CLR)	32	RW	0000_0000h
104h	Wakeup Enable 1 Clear (WAKEUPEN1_CLR)	32	RW	0000_0000h
108h	Wakeup Enable 2 Clear (WAKEUPEN2_CLR)	32	RW	0000_0000h
10Ch	Wakeup Enable 3 Clear (WAKEUPEN3_CLR)	32	RW	0000_0000h
110h	Wakeup Enable 4 Clear (WAKEUPEN4_CLR)	32	RW	0000_0000h
164h	Power Down Wait (PWRDOWN_WAIT)	32	RW	0000_000Ch
190h	eDMA Wakeup Enable (HW_WAKE)	32	RW	0000_0000h
194h	eDMA Wakeup Set (HW_WAKE_SET)	32	W	0000_0000h
198h	eDMA Wakeup Clear (HW_WAKE_CLR)	32	W	0000_0000h
1A0h	Shared Resources Mask (SHARED_MASK0)	32	RW	0000_0000h
1A4h	Shared Resources Mask Set (SHARED_MASK0_SET)	32	W	0000_0000h
1A8h	Shared Resources Mask Clear (SHARED_MASK0_CLR)	32	W	0000_0000h
1B0h	Media Domain Shared Controller Low-power Control (SHA_MED_CCTRL0)	32	RW	0000_0000h
1B4h	Media Domain Shared Resources Target Low-power Control (SHA_MED_TCTRL0)	32	RW	0000_0000h
1BCh	Shared VDD1_SENSE Domain Targets Low-power Control (SHA_SEN_TCTRL0)	32	RW	0000_0000h
1C0h	Private Resources Controllers Low-power Control 0 (PRIVATE_CCTRL0)	32	RW	0000_0000h
1C4h	Private Resources Targets Low-power Control 0 (PRIVATE_TCTRL0)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
1C8h	Private Resources Targets Low-power Control 1 (PRIVATE_TCTRL1)	32	RW	0000_0000h
1CCh	Private Resources Targets Low-power Control 2 (PRIVATE_TCTRL2)	32	RW	0000_0000h
1D0h	Media Domain Shared Controllers Low-power Status (SHA_MED_CSTAT0)	32	R	0000_0000h
1D4h	Media and Sense Domain Shared Targets Low-power Status (SHA_MEDSEN_TSTAT0)	32	R	0000_0000h
1E0h	Private Resources Controller Low-power Status (PRIVATE_CSTAT0)	32	R	0000_0000h
1E4h	Private Resources Targets Low-power Status 0 (PRIVATE_TSTAT0)	32	R	0000_0000h
1E8h	Private Resources Targets Low-power Status 1 (PRIVATE_TSTAT1)	32	R	0000_0000h
1ECh	Low-Power Hints 0 (LP_HINT0)	32	R	0000_0000h
1F0h	Low-Power Hints 1 (LP_HINT1)	32	R	0000_0000h
1F4h	Low-Power Request Deny Status 0 (LP_DENY0)	32	R	0000_0000h

30.4.1.2 Sleep Configuration (SLEEPCFG)

Offset

Register	Offset
SLEEPCFG	0h

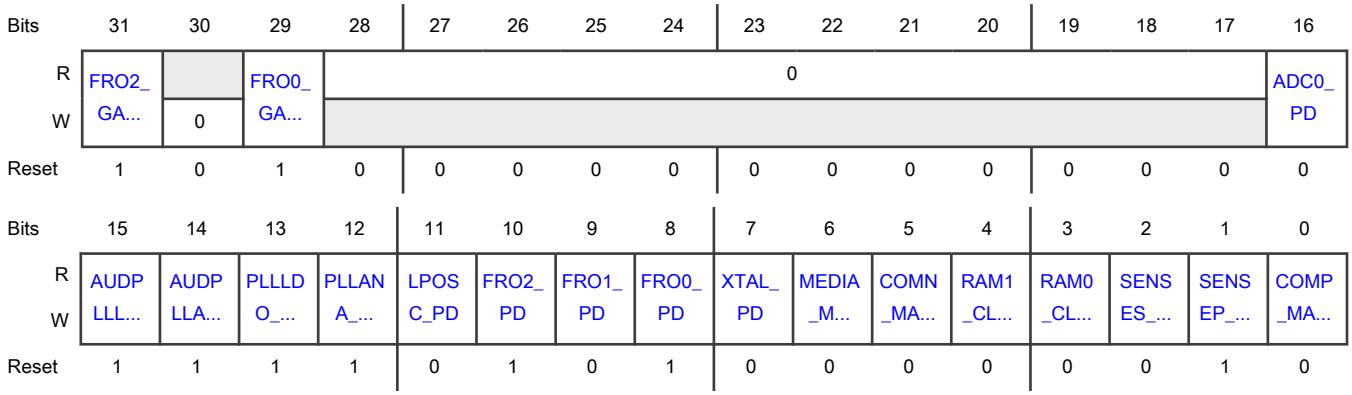
Function

Controls the power requests to various modules while CPU0 is in Deep Sleep mode. The power mode requests selected by this register are aggregated with the CPU1 requests to determine the actual settings. When the chip is in Deep Power Down mode or Full Deep Power Down mode, all the related modules are powered down.

NOTE

You need to clear FRO0_GATE and FRO2_GATE bits after POR and system reset.

Diagram



Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Controls the gating of FRO2. 0b - Removes gating 1b - Applies gating
30 —	Reserved
29 FRO0_GATE	FRO0 Gating Controls the gating of FRO0. 0b - Removes gating 1b - Applies gating
28-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Controls the power to ADC0. **NOTE** Write 1 to SHA_SEN_TCTRL0[ADC0_STOP] first and then write 1 to ADC0_PD to minimize the power consumption of ADC0. 0b - Powers on 1b - Powers down
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Controls the power to audio PLL internal regulator.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Powers on 1b - Powers down
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Controls the power to audio PLL analog function. 0b - Powers on 1b - Powers down
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Controls the power to main PLL internal regulator. 0b - Powers on 1b - Powers down
12 PLLANA_PD	Main PLL Analog Function Power Down Controls the power to main PLL analog function. 0b - Powers on 1b - Powers down
11 LPOSC_PD	LPOSC Power Down Controls the power to LPOSC. **NOTE** LPOSC is used by WWDT and RSTCTL. WWDT module can use MOD[LOCK] to lock SLEEPCON's control over power to the LPOSC. If MOD[LOCK] of either or both of WWDT0 and WWDT1 is configured as 1, the write to this field can change the filed value but does not take effect. 0b - Powers on 1b - Powers down
10 FRO2_PD	FRO2 Power Down Controls the power to FRO2. **NOTE** FRO2 power-down depends on both FRO2_PD and CSR[FROEN] of FRO2. To power down FRO2, you need to write 1 to FRO2_PD and write 0 to CSR[FROEN] of FRO2. To Power on FRO2, you need to write 0 to FRO2_PD or write 1 to CSR[FROEN] of FRO2. If SLEEPCON needs to wait power down ready from FRO2, then CSR[CLKGATE] of FRO2 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 FRO1_PD	FRO1 Power Down Controls the power to FRO1. **NOTE** FRO1 power-down depends on both FRO1_PD and CSR[FROEN] of FRO1. To power down FRO1, you need to write 1 to FRO1_PD and write 0 to CSR[FROEN] of FRO1. To Power on FRO1, you need to write 0 to FRO1_PD or write 1 to CSR[FROEN] of FRO1. If SLEEPCON needs to wait power down ready from FRO1, then CSR[CLKGATE] of FRO1 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
8 FRO0_PD	FRO0 Power Down Controls the power to FRO0. **NOTE** FRO0 power-down depends on both FRO0_PD and CSR[FROEN] of FRO0. To power down FRO0, you need to write 1 to FRO0_PD and write 0 to CSR[FROEN] of FRO0. To Power on FRO0, you need to write 0 to FRO0_PD or write 1 to CSR[FROEN] of FRO0. If SLEEPCON needs to wait power down ready from FRO0, then CSR[CLKGATE] of FRO0 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
7 XTAL_PD	Xtal Oscillator Power Down Controls the power to Xtal oscillator (osc_out). 0b - Powers on 1b - Powers down
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Controls the main clock shut-off of the VDD2_MEDIA and VDDN_MEDIA domains. 0b - Turns on 1b - Turns off
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Controls the main clock shut-off of the VDDN_COM domain. 0b - Turns on 1b - Turns off

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER1 in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER0 in VDD2_COM domain. 0b - Turns on 1b - Turns off
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Controls the main clock shut-off of the private part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
0 COMP_MAINCLK_SHUTOFF	VDD2_COMP Domain compute_main_clk Shut Off Controls the main clock shut-off of the VDD2_COMP domain. 0b - Turns on 1b - Turns off

30.4.1.3 Run Configuration (RUNCFG)

Offset

Register	Offset
RUNCFG	10h

Function

Controls the power to various modules during normal operation. The power mode requests selected by this register are aggregated with the CPU1 requests to determine the actual settings.

To modify a RUNCFG bit, use RUNCFG_CLR and RUNCFG_SET:

- To make it 0, write a 1 to the corresponding bit position in RUNCFG_CLR.
- To make it 1, write a 1 to the corresponding bit position in RUNCFG_SET.

Using RUNCFG_CLR and RUNCFG_SET prevents unintentionally setting or clearing other bits and avoids the possibility of an interrupt changing the value of RUNCFG in the middle of a read-modify-write sequence. It also avoids accidentally changing bits that might have been modified by other software.

NOTE

After confirming that a specific FRO*/LPOSC/Xtal/PLLs is not used, you can configure FRO*_PD and *PLL*_PD bits in RUNCFG by RUNCFG_CLR. Then FRO*/LPOSC/Xtal/PLLs can power down on the fly.

You need to clear FRO0_GATE and FRO2_GATE bits after POR and system reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FRO2_GA...		FRO0_GA...	0												ADC0_PD
W		0														
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUDPLLL...	AUDPLLA...	PLLLDO_...	PLLANA_...	LPOSC_PD	FRO2_PD	FRO1_PD	FRO0_PD	XTAL_PD	MEDIA_M...	COMN_MA...	RAM1_CL...	RAM0_CL...	SENSES_...	SENSEP_...	COMP_MA...
W																
Reset	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0

Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Controls the gating of FRO2. 0b - Removes gating 1b - Applies gating
30 —	Reserved
29 FRO0_GATE	FRO0 Gating Controls the gating of FRO0. 0b - Removes gating 1b - Applies gating
28-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Controls the power to ADC0. Write 1 to SHA_SEN_TCTRL0[ADC0_STOP] first and then write 1 to ADC0_PD to minimize the power consumption of ADC0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Powers on 1b - Powers down
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Controls the power to audio PLL internal regulator. 0b - Powers on 1b - Powers down
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Controls the power to audio PLL analog function. 0b - Powers on 1b - Powers down
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Controls the power to main PLL internal regulator. 0b - Powers on 1b - Powers down
12 PLLANA_PD	Main PLL Analog Function Power Down Controls the power to main PLL analog function. 0b - Powers on 1b - Powers down
11 LPOSC_PD	LPOSC Power Down Controls the power to LPOSC. **NOTE** LPOSC is used by WWDT and RSTCTL. WWDT module can use MOD[LOCK] to lock SLEEPCON's control over power to the LPOSC. If MOD[LOCK] of either or both of WWDT0 and WWDT1 is configured as 1, the write to this field can change the filed value but does not take effect. 0b - Powers on 1b - Powers down
10 FRO2_PD	FRO2 Power Down Controls the power to FRO2.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE FRO2 power-down depends on both FRO2_PD and CSR[FROEN] of FRO2. To power down FRO2, you need to write 1 to FRO2_PD and write 0 to CSR[FROEN] of FRO2. To Power on FRO2, you need to write 0 to FRO2_PD or write 1 to CSR[FROEN] of FRO2. If SLEEPCON needs to wait power down ready from FRO2, then CSR[CLKGATE] of FRO2 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
9 FRO1_PD	FRO1 Power Down Controls the power to FRO1. **NOTE** FRO1 power-down depends on both FRO1_PD and CSR[FROEN] of FRO1. To power down FRO1, you need to write 1 to FRO1_PD and write 0 to CSR[FROEN] of FRO1. To Power on FRO1, you need to write 0 to FRO1_PD or write 1 to CSR[FROEN] of FRO1. If SLEEPCON needs to wait power down ready from FRO1, then CSR[CLKGATE] of FRO1 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
8 FRO0_PD	FRO0 Power Down Controls the power to FRO0. **NOTE** FRO0 power-down depends on both FRO0_PD and CSR[FROEN] of FRO0. To power down FRO0, you need to write 1 to FRO0_PD and write 0 to CSR[FROEN] of FRO0. To Power on FRO0, you need to write 0 to FRO0_PD or write 1 to CSR[FROEN] of FRO0. If SLEEPCON needs to wait power down ready from FRO0, then CSR[CLKGATE] of FRO0 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
7 XTAL_PD	Xtal Oscillator Power Down Controls the power to Xtal oscillator (osc_out). 0b - Powers on 1b - Powers down
6	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEDIA_MAINCLK_SHUTOFF	Controls the main clock shut-off of the VDD2_MEDIA and VDDN_MEDIA domains. 0b - Turns on 1b - Turns off
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Controls the main clock shut-off of the VDDN_COM domain. 0b - Turns on 1b - Turns off
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk shut off Controls the clock shut-off of the RAM_ARBITER1 in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER0 in VDD2_COM domain. 0b - Turns on 1b - Turns off
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Controls the main clock shut-off of the private part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
0 COMP_MAINCLK_SHUTOFF	VDD2_COMP Domain compute_main_clk Shut Off Controls the main clock shut-off of the VDD2_COMP domain. 0b - Turns on 1b - Turns off

30.4.1.4 RUNCFG Set (RUNCFG_SET)

Offset

Register	Offset
RUNCFG_SET	20h

Function

Writing a 1 to a bit position in this register sets the corresponding position in RUNCFG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	FRO2_GA...	0	FRO0_GA...	0												ADC0_PD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	AUDPLLL...	AUDPLLA...	PLLLDO_...	PLLANA_...	LPOSC_PD	FRO2_PD	FRO1_PD	FRO0_PD	XTAL_PD	MEDIA_M...	COMN_MA...	RAM1_CL...	RAM0_CL...	SENSES_...	SENSEP_...	COMP_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30 —	Reserved
29 FRO0_GATE	FRO0 Gating Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28-17 —	Reserved
16	ADC0 Power Down Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
ADC0_PD	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12 PLLANA_PD	Main PLL Analog Function Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 LPOSC_PD	LPOSC Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 FRO2_PD	FRO2 Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 FRO1_PD	FRO1 Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 FRO0_PD	FRO0 Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7 XTAL_PD	Xtal Oscillator Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Set Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - No effect 1b - Sets the bit
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets the bit
0 COMP_MAINCLK_SHUTOFF	VDD2_COMP Domain compute_main_clk Shut off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.1.5 RUNCFG Clear (RUNCFG_CLR)

Offset

Register	Offset
RUNCFG_CLR	30h

Function

Writing a 1 to a bit in this register clears the corresponding bit in the RUNCFG register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	FRO2_GA...	0	FRO0_GA...	0												ADC0_PD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	AUDPLLL...	AUDPLLA...	PLLLDO_...	PLLANA_...	LPOSC_PD	FRO2_PD	FRO1_PD	FRO0_PD	XTAL_PD	MEDIA_M...	COMN_MA...	RAM1_CL...	RAM0_CL...	SENSES_...	SENSEP_...	COMP_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 —	Reserved
29 FRO0_GATE	FRO0 Gating Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12 PLLANA_PD	Main PLL Analog Function Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 LPOSC_PD	LPOSC Power Down Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
10 FRO2_PD	FRO2 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 FRO1_PD	FRO1 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 FRO0_PD	FRO0 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 XTAL_PD	Xtal Oscillator Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3	RAM_ARBITER0 common_ram_clk Shut Off Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
RAM0_CLK_SHUTOFF	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Clear Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - No effect 1b - Clears the bit
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 COMP_MAINCLK_SHUTOFF	VDD2_COMP Domain compute_main_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.1.6 Wakeup Enable 0 (WAKEUPEN0)

Offset

Register	Offset
WAKEUPEN0	60h

Function

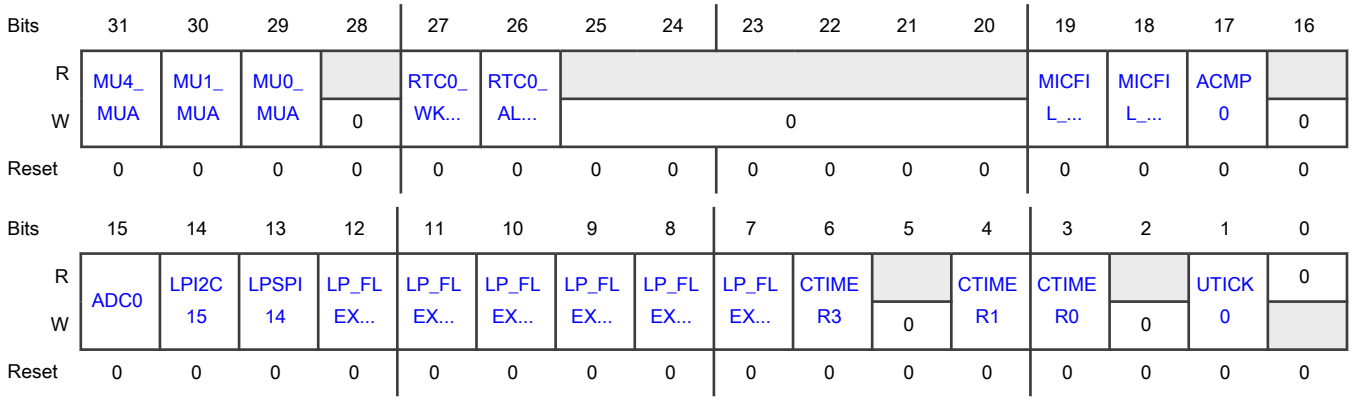
Enables the CPU0 to wake-up from deep sleep mode. If a module's wakeup field is enabled, the CPU0 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN0 bit, use the corresponding WAKEUPEN0_CLR and WAKEUPEN0_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN0_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN0_SET.

Using WAKEUPEN0_CLR and WAKEUPEN0_SET prevents unintentionally setting or clearing other bits

Diagram



Fields

Field	Function
31 MU4_MUA	MU4_MUA Wake-up Enable Enables CPU0 to wake-up on MU4_MUA interrupt. 0b - Disables 1b - Enables
30 MU1_MUA	MU1_MUA Wake-up Enable Enables CPU0 to wake-up on MU1_MUA interrupt. 0b - Disables 1b - Enables
29 MU0_MUA	MU0_MUA Wake-up Enable Enables CPU0 to wake-up on MU0_MUA interrupt. 0b - Disables 1b - Enables
28 —	Reserved
27 RTC0_WKUP	RTC0_WKUP Wake-up Enable Enables CPU0 to wake-up on RTC0 wakeup interrupt to VDD2_COMP domain. 0b - Disables 1b - Enables
26 RTC0_ALARM	RTC0_ALARM Wake-up Enable Enables CPU0 to wake-up on RTC0 alarm interrupt to VDD2_COMP domain. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
25-20 —	Reserved
19 MICFIL_WKUP	MICFIL Wake-up Enable Enables CPU0 to wake-up on MICFIL hardware voice activity detector interrupt or error interrupt. 0b - Disables 1b - Enables
18 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Enables CPU0 to wake-up on MICFIL interrupt for read data or error. 0b - Disables 1b - Enables
17 ACMP0	ACMP0 Wake-up Enable Enables CPU0 to wake-up on ACMP0 interrupt. 0b - Disables 1b - Enables
16 —	Reserved
15 ADC0	ADC0 Wake-up Enable Enables CPU0 to wake-up on ADC0 interrupt. 0b - Disables 1b - Enables
14 LPI2C15	LPI2C15 Wake-up Enable Enables CPU0 to wake-up on LPI2C15 interrupt. 0b - Disables 1b - Enables
13 LPSPI14	LPSPI14 Wake-up Enable Enables CPU0 to wake-up on LPSPI14 interrupt. 0b - Disables 1b - Enables
12 LP_FLEXCOMM5	LP_FLEXCOMM5 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM5 interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
11 LP_FLEXCOMM4	LP_FLEXCOMM4 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM4 interrupt. 0b - Disables 1b - Enables
10 LP_FLEXCOMM3	LP_FLEXCOMM3 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM3 interrupt. 0b - Disables 1b - Enables
9 LP_FLEXCOMM2	LP_FLEXCOMM2 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM2 interrupt. 0b - Disables 1b - Enables
8 LP_FLEXCOMM1	LP_FLEXCOMM1 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM1 interrupt. 0b - Disables 1b - Enables
7 LP_FLEXCOMM0	LP_FLEXCOMM0 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM0 interrupt. 0b - Disables 1b - Enables
6 CTIMER3	CTIMER3 Wake-up Enable Enables CPU0 to wake-up on CTIMER3 interrupt. 0b - Disables 1b - Enables
5 —	Reserved
4 CTIMER1	CTIMER1 Wake-up Enable Enables CPU0 to wake-up on CTIMER1 interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
3 CTIMER0	CTIMER0 Wake-up Enable Enables CPU0 to wake-up on CTIMER0 interrupt. 0b - Disables 1b - Enables
2 —	Reserved
1 UTICK0	UTICK0 Wake-up Enable Enables CPU0 to wake-up on UTICK0 interrupt 0b - Disables 1b - Enables
0 —	Reserved

30.4.1.7 Wakeup Enable 1 (WAKEUPEN1)

Offset

Register	Offset
WAKEUPEN1	64h

Function

Enables the CPU0 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU0 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN1 bit, use the corresponding WAKEUPEN1_CLR and WAKEUPEN1_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN1_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN1_SET.

Using WAKEUPEN1_CLR and WAKEUPEN1_SET prevents unintentionally setting or clearing other bits

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	0			FLEXIO	I3C1	LPSPI16	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LP_FLEX...	PMIC_IR...	0	USBPHY0	WWDT1	WWDT0	USB1	USB0	I3C0	0		LP_FLEX...	LP_FLEX...	OSTIMER	CTIMER4	CTIMER2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 EDMA0_CH4	eDMA0 Channel 4 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 4 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
30 EDMA0_CH3	eDMA0 Channel 3 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 3 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
29 EDMA0_CH2	eDMA0 Channel 2 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 2 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
28 EDMA0_CH1	eDMA0 Channel 1 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 1 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
27 EDMA0_CH0	eDMA0 Channel 0 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 0 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
26-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
23 FLEXIO	FLEXIO Wake-up Enable Enables CPU0 to wake-up on FLEXIO interrupt. 0b - Disables 1b - Enables
22 I3C1	I3C1 Wake-up Enable Enables CPU0 to wake-up on I3C1 interrupt. 0b - Disables 1b - Enables
21 LPSPI16	LPSPI16 Wake-up Enable Enables CPU0 to wake-up on LPSPI16 interrupt. 0b - Disables 1b - Enables
20 LP_FLEXCOMM13	LP_FLEXCOMM13 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM13 interrupt. 0b - Disables 1b - Enables
19 LP_FLEXCOMM12	LP_FLEXCOMM12 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM12 interrupt. 0b - Disables 1b - Enables
18 LP_FLEXCOMM11	LP_FLEXCOMM11 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM11 interrupt. 0b - Disables 1b - Enables
17 LP_FLEXCOMM10	LP_FLEXCOMM10 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM10 interrupt. 0b - Disables 1b - Enables
16 LP_FLEXCOMM9	LP_FLEXCOMM9 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM9 interrupt. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
15 LP_FLEXCOMM8	LP_FLEXCOMM8 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM8 interrupt. 0b - Disables 1b - Enables
14 PMIC_IRQN	PMIC_IRQN Wake-up Enable Enables CPU0 to wake-up on external PMIC interrupt. 0b - Disables 1b - Enables
13 —	Reserved
12 USBPHY0	USBPHY0 Wake-up Enable Enables CPU0 to wake-up on USBPHY0 interrupt. 0b - Disables 1b - Enables
11 WWDT1	WWDT1 Wake-up Enable Enables CPU0 to wake-up on WWDT1 interrupt. 0b - Disables 1b - Enables
10 WWDT0	WWDT0 Wake-up Enable Enables CPU0 to wake-up on WWDT0 interrupt. 0b - Disables 1b - Enables
9 USB1	USB1 Wake-up Enable Enables CPU0 to wake-up on USB1 interrupt. 0b - Disables 1b - Enables
8 USB0	USB0 Wake-up Enable Enables CPU0 to wake-up on USB0 interrupt. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 I3C0	I3C0 Wake-up Enable Enables CPU0 to wake-up on I3C0 interrupt. 0b - Disables 1b - Enables
6-5 —	Reserved
4 LP_FLEXCOMM7	LP_FLEXCOMM7 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM7 interrupt. 0b - Disables 1b - Enables
3 LP_FLEXCOMM6	LP_FLEXCOMM6 Wake-up Enable Enables CPU0 to wake-up on LP_FLEXCOMM6 interrupt. 0b - Disables 1b - Enables
2 OSTIMER	OSTIMER Wake-up Enable Enables CPU0 to wake-up on OSTIMER interrupt. 0b - Disables 1b - Enables
1 CTIMER4	CTIMER4 Wake-up Enable Enables CPU0 to wake-up on CTIMER4 interrupt. 0b - Disables 1b - Enables
0 CTIMER2	CTIMER2 Wake-up Enable Enables CPU0 to wake-up on CTIMER2 interrupt. 0b - Disables 1b - Enables

30.4.1.8 Wakeup Enable 2 (WAKEUPEN2)

Offset

Register	Offset
WAKEUPEN2	68h

Function

Enables the CPU0 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU0 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN2 bit, use the corresponding WAKEUPEN2_CLR and WAKEUPEN2_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN2_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN2_SET.

Using WAKEUPEN2_CLR and WAKEUPEN2_SET prevents unintentionally setting or clearing other bits

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPIO2_C...	GPIO1_C...	GPIO1_C...	GPIO0_C...	GPIO0_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO2_CH0	GPIO2 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO2 channel 0 interrupt. 0b - Disables 1b - Enables
30 GPIO1_CH1	GPIO1 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO1 channel 1 interrupt. 0b - Disables 1b - Enables
29 GPIO1_CH0	GPIO1 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO1 channel 0 interrupt. 0b - Disables 1b - Enables
28 GPIO0_CH1	GPIO0 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO0 channel 1 interrupt. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
27 GPIO0_CH0	GPIO0 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO0 channel 0 interrupt. 0b - Disables 1b - Enables
26 EDMA1_CH15	eDMA1 Channel 15 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 15 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
25 EDMA1_CH14	eDMA1 Channel 14 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 14 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
24 EDMA1_CH13	eDMA1 Channel 13 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 13 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
23 EDMA1_CH12	eDMA1 Channel 12 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 12 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
22 EDMA1_CH11	eDMA1 Channel 11 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 11 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
21 EDMA1_CH10	eDMA1 Channel 10 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 10 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
20 EDMA1_CH9	eDMA1 Channel 9 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 9 interrupt only when DMA wakeup is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
19 EDMA1_CH8	eDMA1 Channel 8 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 8 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
18 EDMA1_CH7	eDMA1 Channel 7 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 7 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
17 EDMA1_CH6	eDMA1 Channel 6 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 6 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
16 EDMA1_CH5	eDMA1 Channel 5 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 5 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
15 EDMA1_CH4	eDMA1 Channel 4 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 4 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
14 EDMA1_CH3	eDMA1 Channel 3 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 3 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
13 EDMA1_CH2	eDMA1 Channel 2 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 2 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
12	eDMA1 Channel 1 Wake-up Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA1_CH1	Enables CPU0 to wake-up on eDMA1 channel 1 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
11 EDMA1_CH0	eDMA1 Channel 0 Wake-up Enable Enables CPU0 to wake-up on eDMA1 channel 0 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
10 EDMA0_CH15	eDMA0 Channel 15 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 15 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
9 EDMA0_CH14	eDMA0 Channel 14 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 14 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
8 EDMA0_CH13	eDMA0 Channel 13 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 13 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
7 EDMA0_CH12	eDMA0 Channel 12 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 12 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
6 EDMA0_CH11	eDMA0 Channel 11 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 11 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
5 EDMA0_CH10	eDMA0 Channel 10 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 10 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 EDMA0_CH9	eDMA0 Channel 9 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 9 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
3 EDMA0_CH8	eDMA0 Channel 8 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 8 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
2 EDMA0_CH7	eDMA0 Channel 7 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 7 interrupt. 0b - Disables 1b - Enables
1 EDMA0_CH6	eDMA0 Channel 6 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 6 interrupt. 0b - Disables 1b - Enables
0 EDMA0_CH5	eDMA0 Channel 5 Wake-up Enable Enables CPU0 to wake-up on eDMA0 channel 5 interrupt. 0b - Disables 1b - Enables

30.4.1.9 Wakeup Enable 3 (WAKEUPEN3)

Offset

Register	Offset
WAKEUPEN3	6Ch

Function

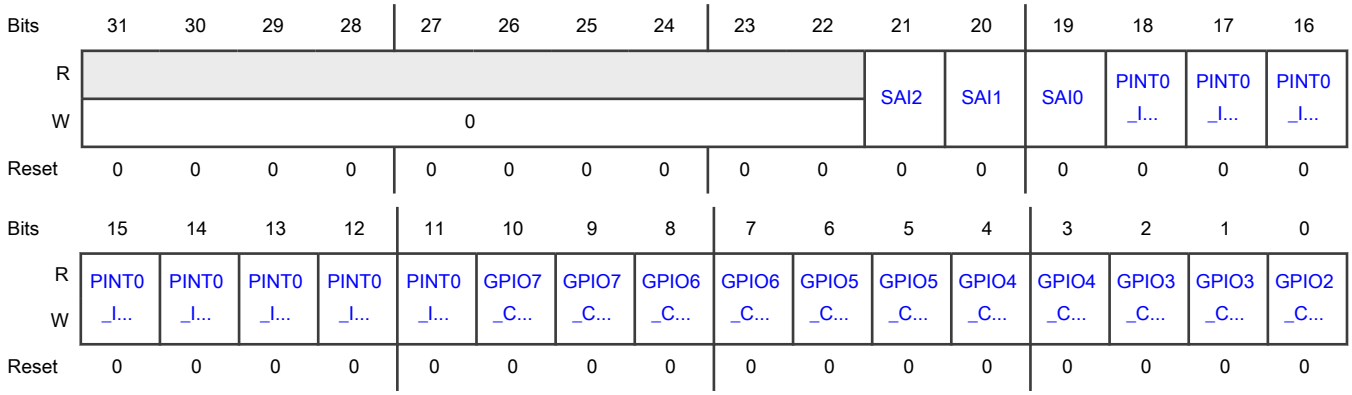
Enables the CPU0 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU0 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN bit, use the corresponding WAKEUPEN3_CLR and WAKEUPEN3_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN3_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN3_SET.

Using WAKEUPEN3_CLR and WAKEUPEN3_SET prevents unintentionally setting or clearing other bits

Diagram



Fields

Field	Function
31-22 —	Reserved
21 SAI2	SAI2 Wake-up Enable Enables CPU0 to wake-up on SAI2 interrupt. 0b - Disables 1b - Enables
20 SAI1	SAI1 Wake-up Enable Enables CPU0 to wake-up on SAI1 interrupt. 0b - Disables 1b - Enables
19 SAI0	SAI0 Wake-up Enable Enables CPU0 to wake-up on SAI0 interrupt. 0b - Disables 1b - Enables
18 PINT0_INTR7	PINT0_INTR7 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 7. 0b - Disables 1b - Enables
17 PINT0_INTR6	PINT0_INTR6 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 6. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 PINT0_INTR5	PINT0_INTR5 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 5. 0b - Disables 1b - Enables
15 PINT0_INTR4	PINT0_INTR4 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 4. 0b - Disables 1b - Enables
14 PINT0_INTR3	PINT0_INTR3 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 3. 0b - Disables 1b - Enables
13 PINT0_INTR2	PINT0_INTR2 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 2. 0b - Disables 1b - Enables
12 PINT0_INTR1	PINT0_INTR1 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 1. 0b - Disables 1b - Enables
11 PINT0_INTR0	PINT0_INTR0 Wake-up Enable Enables CPU0 to wake-up on PINT0 interrupt 0. 0b - Disables 1b - Enables
10 GPIO7_CH1	GPIO7 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO7 channel 1 interrupt. 0b - Disables 1b - Enables
9 GPIO7_CH0	GPIO7 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO7 channel 0 interrupt. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
8 GPIO6_CH1	GPIO6 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO6 channel 1 interrupt. 0b - Disables 1b - Enables
7 GPIO6_CH0	GPIO6 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO6 channel 0 interrupt. 0b - Disables 1b - Enables
6 GPIO5_CH1	GPIO5 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO5 channel 1 interrupt. 0b - Disables 1b - Enables
5 GPIO5_CH0	GPIO5 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO5 channel 0 interrupt. 0b - Disables 1b - Enables
4 GPIO4_CH1	GPIO4 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO4 channel 1 interrupt. 0b - Disables 1b - Enables
3 GPIO4_CH0	GPIO4 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO4 channel 0 interrupt. 0b - Disables 1b - Enables
2 GPIO3_CH1	GPIO3 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO3 channel 1 interrupt. 0b - Disables 1b - Enables
1 GPIO3_CH0	GPIO3 Channel 0 Wake-up Enable Enables CPU0 to wake-up on GPIO3 channel 0 interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
0 GPIO2_CH1	GPIO2 Channel 1 Wake-up Enable Enables CPU0 to wake-up on GPIO2 channel 1 interrupt. 0b - Disables 1b - Enables

30.4.1.10 Wakeup Enable 4 (WAKEUPEN4)

Offset

Register	Offset
WAKEUPEN4	70h

Function

Enables the CPU0 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU0 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN bit, use the corresponding WAKEUPEN_CLR and WAKEUPEN_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN_SET.

Using WAKEUPEN_CLR and WAKEUPEN_SET prevents unintentionally setting or clearing other bits

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FRO2	FRO1	FRO0										
W			0				0									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R						PMC0										
W	0						0	0	0			0	0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 FRO2	FRO2 Wake-up Enable Enables CPU0 to wake-up on FRO2 Async interrupt. 0b - Disables 1b - Enables
27 FRO1	FRO1 Wake-up Enable Enables CPU0 to wake-up on FRO1 Async interrupt. 0b - Disables 1b - Enables
26 FRO0	FRO0 Wake-up Enable Enables CPU0 to wake-up on FRO0 Async interrupt. 0b - Disables 1b - Enables
25-11 —	Reserved
10 PMC0	PMC0 Wake-up Enable Enables CPU0 to wake-up on PMC0 interrupt. 0b - Disables 1b - Enables
9 —	Reserved
8 —	Reserved
7-5 —	Reserved
4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
3-0 —	Reserved

30.4.1.11 Wakeup Enable 0 Set (WAKEUPEN0_SET)

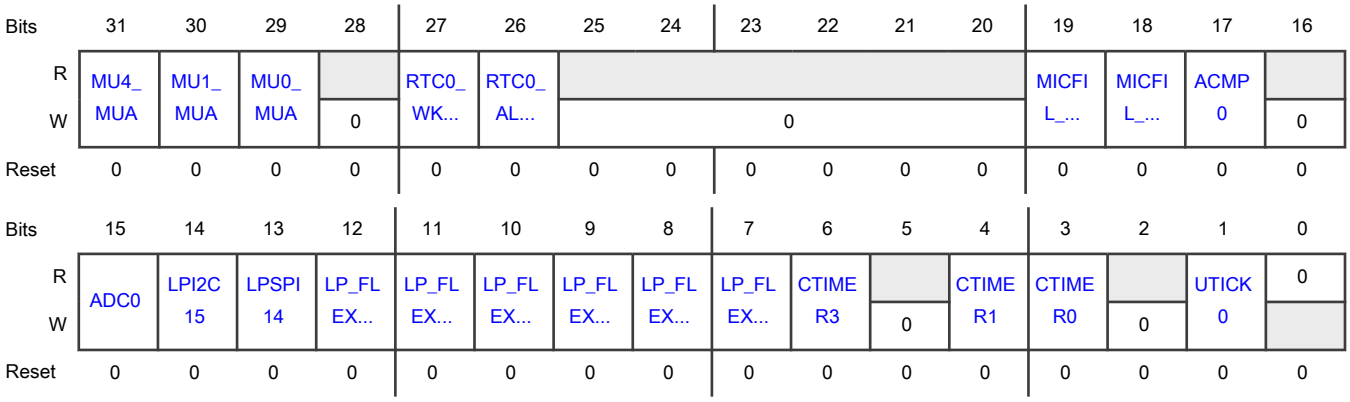
Offset

Register	Offset
WAKEUPEN0_SET	A0h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN0. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31 MU4_MUA	MU4_MUA Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30 MU1_MUA	MU1_MUA Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
29 MU0_MUA	MU0_MUA Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28 —	Reserved
27 RTC0_WKUP	RTC0_WKUP Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
26 RTC0_ALARM	RTC0_ALARM Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
25-20 —	Reserved
19 MICFIL_WKUP	MICFIL Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 ACMP0	ACMP0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16 —	Reserved
15	ADC0 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
ADC0	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 LPI2C15	LPI2C15 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 LPSPI14	LPSPI14 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12 LP_FLEXCOMM5	LP_FLEXCOMM5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 LP_FLEXCOMM4	LP_FLEXCOMM4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 LP_FLEXCOMM3	LP_FLEXCOMM3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 LP_FLEXCOMM2	LP_FLEXCOMM2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
8 LP_FLEXCOMM1	LP_FLEXCOMM1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 LP_FLEXCOMM0	LP_FLEXCOMM0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 CTIMER3	CTIMER3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 —	Reserved
4 CTIMER1	CTIMER1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 CTIMER0	CTIMER0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 —	Reserved
1 UTICK0	UTICK0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.1.12 Wakeup Enable 1 Set (WAKEUPEN1_SET)

Offset

Register	Offset
WAKEUPEN1_SET	A4h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN1. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	0			FLEXIO	I3C1	LPSPI16	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LP_FLEX...	PMIC_IR...		USBPHY0	WWDT1	WWDT0	USB1	USB0	I3C0	0		LP_FLEX...	LP_FLEX...	OSTIMER	CTIMER4	CTIMER2
W			0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 EDMA0_CH4	eDMA0 Channel 4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30 EDMA0_CH3	eDMA0 Channel 3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
29 EDMA0_CH2	eDMA0 Channel 2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28 EDMA0_CH1	eDMA0 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
27 EDMA0_CH0	eDMA0 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
26-24 —	Reserved
23 FLEXIO	FLEXIO Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 I3C1	I3C1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21 LPSPI16	LPSPI16 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20 LP_FLEXCOMM13	LP_FLEXCOMM13 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19 LP_FLEXCOMM12	LP_FLEXCOMM12 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 LP_FLEXCOMM11	LP_FLEXCOMM11 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 LP_FLEXCOMM10	LP_FLEXCOMM10 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16	LP_FLEXCOMM9 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM9	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 LP_FLEXCOMM8	LP_FLEXCOMM8 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 PMIC_IRQN	PMIC_IRQN Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 —	Reserved
12 USBPHY0	USBPHY0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 WWDT1	WWDT1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 WWDT0	WWDT0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 USB1	USB1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
8 USB0	USB0 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
7 I3C0	I3C0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6-5 —	Reserved
4 LP_FLEXCOMM7	LP_FLEXCOMM7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 LP_FLEXCOMM6	LP_FLEXCOMM6 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 OSTIMER	OSTIMER Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 CTIMER4	CTIMER4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 CTIMER2	CTIMER2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.1.13 Wakeup Enable 2 Set (WAKEUPEN2_SET)

Offset

Register	Offset
WAKEUPEN2_SET	A8h

Function
Writing a 1 to a bit position sets the corresponding position in WAKEUPEN2. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPIO2_C...	GPIO1_C...	GPIO1_C...	GPIO0_C...	GPIO0_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO2_CH0	GPIO2 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30 GPIO1_CH1	GPIO1 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
29 GPIO1_CH0	GPIO1 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28 GPIO0_CH1	GPIO0 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
27 GPIO0_CH0	GPIO0 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
26 EDMA1_CH15	eDMA1 Channel 15 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
25 EDMA1_CH14	eDMA1 Channel 14 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
24 EDMA1_CH13	eDMA1 Channel 13 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
23 EDMA1_CH12	eDMA1 Channel 12 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 EDMA1_CH11	eDMA1 Channel 11 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21 EDMA1_CH10	eDMA1 Channel 10 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20	eDMA1 Channel 9 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA1_CH9	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19 EDMA1_CH8	eDMA1 Channel 8 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 EDMA1_CH7	eDMA1 Channel 7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 EDMA1_CH6	eDMA1 Channel 6 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16 EDMA1_CH5	eDMA1 Channel 5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 EDMA1_CH4	eDMA1 Channel 4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 EDMA1_CH3	eDMA1 Channel 3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 EDMA1_CH2	eDMA1 Channel 2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
12 EDMA1_CH1	eDMA1 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 EDMA1_CH0	eDMA1 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 EDMA0_CH15	eDMA0 Channel 15 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 EDMA0_CH14	eDMA0 Channel 14 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
8 EDMA0_CH13	eDMA0 Channel 13 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7 EDMA0_CH12	eDMA0 Channel 12 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 EDMA0_CH11	eDMA0 Channel 11 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 EDMA0_CH10	eDMA0 Channel 10 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets the bit
4 EDMA0_CH9	eDMA0 Channel 9 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 EDMA0_CH8	eDMA0 Channel 8 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 EDMA0_CH7	eDMA0 Channel 7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 EDMA0_CH6	eDMA0 Channel 6 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 EDMA0_CH5	eDMA0 Channel 5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

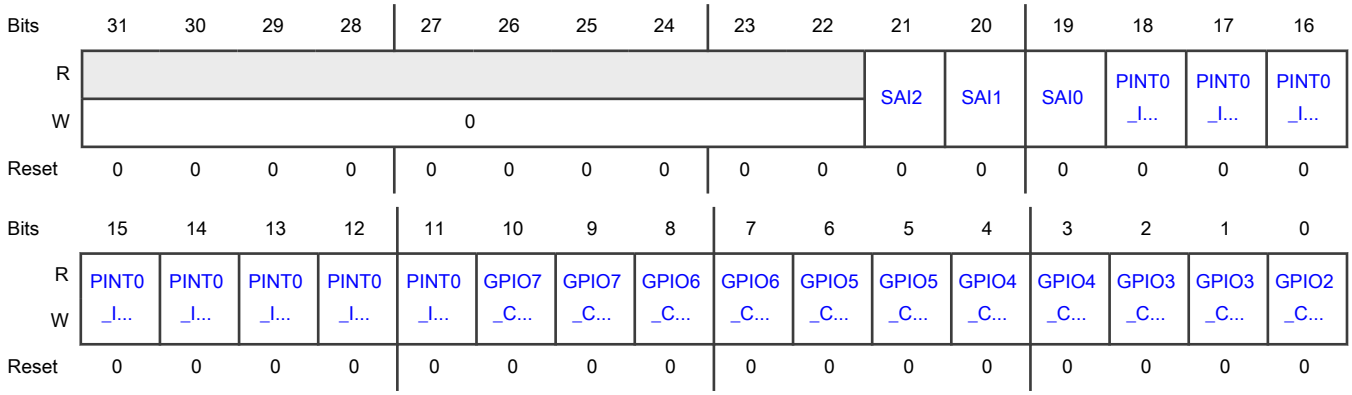
30.4.1.14 Wakeup Enable 3 Set (WAKEUPEN3_SET)

Offset

Register	Offset
WAKEUPEN3_SET	ACh

Function
Writing a 1 to a bit position sets the corresponding position in WAKEUPEN3. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-22 —	Reserved
21 SAI2	SAI2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20 SAI1	SAI1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19 SAI0	SAI0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 PINT0_INTR7	PINT0_INTR7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 PINT0_INTR6	PINT0_INTR6 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 PINT0_INTR5	PINT0_INTR5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 PINT0_INTR4	PINT0_INTR4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 PINT0_INTR3	PINT0_INTR3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 PINT0_INTR2	PINT0_INTR2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12 PINT0_INTR1	PINT0_INTR1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 PINT0_INTR0	PINT0_INTR0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 GPIO7_CH1	GPIO7 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 GPIO7_CH0	GPIO7 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets the bit
8 GPIO6_CH1	GPIO6 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7 GPIO6_CH0	GPIO6 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 GPIO5_CH1	GPIO5 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 GPIO5_CH0	GPIO5 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
4 GPIO4_CH1	GPIO4 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 GPIO4_CH0	GPIO4 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 GPIO3_CH1	GPIO3 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 GPIO3_CH0	GPIO3 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
0 GPIO2_CH1	GPIO2 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.1.15 Wakeup Enable 4 Set (WAKEUPEN4_SET)

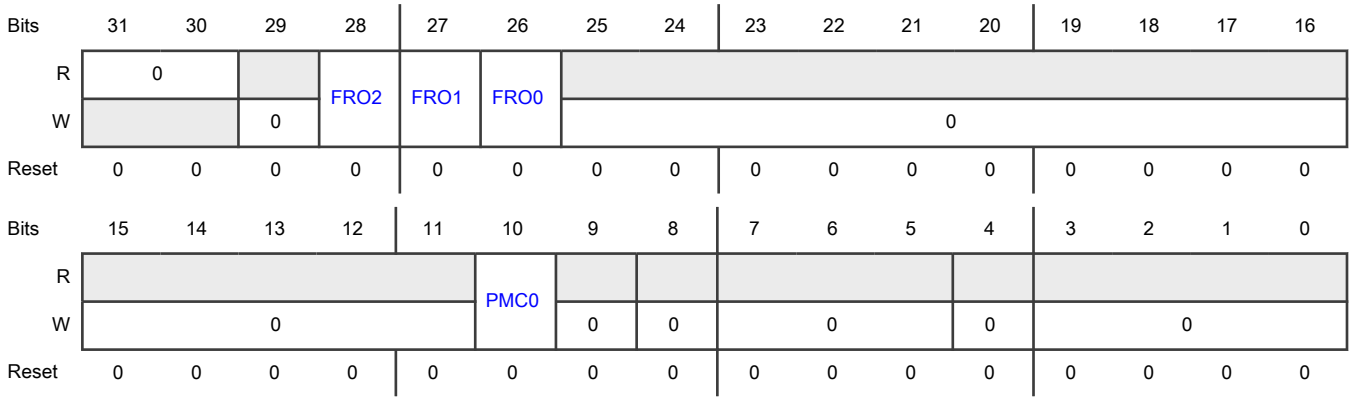
Offset

Register	Offset
WAKEUPEN4_SET	B0h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN4. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
28 FRO2	FRO2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
27 FRO1	FRO1 Wake-up Enable Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
26 FRO0	FRO0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
25-11 —	Reserved
10 PMC0	PMC0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 —	Reserved
8 —	Reserved
7-5 —	Reserved
4 —	Reserved
3-0 —	Reserved

30.4.1.16 Wakeup Enable 0 Clear (WAKEUPEN0_CLR)

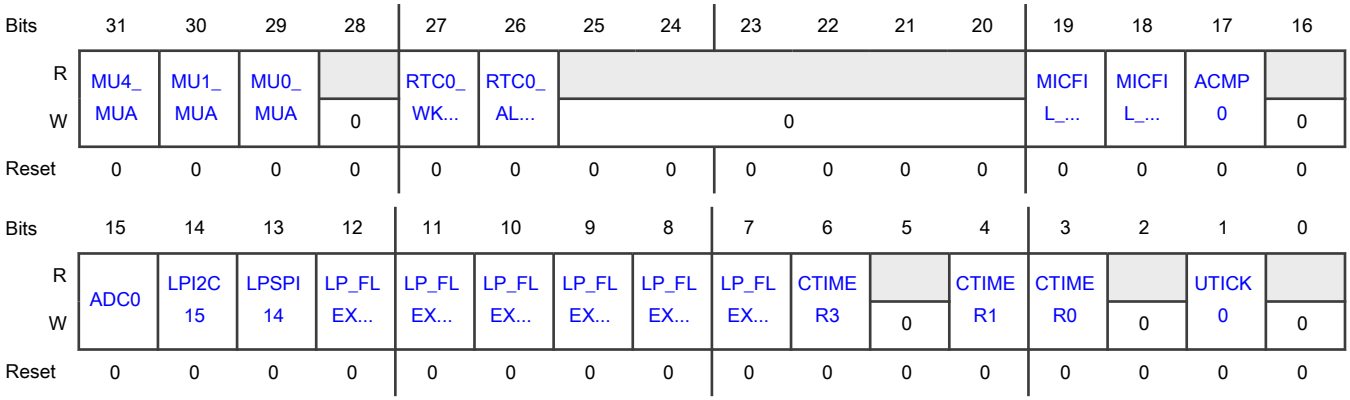
Offset

Register	Offset
WAKEUPEN0_CLR	100h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN0 register. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31 MU4_MUA	MU4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
30 MU1_MUA	MU1_MUA Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
29 MU0_MUA	MU0_MUA Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
27 RTC0_WKUP	RTC0_WKUP Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
26 RTC0_ALARM	RTC0_ALARM Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
25-20 —	Reserved
19 MICFIL_WKUP	MICFIL Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 ACMP0	ACMP0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16 —	Reserved
15 ADC0	ADC0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14	LPI2C15 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPI2C15	0b - No effect 1b - Clears the bit
13 LPSPI14	LPSPI14 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12 LP_FLEXCOMM5	LP_FLEXCOMM5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 LP_FLEXCOMM4	LP_FLEXCOMM4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 LP_FLEXCOMM3	LP_FLEXCOMM3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 LP_FLEXCOMM2	LP_FLEXCOMM2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 LP_FLEXCOMM1	LP_FLEXCOMM1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 LP_FLEXCOMM0	LP_FLEXCOMM0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6	CTIMER3 Wake-up Enable Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER3	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 —	Reserved
4 CTIMER1	CTIMER1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 CTIMER0	CTIMER0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 —	Reserved
1 UTICK0	UTICK0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.1.17 Wakeup Enable 1 Clear (WAKEUPEN1_CLR)

Offset

Register	Offset
WAKEUPEN1_CLR	104h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN1 register. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	0			FLEXIO	I3C1	LPSPI16	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LP_FLEX...	PMIC_IR...	0	USBPHY0	WWDT1	WWDT0	USB1	USB0	I3C0	0		LP_FLEX...	LP_FLEX...	OSTIMER	CTIMER4	CTIMER2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 EDMA0_CH4	eDMA0 Channel 4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
30 EDMA0_CH3	eDMA0 Channel 3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
29 EDMA0_CH2	eDMA0 Channel 2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28 EDMA0_CH1	eDMA0 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
27 EDMA0_CH0	eDMA0 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
26-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
23 FLEXIO	FLEXIO Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 I3C1	I3C1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21 LPSPI16	LPSPI16 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 LP_FLEXCOMM13	LP_FLEXCOMM13 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19 LP_FLEXCOMM12	LP_FLEXCOMM12 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 LP_FLEXCOMM11	LP_FLEXCOMM11 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 LP_FLEXCOMM10	LP_FLEXCOMM10 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16 LP_FLEXCOMM9	LP_FLEXCOMM9 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Clears the bit
15 LP_FLEXCOMM8	LP_FLEXCOMM8 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 PMIC_IRQN	PMIC_IRQN Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 —	Reserved
12 USBPHY0	USBPHY0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 WWDT1	WWDT1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 WWDT0	WWDT0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 USB1	USB1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 USB0	USB0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 I3C0	I3C0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6-5 —	Reserved
4 LP_FLEXCOMM7	LP_FLEXCOMM7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 LP_FLEXCOMM6	LP_FLEXCOMM6 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 OSTIMER	OSTIMER Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 CTIMER4	CTIMER4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 CTIMER2	CTIMER2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.1.18 Wakeup Enable 2 Clear (WAKEUPEN2_CLR)

Offset

Register	Offset
WAKEUPEN2_CLR	108h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN2 register. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPIO2_C...	GPIO1_C...	GPIO1_C...	GPIO0_C...	GPIO0_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA1_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...	EDMA0_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO2_CH0	GPIO2 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
30 GPIO1_CH1	GPIO1 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
29 GPIO1_CH0	GPIO1 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28 GPIO0_CH1	GPIO0 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
27 GPIO0_CH0	GPIO0 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Clears the bit
26 EDMA1_CH15	eDMA1 Channel 15 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
25 EDMA1_CH14	eDMA1 Channel 14 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
24 EDMA1_CH13	eDMA1 Channel 13 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
23 EDMA1_CH12	eDMA1 Channel 12 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 EDMA1_CH11	eDMA1 Channel 11 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21 EDMA1_CH10	eDMA1 Channel 10 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 EDMA1_CH9	eDMA1 Channel 9 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19 EDMA1_CH8	eDMA1 Channel 8 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
18 EDMA1_CH7	eDMA1 Channel 7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 EDMA1_CH6	eDMA1 Channel 6 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16 EDMA1_CH5	eDMA1 Channel 5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15 EDMA1_CH4	eDMA1 Channel 4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 EDMA1_CH3	eDMA1 Channel 3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 EDMA1_CH2	eDMA1 Channel 2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12 EDMA1_CH1	eDMA1 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11	eDMA1 Channel 0 Wake-up Enable Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA1_CH0	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 EDMA0_CH15	eDMA0 Channel 15 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 EDMA0_CH14	eDMA0 Channel 14 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 EDMA0_CH13	eDMA0 Channel 13 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 EDMA0_CH12	eDMA0 Channel 12 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 EDMA0_CH11	eDMA0 Channel 11 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 EDMA0_CH10	eDMA0 Channel 10 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 EDMA0_CH9	eDMA0 Channel 9 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
3 EDMA0_CH8	eDMA0 Channel 8 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 EDMA0_CH7	eDMA0 Channel 7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 EDMA0_CH6	eDMA0 Channel 6 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 EDMA0_CH5	eDMA0 Channel 5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.1.19 Wakeup Enable 3 Clear (WAKEUPEN3_CLR)

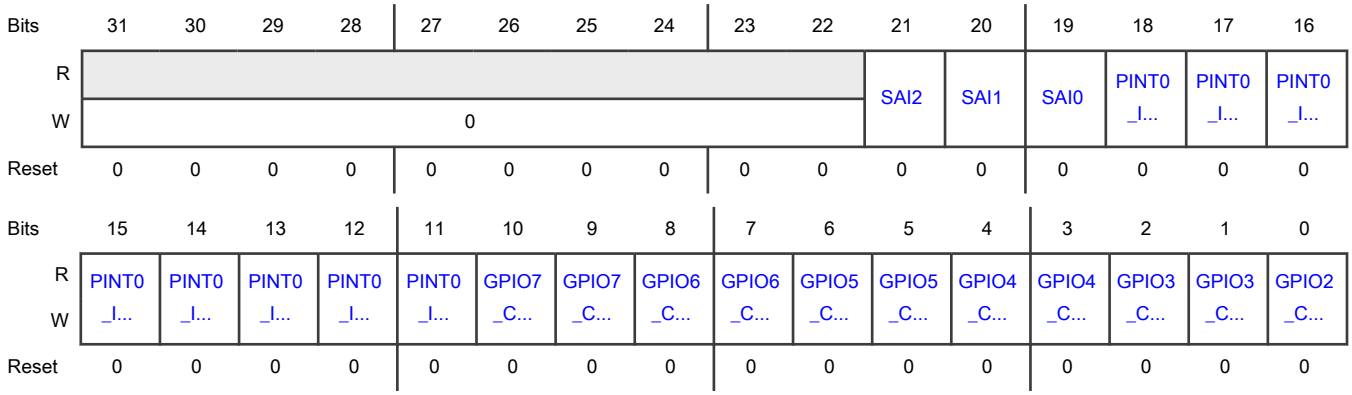
Offset

Register	Offset
WAKEUPEN3_CLR	10Ch

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN3 register. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-22 —	Reserved
21 SAI2	SAI2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 SAI1	SAI1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19 SAI0	SAI0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 PINT0_INTR7	PINT0_INTR7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 PINT0_INTR6	PINT0_INTR6 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 PINT0_INTR5	PINT0_INTR5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15 PINT0_INTR4	PINT0_INTR4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 PINT0_INTR3	PINT0_INTR3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 PINT0_INTR2	PINT0_INTR2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12 PINT0_INTR1	PINT0_INTR1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 PINT0_INTR0	PINT0_INTR0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 GPIO7_CH1	GPIO7 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 GPIO7_CH0	GPIO7 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Clears the bit
8 GPIO6_CH1	GPIO6 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 GPIO6_CH0	GPIO6 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 GPIO5_CH1	GPIO5 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 GPIO5_CH0	GPIO5 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 GPIO4_CH1	GPIO4 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 GPIO4_CH0	GPIO4 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 GPIO3_CH1	GPIO3 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 GPIO3_CH0	GPIO3 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
0 GPIO2_CH1	GPIO2 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.1.20 Wakeup Enable 4 Clear (WAKEUPEN4_CLR)

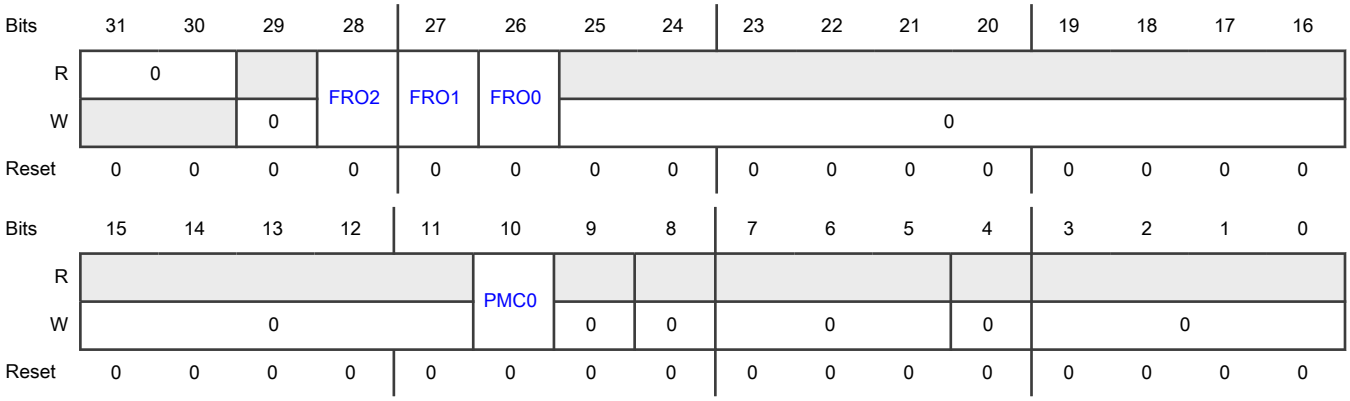
Offset

Register	Offset
WAKEUPEN4_CLR	110h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN4 register. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-30 —	Reserved
29	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
28 FRO2	FRO2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
27 FRO1	FRO1 Wake-up Enable Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
26 FRO0	FRO0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
25-11 —	Reserved
10 PMC0	PMC0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 —	Reserved
8 —	Reserved
7-5 —	Reserved
4 —	Reserved
3-0 —	Reserved

30.4.1.21 Power Down Wait (PWRDOWN_WAIT)

Offset

Register	Offset
PWRDOWN_WAIT	164h

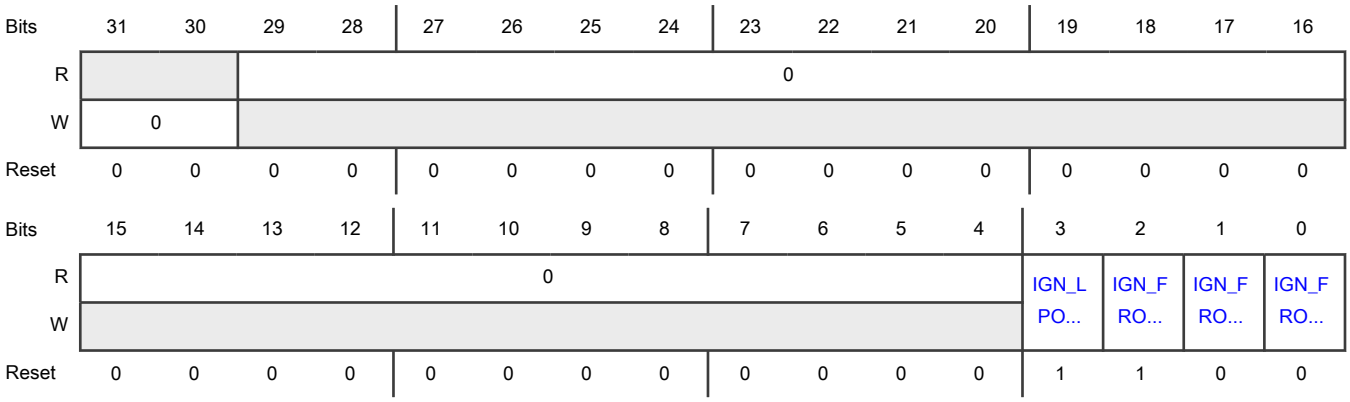
Function

Controls the power down wait of FRO0-2 and LPOSC. Setting the ignore bits below will allow the CPU0 to continue entering a low-power mode without waiting for the corresponding clock to shut down first.

NOTE

If SLEEPCON needs to wait power down ready from FRO, then CSR[CLKGATE] of FRO must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-4 —	Reserved
3 IGN_LPOSCPDR	Ignore LPOSC Power Down Ready Signals Ignores power down ready signal from LPOSC 0b - Waits 1b - Ignores
2 IGN_FRO2PDR	Ignore FRO2 Power Down Ready Signals Ignores power down ready signal from FRO2

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Waits 1b - Ignores
1 IGN_FRO1PDR	Ignore FRO1 Power Down Ready Signals Ignores power down ready signal from FRO1 0b - Waits 1b - Ignores
0 IGN_FRO0PDR	Ignore FRO0 Power Down Ready Signal Ignores power down ready signal from FRO0 0b - Waits 1b - Ignores

30.4.1.22 eDMA Wakeup Enable (HW_WAKE)

Offset

Register	Offset
HW_WAKE	190h

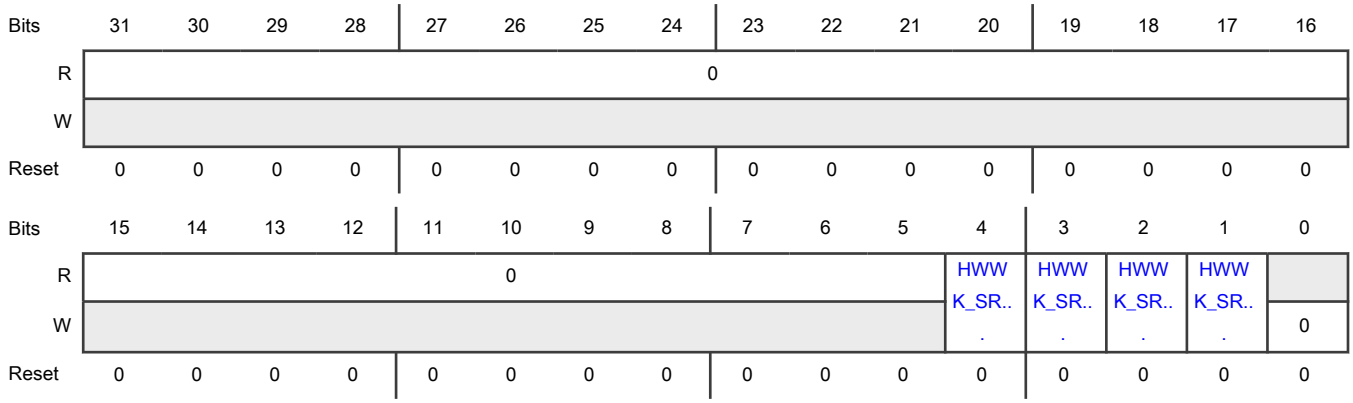
Function

Enables eDMA (eDMA0 and eDMA1) wakeup sources. To modify an HW_WAKE bit, use the HW_WAKE_CLR and HW_WAKE_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in HW_WAKE_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in HW_WAKE_SET.

Using HW_WAKE_CLR and HW_WAKE_SET prevents unintentionally setting or clearing other bits.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4	eDMA Wakeup Source 4 Enables eDMA to wakeup by SAI0-2 DMA request. 0b - Disables 1b - Enables
3 HWWK_SRC3	eDMA Wakeup source 3 Enables eDMA to wakeup by LPSPI14 and LPSPI16 DMA request. 0b - Disables 1b - Enables
2 HWWK_SRC2	eDMA Wakeup Source 2 Enables eDMA to wakeup by LP_FLEXCOMM0-13 DMA request. 0b - Disables 1b - Enables
1 HWWK_SRC1	eDMA Wakeup Source 1 Enables eDMA to wakeup by FLEXIO DMA request. 0b - Disables 1b - Enables
0 —	Reserved

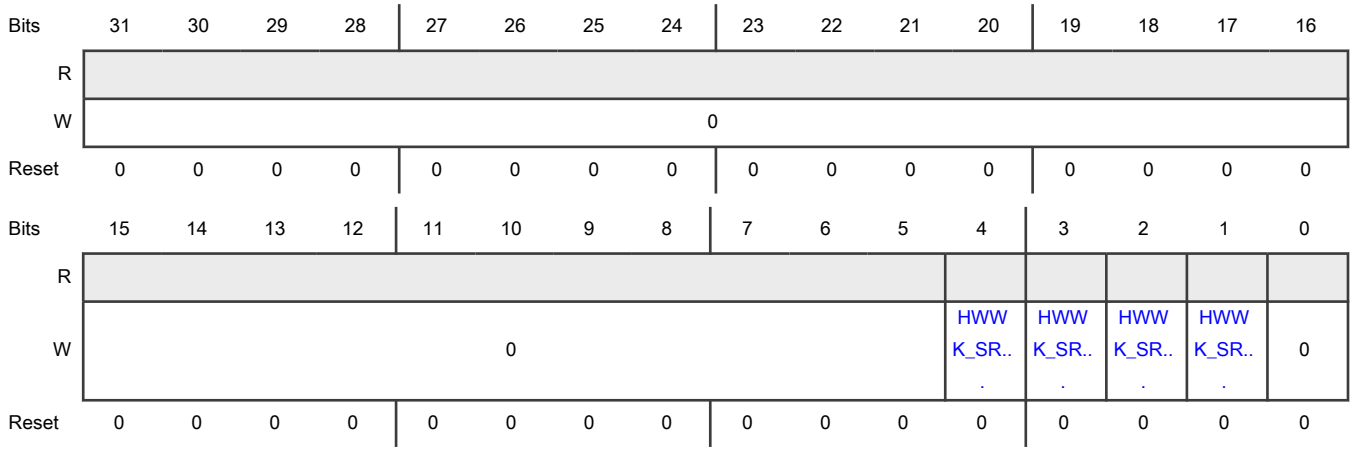
30.4.1.23 eDMA Wakeup Set (HW_WAKE_SET)

Offset

Register	Offset
HW_WAKE_SET	194h

Function
Writing a 1 to a bit position sets the corresponding position in HW_WAKE. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4_SET	eDMA Wakeup Source 4 Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 HWWK_SRC3_SET	eDMA Wakeup Source 3 Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 HWWK_SRC2_SET	eDMA Wakeup Source 2 Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 HWWK_SRC1_SET	eDMA Wakeup Source 1 Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.1.24 eDMA Wakeup Clear (HW_WAKE_CLR)

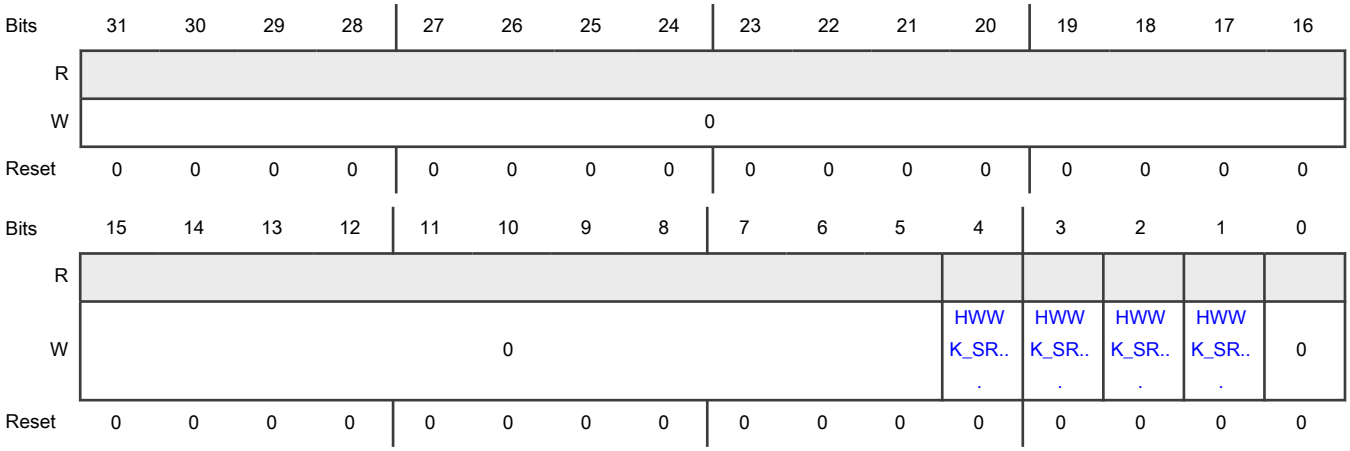
Offset

Register	Offset
HW_WAKE_CLR	198h

Function

Writing a 1 to a bit position clears the corresponding position in HW_WAKE. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4_CLR	eDMA Wakeup Source 4 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 HWWK_SRC3_CLR	eDMA Wakeup Source 3 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 HWWK_SRC2_CLR	eDMA Wakeup Source 2 Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
1 HWWK_SRC1_CLR	eDMA Wakeup Source 1 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.1.25 Shared Resources Mask (SHARED_MASK0)

Offset

Register	Offset
SHARED_MASK0	1A0h

Function

Enables the masking of SLEEPCON0 control for shared modules. To modify a SHARED_MASK0 bit, use the corresponding SHARED_MASK0_CLR and SHARED_MASK0_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in SHARED_MASK0_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in SHARED_MASK0_SET.

Using SHARED_MASK0_CLR and SHARED_MASK0_SET prevents unintentionally setting or clearing other bits.

NOTE

The masking of a shared module must not be disabled in both SLEEPCON0 and SLEEPCON1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								NIC1_MA...	NIC0_MA...	FLEXIO_...	LPSPI16...	LPSPI14...	XSPI2_M...	EZHV_MA...	VGPU_MA...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RTC_MASK				LPI2C15...	MICFIL_...	ACMP0_M...	ADC0_MA...
W											0	0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 NIC1_MASK	NIC_MEDIA1 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
22 NIC0_MASK	NIC_MEDIA0 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
21 FLEXIO_MASK	FLEXIO Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
20 LPSPI16_MASK	LPSPI16 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
19 LPSPI14_MASK	LPSPI14 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
18 XSPI2_MASK	XSPI2 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
17	EZH-V Mask

Table continues on the next page...

Table continued from the previous page...

Field	Function
EZHV_MASK	Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
16 VGPU_MASK	VGPU Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
15-7 —	Reserved
6 RTC_MASK	Mask bit of RTC0 and RTC1 Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK	LPI2C15 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
2 MICFIL_MASK	MICFIL Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
1 ACMP0_MASK	ACMP0 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
0 ADC0_MASK	ADC0 Mask Writing a 1 enables the masking of this SLEEPCON0 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables

30.4.1.26 Shared Resources Mask Set (SHARED_MASK0_SET)

Offset

Register	Offset
SHARED_MASK0_SET	1A4h

Function

Writing a 1 to a bit position sets the corresponding position in SHARED_MASK0. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0								NIC1_MA...	NIC0_MA...	FLEXIO_...	LPSPI16...	LPSPI14...	XSPI2_M...	EZHV_MA...	VGPU_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0									RTC_MAS...	0	0	LPI2C15...	MICFIL_...	ACMP0_M...	ADC0_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23	NIC_MEDIA1 Mask Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
NIC1_MASK_SET	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 NIC0_MASK_SET	NIC_MEDIA0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21 FLEXIO_MASK_SET	FLEXIO Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20 LPSPI16_MASK_SET	LPSPI16 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19 LPSPI14_MASK_SET	LPSPI14 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 XSPI2_MASK_SET	XSPI2 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 EZHV_MASK_SET	EZH-V Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16 VGPU_MASK_SET	VGPU Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-7 —	Reserved
6 RTC_MASK_SET	RTC0 and RTC1 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK_SET	LPI2C15 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 MICFIL_MASK_SET	MICFIL Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 ACMP0_MASK_SET	ACMP0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 ADC0_MASK_SET	ADC0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.1.27 Shared Resources Mask Clear (SHARED_MASK0_CLR)

Offset

Register	Offset
SHARED_MASK0_CLR	1A8h

Function

Writing a 1 to a bit position clears the corresponding position in SHARED_MASK0. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0								NIC1_MA...	NIC0_MA...	FLEXIO_...	LPSPI16...	LPSPI14...	XSPI2_M...	EZHV_MA...	VGPU_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0									RTC_MAS...	0	0	LPI2C15...	MICFIL_...	ACMP0_M...	ADC0_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 NIC1_MASK_CLR	NIC_MEDIA1 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 NIC0_MASK_CLR	NIC_MEDIA0 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21 FLEXIO_MASK_CLR	FLEXIO Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 LPSPI16_MASK_CLR	LPSPI16 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19	LPSPI14 Mask Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPSPI14_MASK_CLR	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 XSPI2_MASK_CLR	XSPI2 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 EZHV_MASK_CLR	EZH-V Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16 VGPU_MASK_CLR	VGPU Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15-7 —	Reserved
6 RTC_MASK_CLR	RTC0 and RTC1 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK_CLR	LPI2C15 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2	MICFIL Mask Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
MICFIL_MASK_CLR	0b - No effect 1b - Clears the bit
1 ACMP0_MASK_CLR	ACMP0 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 ADC0_MASK_CLR	ADC0 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.1.28 Media Domain Shared Controller Low-power Control (SHA_MED_CCTRL0)

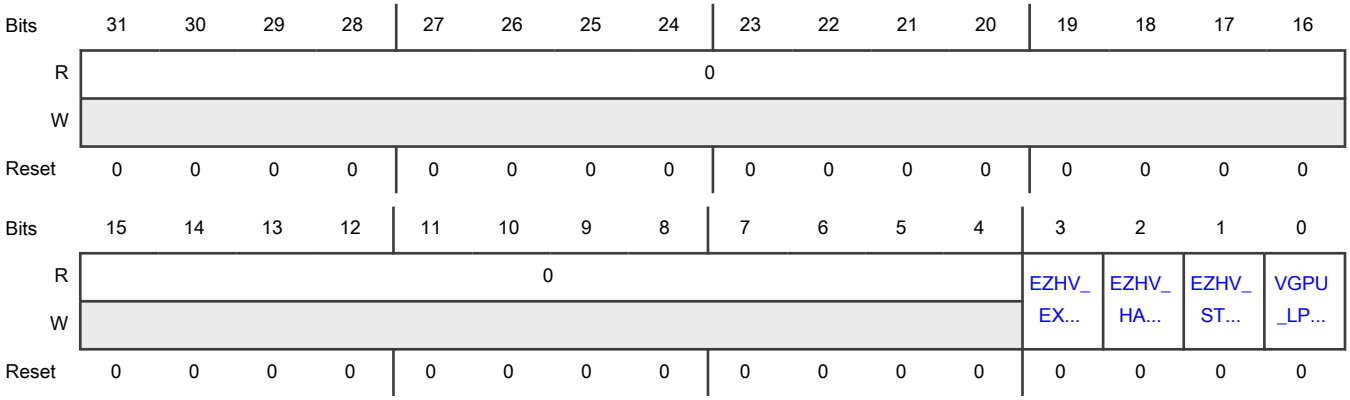
Offset

Register	Offset
SHA_MED_CCTRL0	1B0h

Function

Sends low-power request to shared bus controller if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 EZHV_EXIT_WAIT	EZH-V Exit Wait Mode Sends exit wait mode request to EZH-V. 0b - No request 1b - Sends
2 EZHV_HALT	EZH-V Halt Control Sends Halt request to EZH-V. 0b - No request 1b - Sends request
1 EZHV_STOP	EZH-V Stop Control Sends Stop request to EZH-V. 0b - No request 1b - Sends request
0 VGPU_LPREQ	VGPU Low-Power Control Sends low-power request to VGPU 0b - No request 1b - Sends request

30.4.1.29 Media Domain Shared Resources Target Low-power Control (SHA_MED_TCTRL0)

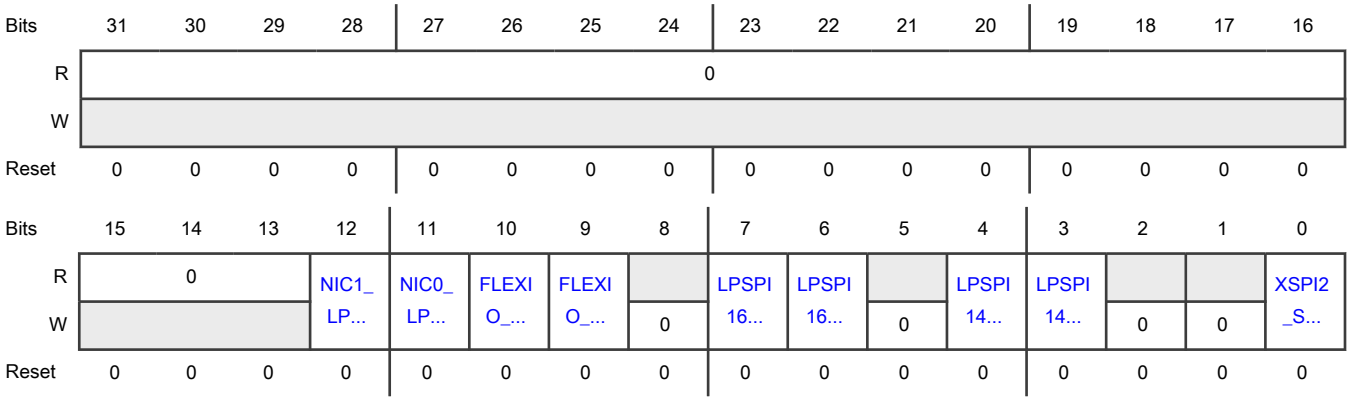
Offset

Register	Offset
SHA_MED_TCTRL0	1B4h

Function

Sends low-power request to shared bus target in the Media domain if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 NIC1_LPREQ	NIC_MEDIA1 Low-Power Request Sends low-power request to NIC_MEDIA1. 0b - No request 1b - Sends request
11 NIC0_LPREQ	NIC_MEDIA0 Low-Power Request Sends low-power request to NIC_MEDIA0. 0b - No request 1b - Sends request
10 FLEXIO_F_LPREQ	FLEXIO Function Low-Power Request Sends function low-power request to FLEXIO. 0b - No request 1b - Sends request
9 FLEXIO_B_LPREQ	FLEXIO Bus Low-Power Request Sends bus low-power request to FLEXIO. 0b - No request 1b - Sends request
8 —	Reserved
7	LPSPI16 Function Low-Power Request

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPSPI16_F_LPREQ	Sends function low-power request to LPSPI16. 0b - No request 1b - Sends request
6 LPSPI16_B_LPREQ	LPSPI16 Bus Low-Power Request Sends bus low-power request to LPSPI16. 0b - No request 1b - Sends request
5 —	Reserved
4 LPSPI14_F_LPREQ	LPSPI14 Function Low-Power Request Sends function low-power request to LPSPI14. 0b - No request 1b - Sends request
3 LPSPI14_B_LPREQ	LPSPI14 Bus Low-Power Request Sends bus low-power request to LPSPI14. 0b - No request 1b - Sends request
2 —	Reserved
1 —	Reserved
0 XSPI2_STOP	XSPI2 Stop Control Sends Stop request to XSPI2. 0b - No request 1b - Sends request

30.4.1.30 Shared VDD1_SENSE Domain Targets Low-power Control (SHA_SEN_TCTRL0)

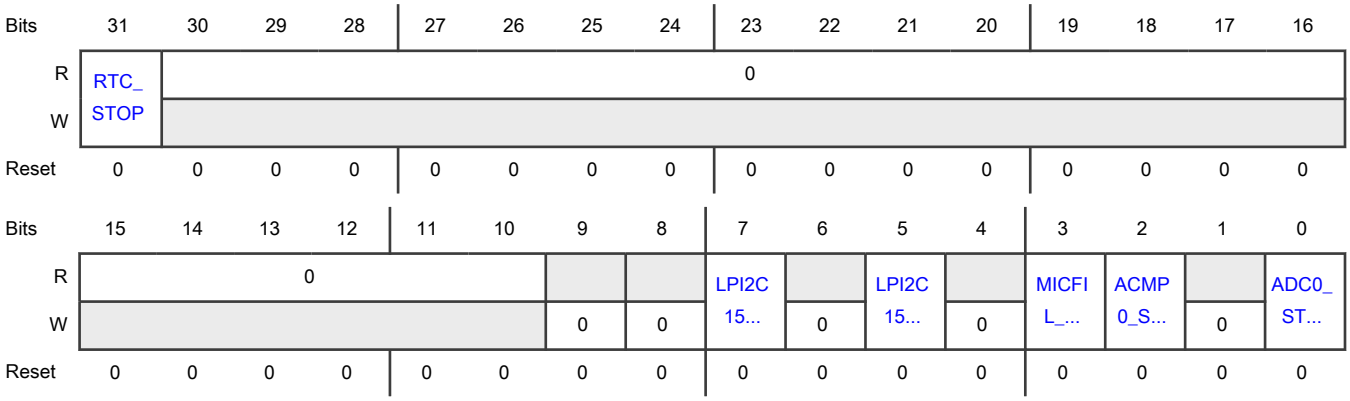
Offset

Register	Offset
SHA_SEN_TCTRL0	1BCh

Function

Sends low-power request to shared bus target in the VDD1_SENSE domain if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31 RTC_STOP	RTC0 and RTC1 Stop Sends Stop request to RTC0 and RTC1 modules. **NOTE** The RTC modules enter low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related SHA_MEDSEN_TSTAT0[RTC_STOPPED] bit will always read as 1. 0b - No request 1b - Sends request
30-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 LPI2C15_F_LPREQ	LPI2C15 Function Low-Power Request Sends function low-power request to LPI2C15. 0b - No request 1b - Sends request
6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
5 LPI2C15_B_LPREQ	LPI2C15 Bus Low-Power Request Sends bus low-power request to LPI2C15. 0b - No request 1b - Sends request
4 —	Reserved
3 MICFIL_STOP	MICFIL Stop Sends Stop request to MICFIL. 0b - No request 1b - Sends request
2 ACMP0_STOP	ACMP0 Stop Sends Stop request to ACMP0. **NOTE** The ACMP0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related SHA_MEDSEN_TSTAT0[ACMP0_STOP] bit will always read as 1. 0b - No request 1b - Sends request
1 —	Reserved
0 ADC0_STOP	ADC0 Stop Sends Stop request to ADC0. 0b - No request 1b - Sends request

30.4.1.31 Private Resources Controllers Low-power Control 0 (PRIVATE_CCTRL0)

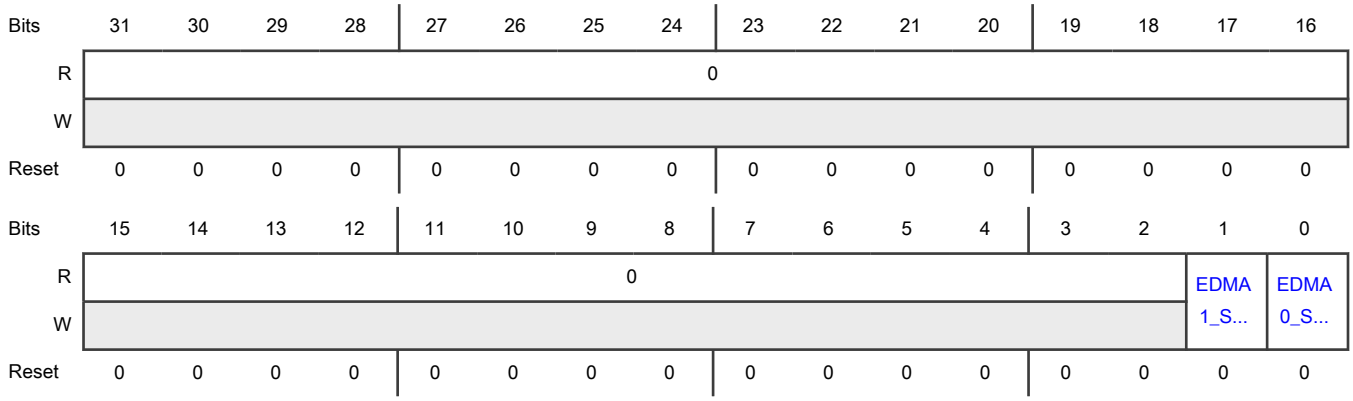
Offset

Register	Offset
PRIVATE_CCTRL0	1C0h

Function

Sends low-power request to private bus controllers.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 EDMA1_STOP	eDMA1 Stop Sends Stop request to eDMA1 0b - No request 1b - Sends request
0 EDMA0_STOP	eDMA0 Stop Sends Stop request to eDMA0 0b - No request 1b - Sends request

30.4.1.32 Private Resources Targets Low-power Control 0 (PRIVATE_TCTRL0)

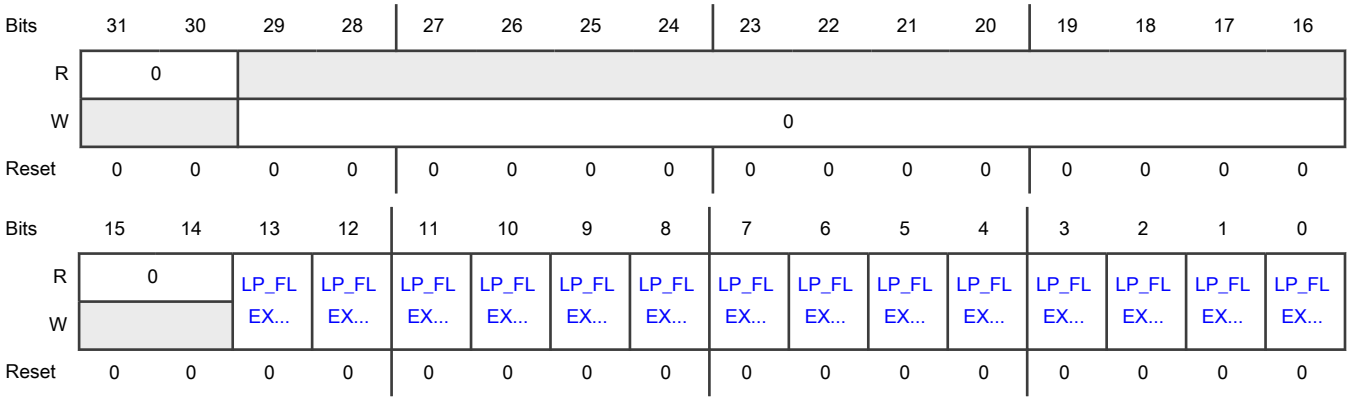
Offset

Register	Offset
PRIVATE_TCTRL0	1C4h

Function

Sends low-power request to private bus targets.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-16 —	Reserved
15-14 —	Reserved
13 LP_FLEXCOMM13_B_LPREQ	LP_FLEXCOMM13 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM13. 0b - No request 1b - Sends request
12 LP_FLEXCOMM12_B_LPREQ	LP_FLEXCOMM12 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM12. 0b - No request 1b - Sends request
11 LP_FLEXCOMM11_B_LPREQ	LP_FLEXCOMM11 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM11. 0b - No request 1b - Sends request
10 LP_FLEXCOMM10_B_LPREQ	LP_FLEXCOMM10 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM10. 0b - No request

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sends request
9 LP_FLEXCOMM9_B_LPREQ	LP_FLEXCOMM9 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM9. 0b - No request 1b - Sends request
8 LP_FLEXCOMM8_B_LPREQ	LP_FLEXCOMM8 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM8. 0b - No request 1b - Sends request
7 LP_FLEXCOMM7_B_LPREQ	LP_FLEXCOMM7 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM7. 0b - No request 1b - Sends request
6 LP_FLEXCOMM6_B_LPREQ	LP_FLEXCOMM6 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM6. 0b - No request 1b - Sends request
5 LP_FLEXCOMM5_B_LPREQ	LP_FLEXCOMM5 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM5. 0b - No request 1b - Sends request
4 LP_FLEXCOMM4_B_LPREQ	LP_FLEXCOMM4 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM4. 0b - No request 1b - Sends request
3 LP_FLEXCOMM3_B_LPREQ	LP_FLEXCOMM3 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM3. 0b - No request 1b - Sends request
2	LP_FLEXCOMM2 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM2.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM2_B_LPREQ	0b - No request 1b - Sends request
1 LP_FLEXCOMM1_B_LPREQ	LP_FLEXCOMM1 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM1. 0b - No request 1b - Sends request
0 LP_FLEXCOMM0_B_LPREQ	LP_FLEXCOMM0 Bus Low-Power Request Sends bus low-power request to LP_FLEXCOMM0. 0b - No request 1b - Sends request

30.4.1.33 Private Resources Targets Low-power Control 1 (PRIVATE_TCTRL1)

Offset

Register	Offset
PRIVATE_TCTRL1	1C8h

Function

Sends low-power request to private bus targets.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-14 —	Reserved
13 LP_FLEXCOMM13_F_LPREQ	LP_FLEXCOMM13 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM13. 0b - No request 1b - Sends request
12 LP_FLEXCOMM12_F_LPREQ	LP_FLEXCOMM12 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM12. 0b - No request 1b - Sends request
11 LP_FLEXCOMM11_F_LPREQ	LP_FLEXCOMM11 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM11. 0b - No request 1b - Sends request
10 LP_FLEXCOMM10_F_LPREQ	LP_FLEXCOMM10 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM10. 0b - No request 1b - Sends request
9 LP_FLEXCOMM9_F_LPREQ	LP_FLEXCOMM9 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM9. 0b - No request 1b - Sends request
8 LP_FLEXCOMM8_F_LPREQ	LP_FLEXCOMM8 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM8. 0b - No request 1b - Sends request
7 LP_FLEXCOMM7_F_LPREQ	LP_FLEXCOMM7 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM7. 0b - No request 1b - Sends request
6	LP_FLEXCOMM6 Function Low-Power Request

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM6_F_LPREQ	Sends function low-power request to LP_FLEXCOMM6. 0b - No request 1b - Sends request
5 LP_FLEXCOMM5_F_LPREQ	LP_FLEXCOMM5 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM5. 0b - No request 1b - Sends request
4 LP_FLEXCOMM4_F_LPREQ	LP_FLEXCOMM4 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM4. 0b - No request 1b - Sends request
3 LP_FLEXCOMM3_F_LPREQ	LP_FLEXCOMM3 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM3. 0b - No request 1b - Sends request
2 LP_FLEXCOMM2_F_LPREQ	LP_FLEXCOMM2 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM2. 0b - No request 1b - Sends request
1 LP_FLEXCOMM1_F_LPREQ	LP_FLEXCOMM1 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM1. 0b - No request 1b - Sends request
0 LP_FLEXCOMM0_F_LPREQ	LP_FLEXCOMM0 Function Low-Power Request Sends function low-power request to LP_FLEXCOMM0. 0b - No request 1b - Sends request

30.4.1.34 Private Resources Targets Low-power Control 2 (PRIVATE_TCTRL2)

Offset

Register	Offset
PRIVATE_TCTRL2	1CCh

Function

Sends low-power request to private bus targets.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														I3C1_ST...	I3C0_ST...
W													0	0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		XSPI1_S...		XSPI0_S...		SAI2_LP...	SAI1_LP...	SAI0_LP...	GPIO7_L...	GPIO6_L...	GPIO5_L...	GPIO4_L...	GPIO3_L...	GPIO2_L...	GPIO1_L...	GPIO0_L...
W	0		0		0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19 —	Reserved
18 —	Reserved
17 I3C1_STOP	I3C1 Stop Sends Stop request to I3C1. **NOTE** The I3C1 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related PRIVATE_TSTAT1[I3C1_STOP] bit will always read as 1. 0b - No request 1b - Sends request

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 I3C0_STOP	I3C0 Stop Sends Stop request to I3C0. **NOTE** The I3C0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related PRIVATE_TSTAT1[I3C0_STOP] bit will always read as 1. 0b - No request 1b - Sends request
15 —	Reserved
14 XSPI1_STOP	XSPI1 Stop Sends Stop request to XSPI1. 0b - No request 1b - Sends request
13 —	Reserved
12 XSPI0_STOP	XSPI0 Stop Sends Stop request to XSPI0. 0b - No request 1b - Sends request
11 —	Reserved
10 SAI2_LPREQ	SAI2 Low-Power Request Sends low-power request to SAI2. 0b - No request 1b - Sends request
9 SAI1_LPREQ	SAI1 Low-Power Request Sends low-power request to SAI1. 0b - No request 1b - Sends request
8	SAI0 Low-Power Request

Table continues on the next page...

Table continued from the previous page...

Field	Function
SAI0_LPREQ	Sends low-power request to SAI0. 0b - No request 1b - Sends request
7 GPIO7_LPREQ	GPIO7 Low-Power Request Sends low-power request to GPIO7. 0b - No request 1b - Sends request
6 GPIO6_LPREQ	GPIO6 Low-Power Request Sends low-power request to GPIO6. 0b - No request 1b - Sends request
5 GPIO5_LPREQ	GPIO5 Low-Power Request Sends low-power request to GPIO5. 0b - No request 1b - Sends request
4 GPIO4_LPREQ	GPIO4 Low-Power Request Sends low-power request to GPIO4. 0b - No request 1b - Sends request
3 GPIO3_LPREQ	GPIO3 Low-Power Request Sends low-power request to GPIO3. 0b - No request 1b - Sends request
2 GPIO2_LPREQ	GPIO2 Low-Power Request Sends low-power request to GPIO2. 0b - No request 1b - Sends request
1 GPIO1_LPREQ	GPIO1 Low-Power Request Sends low-power request to GPIO1. 0b - No request 1b - Sends request

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 GPIO0_LPREQ	GPIO0 Low-Power Request Sends low-power request to GPIO0. 0b - No request 1b - Sends request

30.4.1.35 Media Domain Shared Controllers Low-power Status (SHA_MED_CSTAT0)

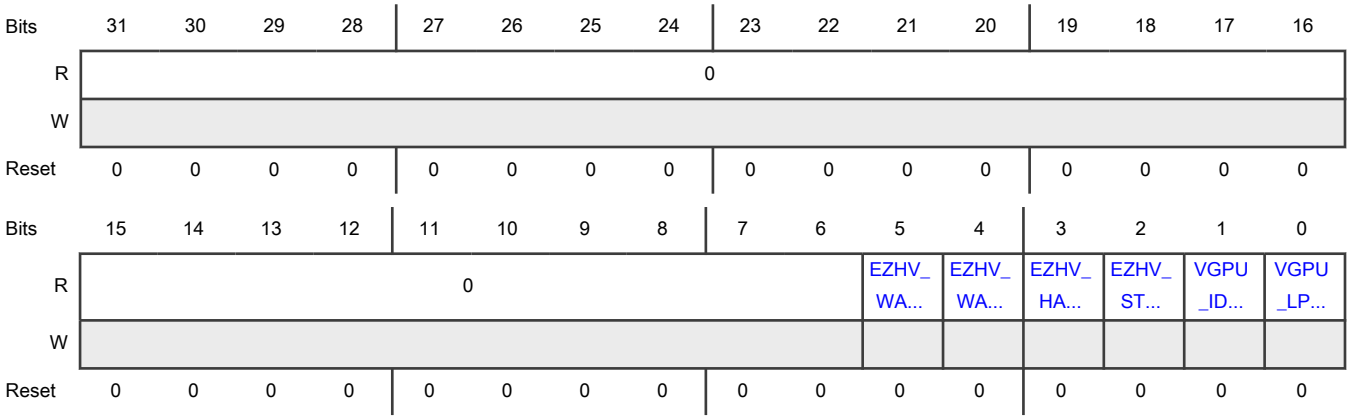
Offset

Register	Offset
SHA_MED_CSTAT0	1D0h

Function

Indicates the low-power ready status for bus controllers in the Media domain.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 EZHV_WAKEUP	Wakeup Request from EZH-V Indicates whether EZH-V needs clock running to wakeup 0b - Not needs 1b - Needs

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 EZHV_WAITING	EZH-V Waiting Status Indicates the waiting status of EZH-V. 0b - Not in waiting status 1b - In waiting status
3 EZHV_HALTED	EZH-V Halt Status Indicates the Halt status of EZH-V. 0b - Not in Halt status 1b - In Halt status
2 EZHV_STOPPED	EZH-V Stop Status Indicates the Stop status of EZH-V. 0b - Not in Stop status 1b - In Stop status
1 VGPU_IDLE	VGPU Idle Status Indicates VGPU Idle status. 0b - Not in Idle status, VGPU has operation to perform. 1b - In Idle status, VGPU might accept low-power request.
0 VGPU_LPACCEPT	VGPU Low-power Accept Status Indicates accept status to low-power request of VGPU. 0b - Not accept 1b - Accept

30.4.1.36 Media and Sense Domain Shared Targets Low-power Status (SHA_MEDSEN_TSTAT0)

Offset

Register	Offset
SHA_MEDSEN_TSTAT0	1D4h

Function
Indicates the low-power status for bus targets in Media and Sense domains.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RTC_STO...	0								0	0	LPI2C15...	LPI2C15...	MICFIL_...	ACMP0_S...	ADC0_ST...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							NIC1_LP...	NIC0_LP...	FLEXIO_...	FLEXIO_...	LPSPI16...	LPSPI16...	LPSPI14...	LPSPI14...	XSPI2_S...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 RTC_STOPPED	RTC0 and RTC1 Stop Status Indicates the Stop status of RTC modules. **NOTE** The RTC modules enter low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
30-23 —	Reserved
22 —	Reserved
21 —	Reserved
20 LPI2C15_F_LPACCEPT	LPI2C15 Function Low-power Accept Status Indicates the accept status to the function low-power request of LPI2C15. 0b - Not accept 1b - Accept
19 LPI2C15_B_LPACCEPT	LPI2C15 Bus Low-power Accept Status Indicates the accept status to the bus low-power request of LPI2C15. 0b - Not accept 1b - Accept

Table continues on the next page...

Table continued from the previous page...

Field	Function
18 MICFIL_STOPPED	MICFIL Stop Status Indicates the Stop status of MICFIL. 0b - Not in Stop status 1b - In Stop status
17 ACMP0_STOP	ACMP0 Stop Status Indicates the Stop status of ACMP0. **NOTE** The ACMP0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
16 ADC0_STOP	ADC0 Stop Status Indicates the Stop status of ADC0. 0b - Not in Stop status 1b - In Stop status
15-9 —	Reserved
8 NIC1_LPACCEPT	NIC_MEDIA1 Low-power Accept Status Indicates the accept status to low-power request of NIC_MEDIA1. 0b - Not accept 1b - Accept
7 NIC0_LPACCEPT	NIC_MEDIA0 Accept Low-power Status Indicates the accept status to low-power request of NIC_MEDIA0. 0b - Not accept 1b - Accept
6 FLEXIO_F_LPACCEPT	FLEXIO Function Low-power Accept Status Indicates the accept status to function low-power request of FLEXIO 0b - Not accept 1b - Accept
5 FLEXIO_B_LPACCEPT	FLEXIO Bus Low-power Accept Status Indicates the accept status to bus low-power request of FLEXIO

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not accept 1b - Accept
4 LPSPI16_F_LP ACCEPT	LPSPI16 Function Low-power Accept Status Indicates the accept status to function low-power request of LPSPI16. 0b - Not accept 1b - Accept
3 LPSPI16_B_LP ACCEPT	LPSPI16 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LPSPI16. 0b - Not accept 1b - Accept
2 LPSPI14_F_LP ACCEPT	LPSPI14 Function Low-power Accept Status Indicates the accept status to function low-power request of LPSPI14 0b - Not accept 1b - Accept
1 LPSPI14_B_LP ACCEPT	LPSPI14 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LPSPI14. 0b - Not accept 1b - Accept
0 XSPI2_STOP	XSPI2 Stop Status Indicates Stop status of XSPI2 0b - Not in Stop status 1b - In Stop status

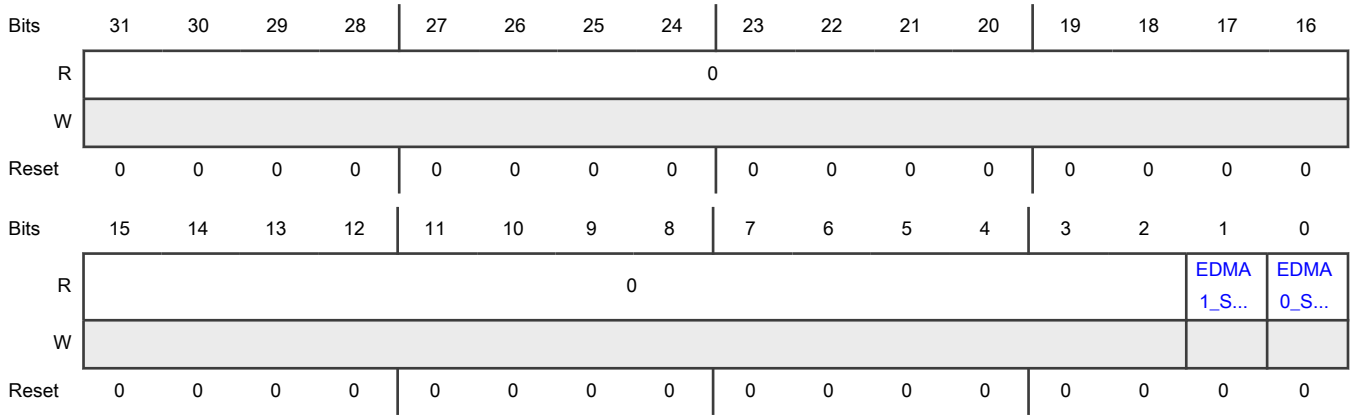
30.4.1.37 Private Resources Controller Low-power Status (PRIVATE_CSTAT0)

Offset

Register	Offset
PRIVATE_CSTAT0	1E0h

Function
Indicates the low-power ready status of bus controllers that can only be controlled by SLEEPCON0.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 EDMA1_STOP	eDMA1 Stop Status Indicates the Stop status of eDMA1. 0b - Not in Stop status 1b - In Stop status
0 EDMA0_STOP	eDMA0 Stop Status Indicates the Stop status of eDMA0. 0b - Not in Stop status 1b - In Stop status

30.4.1.38 Private Resources Targets Low-power Status 0 (PRIVATE_TSTAT0)

Offset

Register	Offset
PRIVATE_TSTAT0	1E4h

Function

Indicates low-power ready status for bus targets that can only be controlled by SLEEPCON0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29 LP_FLEXCOMM13_F_LPACCEPT	LP_FLEXCOMM13 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM13. 0b - Not accept 1b - Accept
28 LP_FLEXCOMM12_F_LPACCEPT	LP_FLEXCOMM12 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM12. 0b - Not accept 1b - Accept
27 LP_FLEXCOMM11_F_LPACCEPT	LP_FLEXCOMM11 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM11. 0b - Not accept 1b - Accept
26 LP_FLEXCOMM10_F_LPACCEPT	LP_FLEXCOMM10 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM10. 0b - Not accept 1b - Accept
25	LP_FLEXCOMM9 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM9.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM9_F_LPACCEPT	0b - Not accept 1b - Accept
24 LP_FLEXCOMM8_F_LPACCEPT	LP_FLEXCOMM8 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM8. 0b - Not accept 1b - Accept
23 LP_FLEXCOMM7_F_LPACCEPT	LP_FLEXCOMM7 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM7. 0b - Not accept 1b - Accept
22 LP_FLEXCOMM6_F_LPACCEPT	LP_FLEXCOMM6 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM6. 0b - Not accept 1b - Accept
21 LP_FLEXCOMM5_F_LPACCEPT	LP_FLEXCOMM5 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM5. 0b - Not accept 1b - Accept
20 LP_FLEXCOMM4_F_LPACCEPT	LP_FLEXCOMM4 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM4. 0b - Not accept 1b - Accept
19 LP_FLEXCOMM3_F_LPACCEPT	LP_FLEXCOMM3 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM3. 0b - Not accept 1b - Accept
18 LP_FLEXCOMM2_F_LPACCEPT	LP_FLEXCOMM2 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM2. 0b - Not accept 1b - Accept
17	LP_FLEXCOMM1 Function Low-power Accept Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM1_F_LPACCEPT	Indicates the accept status to function low-power request of LP_FLEXCOMM1. 0b - Not accept 1b - Accept
16 LP_FLEXCOMM0_F_LPACCEPT	LP_FLEXCOMM0 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM0. 0b - Not accept 1b - Accept
15-14 —	Reserved
13 LP_FLEXCOMM13_B_LPACCEPT	LP_FLEXCOMM13 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM13. 0b - Not accept 1b - Accept
12 LP_FLEXCOMM12_B_LPACCEPT	LP_FLEXCOMM12 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM12. 0b - Not accept 1b - Accept
11 LP_FLEXCOMM11_B_LPACCEPT	LP_FLEXCOMM11 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM11. 0b - Not accept 1b - Accept
10 LP_FLEXCOMM10_B_LPACCEPT	LP_FLEXCOMM10 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM10. 0b - Not accept 1b - Accept
9 LP_FLEXCOMM9_B_LPACCEPT	LP_FLEXCOMM9 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM9. 0b - Not accept 1b - Accept
8	LP_FLEXCOMM8 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM8.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM8_B_LPACCEPT	0b - Not accept 1b - Accept
7 LP_FLEXCOMM7_B_LPACCEPT	LP_FLEXCOMM7 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM7. 0b - Not accept 1b - Accept
6 LP_FLEXCOMM6_B_LPACCEPT	LP_FLEXCOMM6 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM6. 0b - Not accept 1b - Accept
5 LP_FLEXCOMM5_B_LPACCEPT	LP_FLEXCOMM5 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM5. 0b - Not accept 1b - Accept
4 LP_FLEXCOMM4_B_LPACCEPT	LP_FLEXCOMM4 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM4. 0b - Not accept 1b - Accept
3 LP_FLEXCOMM3_B_LPACCEPT	LP_FLEXCOMM3 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM3. 0b - Not accept 1b - Accept
2 LP_FLEXCOMM2_B_LPACCEPT	LP_FLEXCOMM2 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM2. 0b - Not accept 1b - Accept
1 LP_FLEXCOMM1_B_LPACCEPT	LP_FLEXCOMM1 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM1. 0b - Not accept 1b - Accept
0	LP_FLEXCOMM0 Bus Low-power Accept Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM0_B_LPACCEPT	Indicates the accept status to bus low-power request of LP_FLEXCOMM0. 0b - Not accept 1b - Accept

30.4.1.39 Private Resources Targets Low-power Status 1 (PRIVATE_TSTAT1)

Offset

Register	Offset
PRIVATE_TSTAT1	1E8h

Function

Indicates low-power status for bus targets that can only be controlled by SLEEPCON0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														I3C1_ST...	I3C0_ST...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XSPI1_S...	XSPI0_S...	0	SAI2_LP...	SAI1_LP...	SAI0_LP...	GPIO7_L...	GPIO6_L...	GPIO5_L...	GPIO4_L...	GPIO3_L...	GPIO2_L...	GPIO1_L...	GPIO0_L...	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-18 —	Reserved
17 I3C1_STOP	I3C1 Stop Status Indicates the Stop status of I3C1. **NOTE** The I3C1 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not in Stop status 1b - In Stop status
16 I3C0_STOP	I3C0 Stop Status Indicates the Stop status of I3C0. **NOTE** The I3C0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
15 XSPI1_STOP	XSPI1 Stop Status Indicates the Stop status of XSPI1. 0b - Not in Stop status 1b - In Stop status
14 XSPI0_STOP	XSPI0 Stop Status Indicates the Stop status of XSPI0. 0b - Not in Stop status 1b - In Stop status
13 —	Reserved
12 SAI2_LPACCEPT	SAI2 Low-power Accept Status Indicates the accept status to low-power request of SAI2. 0b - Not accept 1b - Accept
11 SAI1_LPACCEPT	SAI1 Low-power Accept Status Indicates the accept status to low-power request of SAI1. 0b - Not accept 1b - Accept
10 SAI0_LPACCEPT	SAI0 Low-power Accept Status Indicates the accept status to low-power request of SAI0. 0b - Not accept 1b - Accept

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 GPIO7_LPACCEPT	GPIO7 Low-power Accept Status Indicates the accept status to low-power request of GPIO7. 0b - Not accept 1b - Accept
8 GPIO6_LPACCEPT	GPIO6 Low-power Accept Status Indicates the accept status to low-power request of GPIO6. 0b - Not accept 1b - Accept
7 GPIO5_LPACCEPT	GPIO5 Low-power Accept Status Indicates the accept status to low-power request of GPIO5. 0b - Not accept 1b - Accept
6 GPIO4_LPACCEPT	GPIO4 Low-power Accept Status Indicates the accept status to low-power request of GPIO4. 0b - Not accept 1b - Accept
5 GPIO3_LPACCEPT	GPIO3 Low-power Accept Status Indicates the accept status to low-power request of GPIO3. 0b - Not accept 1b - Accept
4 GPIO2_LPACCEPT	GPIO2 Low-power Accept Status Indicates the accept status to low-power request of GPIO2. 0b - Not accept 1b - Accept
3 GPIO1_LPACCEPT	GPIO1 Low-power Accept Status Indicates the accept status to low-power request of GPIO1. 0b - Not accept 1b - Accept
2 GPIO0_LPACCEPT	GPIO0 Low-power Accept Status Indicates the accept status to low-power request of GPIO0. 0b - Not accept

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Accept
1 —	Reserved
0 —	Reserved

30.4.1.40 Low-Power Hints 0 (LP_HINT0)

Offset

Register	Offset
LP_HINT0	1ECh

Function

Indicates the active or inactive status of modules. If a module is inactive, you can use xxxxCTRLn register to send low-power request to it.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LPSPI16...	LPSPI14...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPSPI16...	LPSPI14...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LPSPI16_F_LPHINT	LPSPI16 Function Clock Related Logic Status Indicates status of LPSPI16 that driven by function clock. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 LPSPI14_F_LPHINT	LPSPI14 Function Clock Related Logic Status Indicates status of LPSPI14 that driven by function clock. 0b - Inactive 1b - Active
29 LP_FLEXCOMM13_F_LPHINT	LP_FLEXCOMM13 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM13 that driven by function clock. 0b - Inactive 1b - Active
28 LP_FLEXCOMM12_F_LPHINT	LP_FLEXCOMM12 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM12 that driven by function clock. 0b - Inactive 1b - Active
27 LP_FLEXCOMM11_F_LPHINT	LP_FLEXCOMM11 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM11 that driven by function clock. 0b - Inactive 1b - Active
26 LP_FLEXCOMM10_F_LPHINT	LP_FLEXCOMM10 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM10 that driven by function clock. 0b - Inactive 1b - Active
25 LP_FLEXCOMM9_F_LPHINT	LP_FLEXCOMM9 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM9 that driven by function clock. 0b - Inactive 1b - Active
24 LP_FLEXCOMM8_F_LPHINT	LP_FLEXCOMM8 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM8 that driven by function clock. 0b - Inactive 1b - Active
23 LP_FLEXCOMM7_F_LPHINT	LP_FLEXCOMM7 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM7 that driven by function clock. 0b - Inactive

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Active
22 LP_FLEXCOMM6_F_LPHINT	LP_FLEXCOMM6 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM6 that driven by function clock. 0b - Inactive 1b - Active
21 LP_FLEXCOMM5_F_LPHINT	LP_FLEXCOMM5 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM5 that driven by function clock. 0b - Inactive 1b - Active
20 LP_FLEXCOMM4_F_LPHINT	LP_FLEXCOMM4 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM4 that driven by function clock. 0b - Inactive 1b - Active
19 LP_FLEXCOMM3_F_LPHINT	LP_FLEXCOMM3 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM3 that driven by function clock. 0b - Inactive 1b - Active
18 LP_FLEXCOMM2_F_LPHINT	LP_FLEXCOMM2 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM2 that driven by function clock. 0b - Inactive 1b - Active
17 LP_FLEXCOMM1_F_LPHINT	LP_FLEXCOMM1 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM1 that driven by function clock. 0b - Inactive 1b - Active
16 LP_FLEXCOMM0_F_LPHINT	LP_FLEXCOMM0 Function Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM0 that driven by function clock. 0b - Inactive 1b - Active
15	LPSPI16 Bus Clock Related Logic Status Indicates status of logic in LPSPI16 that driven by bus clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPSPI16_B_LPHINT	0b - Inactive 1b - Active
14 LPSPI14_B_LPHINT	LPSPI14 Bus Clock Related Logic Status Indicates status of logic in LPSPI14 that driven by bus clock. 0b - Inactive 1b - Active
13 LP_FLEXCOMM13_B_LPHINT	LP_FLEXCOMM13 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM13 that driven by bus clock. 0b - Inactive 1b - Active
12 LP_FLEXCOMM12_B_LPHINT	LP_FLEXCOMM12 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM12 that driven by bus clock. 0b - Inactive 1b - Active
11 LP_FLEXCOMM11_B_LPHINT	LP_FLEXCOMM11 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM11 that driven by bus clock. 0b - Inactive 1b - Active
10 LP_FLEXCOMM10_B_LPHINT	LP_FLEXCOMM10 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM10 that driven by bus clock. 0b - Inactive 1b - Active
9 LP_FLEXCOMM9_B_LPHINT	LP_FLEXCOMM9 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM9 that driven by bus clock. 0b - Inactive 1b - Active
8 LP_FLEXCOMM8_B_LPHINT	LP_FLEXCOMM8 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM8 that driven by bus clock. 0b - Inactive 1b - Active
7	LP_FLEXCOMM7 Bus Clock Related Logic Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM7_B_LPHINT	Indicates status of logic in LP_FLEXCOMM7 that driven by bus clock. 0b - Inactive 1b - Active
6 LP_FLEXCOMM6_B_LPHINT	LP_FLEXCOMM6 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM6 that driven by bus clock. 0b - Inactive 1b - Active
5 LP_FLEXCOMM5_B_LPHINT	LP_FLEXCOMM5 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM5 that driven by bus clock. 0b - Inactive 1b - Active
4 LP_FLEXCOMM4_B_LPHINT	LP_FLEXCOMM4 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM4 that driven by bus clock. 0b - Inactive 1b - Active
3 LP_FLEXCOMM3_B_LPHINT	LP_FLEXCOMM3 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM3 that driven by bus clock. 0b - Inactive 1b - Active
2 LP_FLEXCOMM2_B_LPHINT	LP_FLEXCOMM2 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM2 that driven by bus clock. 0b - Inactive 1b - Active
1 LP_FLEXCOMM1_B_LPHINT	LP_FLEXCOMM1 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM1 that driven by bus clock. 0b - Inactive 1b - Active
0 LP_FLEXCOMM0_B_LPHINT	LP_FLEXCOMM0 Bus Clock Related Logic Status Indicates status of logic in LP_FLEXCOMM0 that driven by bus clock. 0b - Inactive 1b - Active

30.4.1.41 Low-Power Hints 1 (LP_HINT1)

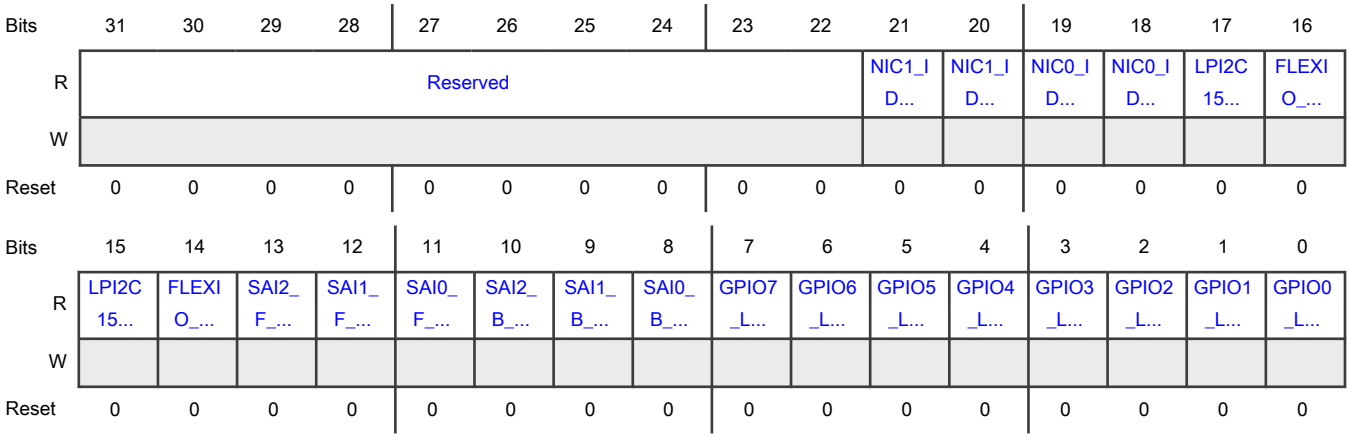
Offset

Register	Offset
LP_HINT1	1F0h

Function

Indicates the active or inactive status of modules. If a module is inactive, you can use xxxxCTRLn register to send low-power request to it.

Diagram



Fields

Field	Function
31-22 —	Reserved
21 NIC1_IDLE1	NIC_MEDIA1 Idle Status Indicates status of NIC_MEDIA1. 0b - Active 1b - Idle
20 NIC1_IDLE0	NIC_MEDIA1 Idle Status Indicates status of NIC_MEDIA1. 0b - Active 1b - Idle
19 NIC0_IDLE1	NIC_MEDIA0 Idle Status Indicates status of NIC_MEDIA0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Active 1b - Idle
18 NIC0_IDLE0	NIC_MEDIA0 Idle Status Indicates status of NIC_MEDIA0. 0b - Active 1b - Idle
17 LPI2C15_F_LPHINT	LPI2C15 Function Clock Related Logic Status Indicates status of logic in LPI2C15 that driven by function clock. 0b - Inactive 1b - Active
16 FLEXIO_F_LPHINT	FLEXIO Function Clock Related Logic Status Indicates status of logic in FLEXIO that driven by function clock. 0b - Inactive 1b - Active
15 LPI2C15_B_LPHINT	LPI2C15 Bus Clock Related Logic Status Indicates status of logic in LPI2C15 that driven by bus clock. 0b - Inactive 1b - Active
14 FLEXIO_B_LPHINT	FLEXIO Bus Clock Related Logic Status Indicates status of logic in FLEXIO that driven by bus clock. 0b - Inactive 1b - Active
13 SAI2_F_LPHINT	SAI2 Function Clock Related Logic Status Indicates status of logic in SAI2 that driven by function clock. 0b - Inactive 1b - Active
12 SAI1_F_LPHINT	SAI1 Function Clock Related Logic Status Indicates status of logic in SAI1 that driven by function clock. 0b - Inactive 1b - Active
11	SAI0 Function Clock Related Logic Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
SAI0_F_LPHINT	Indicates status of logic in SAI0 that driven by function clock. 0b - Inactive 1b - Active
10 SAI2_B_LPHINT	SAI2 Bus Clock Related Logic Status Indicates status of logic in SAI2 that driven by bus clock. 0b - Inactive 1b - Active
9 SAI1_B_LPHINT	SAI1 Bus Clock Related Logic Status Indicates status of logic in SAI1 that driven by bus clock. 0b - Inactive 1b - Active
8 SAI0_B_LPHINT	SAI0 Bus Clock Related Logic Status Indicates status of logic in SAI0 that driven by bus clock. 0b - Inactive 1b - Active
7 GPIO7_LPHINT	GPIO7 Status Indicates status of GPIO7. 0b - Inactive 1b - Active
6 GPIO6_LPHINT	GPIO6 Status Indicates status of GPIO6. 0b - Inactive 1b - Active
5 GPIO5_LPHINT	GPIO5 Status Indicates status of GPIO5. 0b - Inactive 1b - Active
4 GPIO4_LPHINT	GPIO4 Status Indicates status of GPIO4. 0b - Inactive 1b - Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
3 GPIO3_LPHINT	GPIO3 Status Indicates status of GPIO3. 0b - Inactive 1b - Active
2 GPIO2_LPHINT	GPIO2 Status Indicates status of GPIO2. 0b - Inactive 1b - Active
1 GPIO1_LPHINT	GPIO1 Status Indicates status of GPIO1. 0b - Inactive 1b - Active
0 GPIO0_LPHINT	GPIO0 Status Indicates status of GPIO0. 0b - Inactive 1b - Active

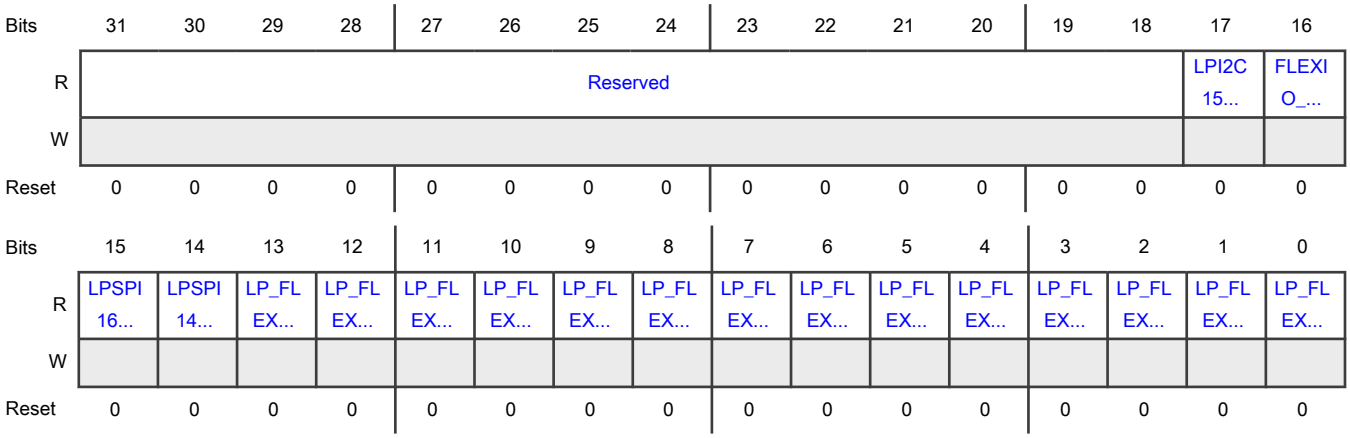
30.4.1.42 Low-Power Request Deny Status 0 (LP_DENY0)

Offset

Register	Offset
LP_DENY0	1F4h

Function
Indicates the module status to low-power request. If a module denied the low-power request, you need configure to send the low-power request again.

Diagram



Fields

Field	Function
31-18 —	Reserved
17 LPI2C15_LPDENY	LPI2C15 Low-Power Deny Status Indicates status of LPI2C15 to low-power request. 0b - Not deny 1b - Deny
16 FLEXIO_LPDENY	FLEXIO Low-Power Deny Status Indicates status of FLEXIO to low-power request. 0b - Not deny 1b - Deny
15 LPSPI16_LPDENY	LPSPI16 Low-Power Deny Status Indicates status of LPSPI16 to low-power request. 0b - Not deny 1b - Deny
14 LPSPI14_LPDENY	LPSPI14 Low-Power Deny Status Indicates status of LPSPI14 to low-power request. 0b - Not deny 1b - Deny
13 LP_FLEXCOMM13_LPDENY	LP_FLEXCOMM13 Low-Power Deny Status Indicates status of LP_FLEXCOMM13 to low-power request.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not deny 1b - Deny
12 LP_FLEXCOMM12_LPDENY	LP_FLEXCOMM12 Low-Power Deny Status Indicates status of LP_FLEXCOMM12 to low-power request. 0b - Not deny 1b - Deny
11 LP_FLEXCOMM11_LPDENY	LP_FLEXCOMM11 Low-Power Deny Status Indicates status of LP_FLEXCOMM11 to low-power request. 0b - Not deny 1b - Deny
10 LP_FLEXCOMM10_LPDENY	LP_FLEXCOMM10 Low-Power Deny Status Indicates status of LP_FLEXCOMM10 to low-power request. 0b - Not deny 1b - Deny
9 LP_FLEXCOMM9_LPDENY	LP_FLEXCOMM9 Low-Power Deny Status Indicates status of LP_FLEXCOMM9 to low-power request. 0b - Not deny 1b - Deny
8 LP_FLEXCOMM8_LPDENY	LP_FLEXCOMM8 Low-Power Deny Status Indicates status of LP_FLEXCOMM8 to low-power request. 0b - Not deny 1b - Deny
7 LP_FLEXCOMM7_LPDENY	LP_FLEXCOMM7 Low-Power Deny Status Indicates status of LP_FLEXCOMM7 to low-power request. 0b - Not deny 1b - Deny
6 LP_FLEXCOMM6_LPDENY	LP_FLEXCOMM6 Low-Power Deny Status Indicates status of LP_FLEXCOMM6 to low-power request. 0b - Not deny 1b - Deny
5	LP_FLEXCOMM5 Low-Power Deny Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM5_LPDENY	Indicates status of LP_FLEXCOMM5 to low-power request. 0b - Not deny 1b - Deny
4 LP_FLEXCOMM4_LPDENY	LP_FLEXCOMM4 Low-Power Deny Status Indicates status of LP_FLEXCOMM4 to low-power request. 0b - Not deny 1b - Deny
3 LP_FLEXCOMM3_LPDENY	LP_FLEXCOMM3 Low-Power Deny Status Indicates status of LP_FLEXCOMM3 to low-power request. 0b - Not deny 1b - Deny
2 LP_FLEXCOMM2_LPDENY	LP_FLEXCOMM2 Low-Power Deny Status Indicates status of LP_FLEXCOMM2 to low-power request. 0b - Not deny 1b - Deny
1 LP_FLEXCOMM1_LPDENY	LP_FLEXCOMM1 Low-Power Deny Status Indicates status of LP_FLEXCOMM1 to low-power request. 0b - Not deny 1b - Deny
0 LP_FLEXCOMM0_LPDENY	LP_FLEXCOMM0 Low-Power Deny Status Indicates status of LP_FLEXCOMM0 to low-power request. 0b - Not deny 1b - Deny

30.4.2 SLEEPCON1 register descriptions

30.4.2.1 SLEEPCON1 memory map

SLEEPCON1 base address: 4004_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Sleep Configuration (SLEEPCFG)	32	RW	8000_F402h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
10h	Run Configuration (RUNCFG)	32	RW	8000_F402h
20h	RUNCFG Set (RUNCFG_SET)	32	W	0000_0000h
30h	RUNCFG Clear (RUNCFG_CLR)	32	W	0000_0000h
60h	Wakeup Enable 0 (WAKEUPEN0)	32	RW	0000_0000h
64h	Wakeup Enable 1 (WAKEUPEN1)	32	RW	0000_0000h
68h	Wakeup Enable 2 (WAKEUPEN2)	32	RW	0000_0000h
A0h	Wakeup Enable 0 Set (WAKEUPEN0_SET)	32	W	0000_0000h
A4h	Wakeup Enable 1 Set (WAKEUPEN1_SET)	32	W	0000_0000h
A8h	Wakeup Enable 2 Set (WAKEUPEN2_SET)	32	W	0000_0000h
100h	Wakeup Enable 0 Clear (WAKEUPEN0_CLR)	32	W	0000_0000h
104h	Wakeup Enable 1 Clear (WAKEUPEN1_CLR)	32	W	0000_0000h
108h	Wakeup Enable 2 Clear (WAKEUPEN2_CLR)	32	W	0000_0000h
15Ch	LPOSC Delay (LPOSC_DELAY)	32	RW	0000_0014h
164h	Power Down Wait (PWRDOWN_WAIT)	32	RW	0000_0000h
190h	eDMA Wakeup Enable (HW_WAKE)	32	RW	0000_0000h
194h	eDMA Wakeup Set (HW_WAKE_SET)	32	W	0000_0000h
198h	eDMA Wakeup Clear (HW_WAKE_CLR)	32	W	0000_0000h
1A0h	Shared Resources Mask (SHARED_MASK0)	32	RW	0000_0000h
1A4h	Shared Resources Mask Set (SHARED_MASK0_SET)	32	W	0000_0000h
1A8h	Shared Resources Mask Clear (SHARED_MASK0_CLR)	32	W	0000_0000h
1B0h	Media Domain Shared Controller Low-power Control (SHA_MED_CCTRL0)	32	RW	0000_0000h
1B4h	Media Domain Shared Resources Target Low-power Control (SHA_MED_TCTRL0)	32	RW	0000_0000h
1BCh	Shared VDD1_SENSE Domain Targets Low-power Control (SHA_SEN_TCTRL0)	32	RW	0000_0000h
1C0h	Private Resources Controllers Low-power Control 0 (PRIVATE_CCTRL0)	32	RW	0000_0000h
1C4h	Private Resources Target Low-power Control (PRIVATE_TCTRL0)	32	RW	0000_0000h
1D0h	Media Domain Shared Controllers Low-power Status (SHA_MED_CSTAT0)	32	R	0000_0000h
1D4h	Media and Sense Domain Shared Targets Low-power Status (SHA_MEDSEN_TSTAT0)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
1E0h	Private Resources Controllers Low-power Status (PRIVATE_CSTAT0)	32	R	0000_0000h
1E4h	Private Resources Targets Low-power Status 0 (PRIVATE_TSTAT0)	32	R	0000_0000h
1ECh	Low-Power Hint Status 0 (LP_HINT0)	32	R	0000_0000h
1F4h	Low-Power Deny Status 0 (LP_DENY0)	32	R	0000_0000h

30.4.2.2 Sleep Configuration (SLEEPCFG)

Offset

Register	Offset
SLEEPCFG	0h

Function

Controls the power requests to various modules while CPU1 is in Deep Sleep mode. The power mode requests selected by this register are aggregated with the CPU0 requests to determine the actual settings. When the CPU1 is in Deep Power Down mode or Full Deep Power Down mode, all the related modules are powered down.

NOTE

You need to clear FRO2_GATE bit after POR and system reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FRO2_GA...	0														ADC0_PD
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUDPLLL...	AUDPLLA...	PLLLDO_...	PLLANA_...	LPOSC_PD	FRO2_PD	0		XTAL_PD	MEDIA_M...	COMN_MA...	RAM1_CL...	RAM0_CL...	SENSES_...	SENSEP_...	0
W																
Reset	1	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31	FRO2 Gating

Table continues on the next page...

Table continued from the previous page...

Field	Function
FRO2_GATE	Controls the gating of FRO2. 0b - Removes gating 1b - Applies gating
30-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Controls the power to ADC0. Write 1 to SHA_SEN_TCTRL0[ADC0_STOP] first and then write 1 to ADC0_PD to minimize the power consumption of ADC0. 0b - Powers on 1b - Powers down
15 AUDPLLLDO_PD	Audio PLL internal regulator Power Down Controls the power to audio PLL internal regulator. 0b - Powers on 1b - Powers down
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Controls the power to audio PLL analog function. 0b - Powers on 1b - Powers down
13 PLLLDO_PD	Main PLL internal regulator Power Down Controls the power to main PLL internal regulator. 0b - Powers on 1b - Powers down
12 PLLANA_PD	Main PLL Analog Function Power Down Controls the power to main PLL analog function. 0b - Powers on 1b - Powers down
11 LPOSC_PD	LPOSC Power Down Controls the power to LPOSC.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE LPOSC is used by WWDT and RSTCTL. WWDT module can use MOD[LOCK] to lock SLEEPCON's control over power to the LPOSC. If MOD[LOCK] of either or both of WWDT2 and WWDT3 is configured as 1, the write to this field can change the filed value but does not take effect. 0b - Powers on 1b - Powers down
10 FRO2_PD	FRO2 Power Down Controls the power to FRO2. **NOTE** FRO2 power-down depends on both FRO2_PD and CSR[FROEN] of FRO2. To power down FRO2, you need to write 1 to FRO2_PD and write 0 to CSR[FROEN] of FRO2. To Power on FRO2, you need to write 0 to FRO2_PD or write 1 to CSR[FROEN] of FRO2. If SLEEPCON needs to wait power down ready from FRO2, then CSR[CLKGATE] of FRO2 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
9-8 —	Reserved
7 XTAL_PD	Xtal Oscillator Power Down Controls the power to Xtal oscillator (osc_out). 0b - Powers on 1b - Powers down
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Controls the main clock shut-off of the VDD2_MEDIA and VDDN_MEDIA domains. 0b - Turns on 1b - Turns off
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Controls the main clock shut-off of the VDDN_COM domain. 0b - Turns on 1b - Turns off
4	RAM_ARBITER1 sense_ram_clk Shut Off

Table continues on the next page...

Table continued from the previous page...

Field	Function
RAM1_CLK_SHUTOFF	Controls the clock shut-off of the RAM_ARBITER1. 0b - Turns on 1b - Turns off
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER0 in VDD2_COM domain. 0b - Turns on 1b - Turns off
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Controls the main clock shut-off of the private part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
0 —	Reserved

30.4.2.3 Run Configuration (RUNCFG)

Offset

Register	Offset
RUNCFG	10h

Function

Controls the power to various modules during normal operation. The power mode requests selected by this register are aggregated with the CPU0 requests to determine the actual settings.

To modify a RUNCFG bit, use RUNCFG_CLR and RUNCFG_SET:

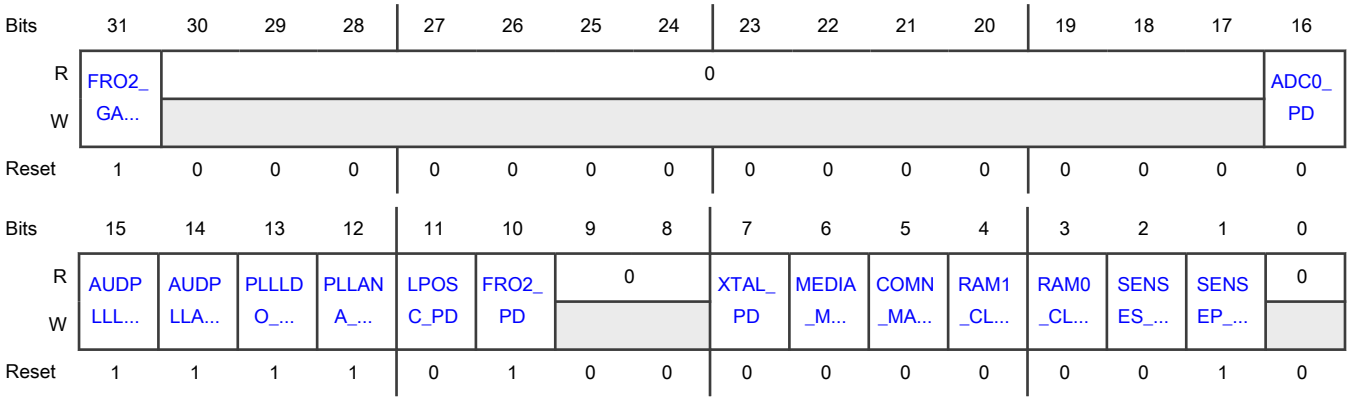
- To make it 0, write a 1 to the corresponding bit position in RUNCFG_CLR.
- To make it 1, write a 1 to the corresponding bit position in RUNCFG_SET.

NOTE

After confirming that a specific FRO*/LPOSC/Xtal/PLLs is not used, you can configure FRO*_PD and *PLL*_PD bits in RUNCFG by RUNCFG_CLR. Then FRO*/LPOSC/Xtal/PLLs can power down on the fly.

You need to clear FRO2_GATE bit after POR and system reset.

Diagram



Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Controls the gating of FRO2. 0b - Removes gating 1b - Applies gating
30-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Controls the power to the ADC0. Write 1 to SHA_SEN_TCTRL0[ADC0_STOP] first and then write 1 to ADC0_PD to minimize the power consumption of ADC0. 0b - Powers on 1b - Powers down
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Controls the power to audio PLL internal regulator. 0b - Powers on 1b - Powers down
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Controls the power to audio PLL analog function. 0b - Powers on 1b - Powers down
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Controls the power to main PLL internal regulator.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Powers on 1b - Powers down
12 PLLANA_PD	Main PLL Analog Function Power Down Controls the power to main PLL analog function. 0b - Powers on 1b - Powers down
11 LPOSC_PD	LPOSC Power Down Controls the power to LPOSC. **NOTE** LPOSC is used by WWDT and RSTCTL. WWDT module can use MOD[LOCK] to lock SLEEPCON's control over power to the LPOSC. If MOD[LOCK] of either or both of WWDT2 and WWDT3 is configured as 1, the write to this field can change the filed value but does not take effect. 0b - Powers on 1b - Powers down
10 FRO2_PD	FRO2 Power Down Controls the power to FRO2. **NOTE** FRO2 power-down depends on both FRO2_PD and CSR[FROEN] of FRO2. To power down FRO2, you need to write 1 to FRO2_PD and write 0 to CSR[FROEN] of FRO2. To Power on FRO2, you need to write 0 to FRO2_PD or write 1 to CSR[FROEN] of FRO2. If SLEEPCON needs to wait power down ready from FRO2, then CSR[CLKGATE] of FRO2 must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode. 0b - Powers on 1b - Powers down
9-8 —	Reserved
7 XTAL_PD	Xtal Oscillator Power Down Controls the power to Xtal oscillator (osc_out). 0b - Powers on 1b - Powers down
6	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEDIA_MAINCLK_SHUTOFF	Controls the main clock shut-off of the VDD2_MEDIA and VDDN_MEDIA domains. 0b - Turns on 1b - Turns off
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Controls the main clock shut-off of the VDDN_COM domain. 0b - Turns on 1b - Turns off
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER1 in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Controls the clock shut-off of the RAM_ARBITER0 in VDD2_COM domain. 0b - Turns on 1b - Turns off
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Controls the clock shut-off of the shared part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Controls the main clock shut-off of the private part in VDD1_SENSE domain. 0b - Turns on 1b - Turns off
0 —	Reserved

30.4.2.4 RUNCFG Set (RUNCFG_SET)

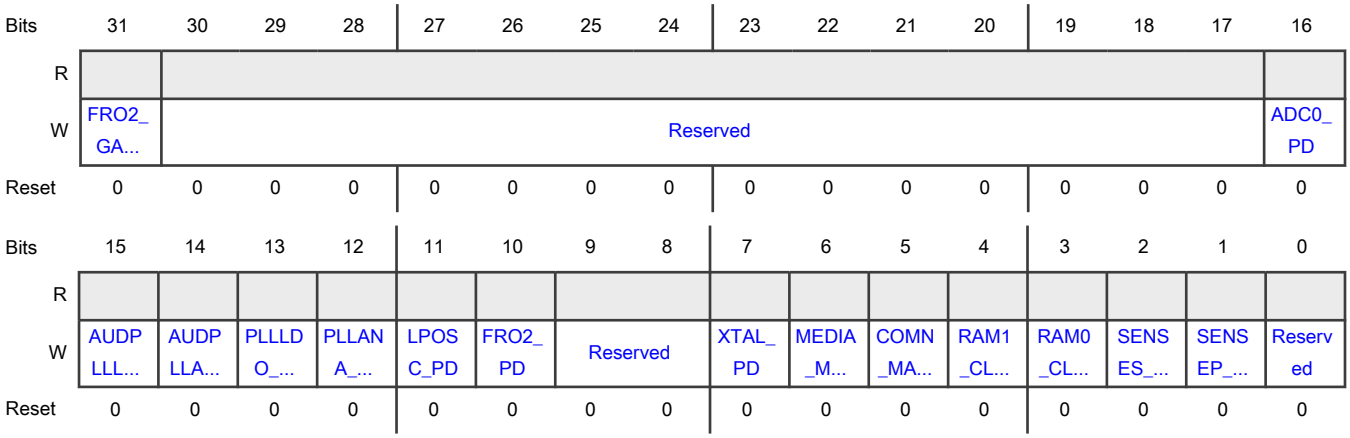
Offset

Register	Offset
RUNCFG_SET	20h

Function

Writing a 1 to a bit position in this register sets the corresponding position in RUNCFG.

Diagram



Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13	Main PLL Internal Regulator Power Down Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
PLLLDO_PD	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12 PLLANA_PD	Main PLL Analog Function Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 LPOSC_PD	LPOSC Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 FRO2_PD	FRO2 Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9-8 —	Reserved
7 XTAL_PD	Xtal Oscillator Power Down Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
4	RAM_ARBITER1 sense_ram_clk Shut Off Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
RAM1_CLK_SHUTOFF	0b - No effect 1b - Sets the bit
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.2.5 RUNCFG Clear (RUNCFG_CLR)

Offset

Register	Offset
RUNCFG_CLR	30h

Function

Writing a 1 to a bit position in this register clears the corresponding position in the RUNCFG register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	FRO2_GA...	Reserved														ADC0_PD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	AUDPLLL...	AUDPLLA...	PLLLDO_...	PLLANA_...	LPOSC_PD	FRO2_PD	Reserved		XTAL_PD	MEDIA_M...	COMN_MA...	RAM1_CL...	RAM0_CL...	SENSES_...	SENSEP_...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 FRO2_GATE	FRO2 Gating Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
30-17 —	Reserved
16 ADC0_PD	ADC0 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15 AUDPLLLDO_PD	Audio PLL Internal Regulator Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 AUDPLLANA_PD	Audio PLL Analog Function Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 PLLLDO_PD	Main PLL Internal Regulator Power Down Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
12 PLLANA_PD	Main PLL Analog Function Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 LPOSC_PD	LPOSC Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 FRO2_PD	FRO2 Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9-8 —	Reserved
7 XTAL_PD	Xtal Oscillator Power Down Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 MEDIA_MAINCLK_SHUTOFF	VDD2_MEDIA and VDDN_MEDIA Domains media_main_clk and media_vddn_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 COMN_MAINCLK_SHUTOFF	VDDN_COM Domain common_vddn_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 RAM1_CLK_SHUTOFF	RAM_ARBITER1 sense_ram_clk Shut Off Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
3 RAM0_CLK_SHUTOFF	RAM_ARBITER0 common_ram_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 SENSES_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk_1 Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 SENSEP_MAINCLK_SHUTOFF	VDD1_SENSE Domain sense_main_clk Shut Off Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.2.6 Wakeup Enable 0 (WAKEUPEN0)

Offset

Register	Offset
WAKEUPEN0	60h

Function

Enables the CPU1 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU1 will wake up when receiving the module's interrupt.

To modify a WAKEUPEN0 bit, use the corresponding WAKEUPEN0_CLR and WAKEUPEN0_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN0_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN0_SET.

Using WAKEUPEN0_CLR and WAKEUPEN0_SET prevents unintentionally setting or clearing other bits

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		OSTIMER	PMC1	MU3_MUA	MU2_MUB	MU1_MUB		RTC1_WK...	RTC1_AL...	MICFIL_...			MICFIL_...	ACMP0		
W	0						0				0				0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC0	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LPI2C15	CTIMER7	CTIMER6	CTIMER5		UTICK1		FRO2	FRO1	FRO0	0
W										0		0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 OSTIMER	OSTIMER Wake-up Enable Enables CPU1 to wake-up on OSTIMER interrupt. 0b - Disables 1b - Enables
29 PMC1	PMC1 Wake-up Enable Enables CPU1 to wake-up on PMC1 interrupt. 0b - Disables 1b - Enables
28 MU3_MUA	MU3_MUA Wake-up Enable Enables CPU1 to wake-up on MU3_MUA interrupt. 0b - Disables 1b - Enables
27 MU2_MUB	MU2_MUB Wake-up Enable Enables CPU1 to wake-up on MU2_MUB interrupt. 0b - Disables 1b - Enables
26 MU1_MUB	MU1_MUB Wake-up Enable Enables CPU1 to wake-up on MU1_MUB interrupt. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 —	Reserved
24 RTC1_WKUP	RTC1_WKUP Wake-up Enable Enables CPU1 to wake-up on RTC1 wakeup interrupt to VDD1_SENSE domain. 0b - Disables 1b - Enables
23 RTC1_ALARM	RTC1_ALARM Wake-up Enable Enables CPU1 to wake-up on RTC1 alarm interrupt to VDD1_SENSE domain. 0b - Disables 1b - Enables
22 MICFIL_WKUP	MICFIL Wake-up Enable Enables CPU1 to wake-up on MICFIL hardware voice activity detector interrupt or error interrupt. 0b - Disables 1b - Enables
21-20 —	Reserved
19 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Enables CPU1 to wake-up on MICFIL interrupt for read data or error. 0b - Disables 1b - Enables
18 ACMP0	ACMP0 Wake-up Enable Enables CPU1 to wake-up on ACMP0 interrupt. 0b - Disables 1b - Enables
17-16 —	Reserved
15 ADC0	ADC0 Wake-up Enable Enables CPU1 to wake-up on ADC0 interrupt. 0b - Disables 1b - Enables
14	LP_FLEXCOMM20 Wake-up Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM20	Enables CPU1 to wake-up on LP_FLEXCOMM20 interrupt. 0b - Disables 1b - Enables
13 LP_FLEXCOMM19	LP_FLEXCOMM19 Wake-up Enable Enables CPU1 to wake-up on LP_FLEXCOMM19 interrupt. 0b - Disables 1b - Enables
12 LP_FLEXCOMM18	LP_FLEXCOMM18 Wake-up Enable Enables CPU1 to wake-up on LP_FLEXCOMM18 interrupt. 0b - Disables 1b - Enables
11 LP_FLEXCOMM17	LP_FLEXCOMM17 Wake-up Enable Enables CPU1 to wake-up on LP_FLEXCOMM17 interrupt. 0b - Disables 1b - Enables
10 LPI2C15	LPI2C15 Wake-up Enable Enables CPU1 to wake-up on LPI2C15 interrupt. 0b - Disables 1b - Enables
9 CTIMER7	CTIMER7 Wake-up Enable Enables CPU1 to wake-up on CTIMER7 interrupt. 0b - Disables 1b - Enables
8 CTIMER6	CTIMER6 Wake-up Enable Enables CPU1 to wake-up on CTIMER6 interrupt. 0b - Disables 1b - Enables
7 CTIMER5	CTIMER5 Wake-up Enable Enables CPU1 to wake-up on CTIMER5 interrupt. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
6 —	Reserved
5 UTICK1	UTICK1 Wake-up Enable Enables CPU1 to wake-up on UTICK1 interrupt. 0b - Disables 1b - Enables
4 —	Reserved
3 FRO2	FRO2 Wake-up Enable Enables CPU1 to wake-up on FRO2 Async interrupt. 0b - Disables 1b - Enables
2 FRO1	FRO1 Wake-up Enable Enables CPU1 to wake-up on FRO1 Async interrupt. 0b - Disables 1b - Enables
1 FRO0	FRO0 Wake-up Enable Enables CPU1 to wake-up on FRO0 Async interrupt. 0b - Disables 1b - Enables
0 —	Reserved

30.4.2.7 Wakeup Enable 1 (WAKEUPEN1)

Offset

Register	Offset
WAKEUPEN1	64h

Function

Enables the CPU1 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU1 will wake up when receiving the module's wakeup interrupt.

To modify a WAKEUPEN1 bit, use the corresponding WAKEUPEN1_CLR and WAKEUPEN1_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN1_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN1_SET.

Using WAKEUPEN1_CLR and WAKEUPEN1_SET prevents unintentionally setting or clearing other bits

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPIO9_C...	GPIO8_C...	GPIO8_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDMA2_C...	EDMA2_C...	EDMA2_C...				FLEXIO	I3C3	PMIC_IR...	USBPHY0	WWDT3	WWDT2	USB1	USB0	I3C2	
W				0												0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO9_CH0	GPIO9 Channel 0 Wake-up Enable Enables CPU1 to wake-up on GPIO9 channel 0 interrupt. 0b - Disables 1b - Enables
30 GPIO8_CH1	GPIO8 Channel 1 Wake-up Enable Enables CPU1 to wake-up on GPIO8 channel 1 interrupt. 0b - Disables 1b - Enables
29 GPIO8_CH0	GPIO8 Channel 0 Wake-up Enable Enables CPU1 to wake-up on GPIO8 channel 0 interrupt. 0b - Disables 1b - Enables
28 EDMA3_CH7	eDMA3 Channel 7 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 7 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
27 EDMA3_CH6	eDMA3 Channel 6 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 6 interrupt only when DMA wakeup is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
26 EDMA3_CH5	eDMA3 Channel 5 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 5 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
25 EDMA3_CH4	eDMA3 Channel 4 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 4 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
24 EDMA3_CH3	eDMA3 Channel 3 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 3 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
23 EDMA3_CH2	eDMA3 Channel 2 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 2 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
22 EDMA3_CH1	eDMA3 Channel 1 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 1 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
21 EDMA3_CH0	eDMA3 Channel 0 Wake-up Enable Enables CPU1 to wake-up on eDMA3 channel 0 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
20 EDMA2_CH7	eDMA2 Channel 7 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 7 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
19	eDMA2 Channel 6 Wake-up Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA2_CH6	Enables CPU1 to wake-up on eDMA2 channel 6 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
18 EDMA2_CH5	eDMA2 Channel 5 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 5 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
17 EDMA2_CH4	eDMA2 Channel 4 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 4 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
16 EDMA2_CH3	eDMA2 Channel 3 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 3 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
15 EDMA2_CH2	eDMA2 Channel 2 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 2 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
14 EDMA2_CH1	eDMA2 Channel 1 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 1 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
13 EDMA2_CH0	eDMA2 Channel 0 Wake-up Enable Enables CPU1 to wake-up on eDMA2 channel 0 interrupt only when DMA wakeup is enabled. 0b - Disables 1b - Enables
12-10 —	Reserved
9 FLEXIO	FLEXIO Wake-up Enable Enables CPU1 to wake-up on FLEXIO interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
8 I3C3	I3C3 Wake-up Enable Enables CPU1 to wake-up on I3C3 interrupt. 0b - Disables 1b - Enables
7 PMIC_IRQN	PMIC_IRQN Wake-up Enable Enables CPU1 to wake-up on external PMIC_IRQN interrupt. 0b - Disables 1b - Enables
6 USBPHY0	USBPHY0 Wake-up Enable Enables CPU1 to wake-up on USBPHY0 interrupt. 0b - Disables 1b - Enables
5 WWDT3	WWDT3 Wake-up Enable Enables CPU1 to wake-up on WWDT3 interrupt. 0b - Disables 1b - Enables
4 WWDT2	WWDT2 Wake-up Enable Enables CPU1 to wake-up on WWDT2 interrupt. 0b - Disables 1b - Enables
3 USB1	USB1 Wake-up Enable Enables CPU1 to wake-up on USB1 interrupt. 0b - Disables 1b - Enables
2 USB0	USB0 Wake-up Enable Enables CPU1 to wake-up on USB0 interrupt. 0b - Disables 1b - Enables
1	I3C2 Wake-up Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
I3C2	Enables CPU1 to wake-up on I3C2 interrupt. 0b - Disables 1b - Enables
0 —	Reserved

30.4.2.8 Wakeup Enable 2 (WAKEUPEN2)

Offset

Register	Offset
WAKEUPEN2	68h

Function

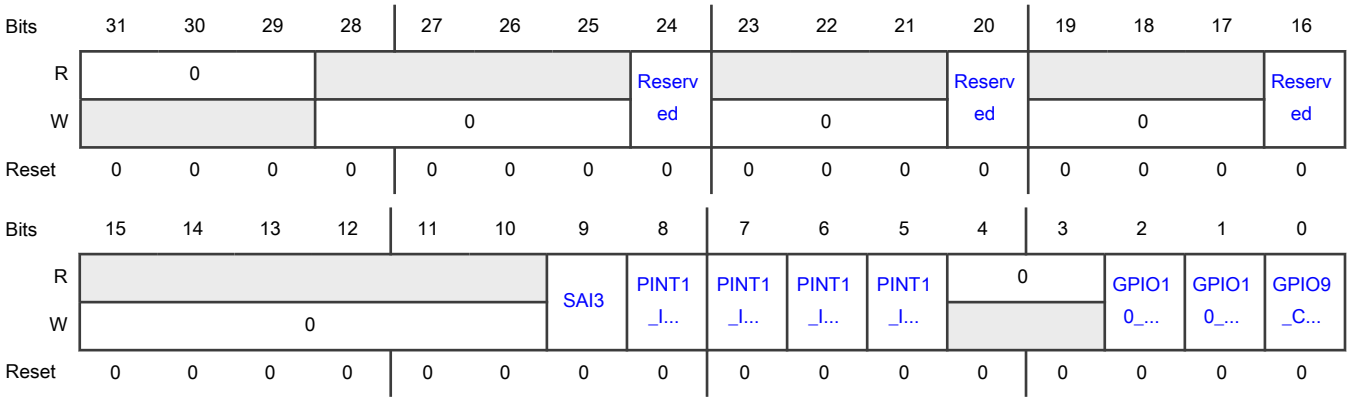
Enables the CPU1 to wake-up from deep-sleep mode. If a module's wakeup field is enabled, the CPU1 will wake up when receiving the module's wakeup interrupt.

To modify a WAKEUPEN2 bit, use the corresponding WAKEUPEN2_CLR and WAKEUPEN2_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in WAKEUPEN2_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in WAKEUPEN2_SET.

Using WAKEUPEN2_CLR and WAKEUPEN2_SET prevents unintentionally setting or clearing other bits

Diagram



Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
28-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 —	Reserved
15-10 —	Reserved
9 SAI3	SAI3 Wake-up Enable Enables CPU1 to wake-up on SAI3 interrupt. 0b - Disables 1b - Enables
8 PINT1_INTR3	PINT1_INTR3 Wake-up Enable Enables CPU1 to wake-up on PINT1 interrupt 3. 0b - Disables 1b - Enables
7 PINT1_INTR2	PINT1_INTR2 Wake-up Enable Enables CPU1 to wake-up on PINT1 interrupt 2. 0b - Disables 1b - Enables
6 PINT1_INTR1	PINT1_INTR1 Wake-up Enable Enables CPU1 to wake-up on PINT1 interrupt 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
5 PINT1_INTR0	PINT1_INTR0 Wake-up Enable Enables CPU1 to wake-up on PINT1 interrupt 0. 0b - Disables 1b - Enables
4-3 —	Reserved
2 GPIO10_CH1	GPIO10 Channel 1 Wake-up Enable Enables CPU1 to wake-up on GPIO10 channel 1 interrupt. 0b - Disables 1b - Enables
1 GPIO10_CH0	GPIO10 Channel 0 Wake-up Enable Enables CPU1 to wake-up on GPIO10 channel 0 interrupt. 0b - Disables 1b - Enables
0 GPIO9_CH1	GPIO9 Channel 1 Wake-up Enable Enables CPU1 to wake-up on GPIO9 channel 1 interrupt. 0b - Disables 1b - Enables

30.4.2.9 Wakeup Enable 0 Set (WAKEUPEN0_SET)

Offset

Register	Offset
WAKEUPEN0_SET	A0h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN0. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0	OSTIMER	PMC1	MU3_MUA	MU2_MUB	MU1_MUB	0	RTC1_WK...	RTC1_AL...	MICFIL_...	0		MICFIL_...	ACMP0	0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	ADC0	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LPI2C15	CTIMER7	CTIMER6	CTIMER5	0	UTICK1	0	FRO2	FRO1	FRO0	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 OSTIMER	OSTIMER Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
29 PMC1	PMC1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28 MU3_MUA	MU3_MUA Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
27 MU2_MUB	MU2_MUB Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
26 MU1_MUB	MU1_MUB Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
25 —	Reserved
24 RTC1_WKUP	RTC1_WKUP Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
23 RTC1_ALARM	RTC1_ALARM Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 MICFIL_WKUP	MICFIL Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21-20 —	Reserved
19 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 ACMP0	ACMP0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17-16 —	Reserved
15 ADC0	ADC0 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
14 LP_FLEXCOMM20	LP_FLEXCOMM20 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 LP_FLEXCOMM19	LP_FLEXCOMM19 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12 LP_FLEXCOMM18	LP_FLEXCOMM18 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
11 LP_FLEXCOMM17	LP_FLEXCOMM17 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
10 LPI2C15	LPI2C15 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
9 CTIMER7	CTIMER7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
8 CTIMER6	CTIMER6 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7	CTIMER5 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER5	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 —	Reserved
5 UTICK1	UTICK1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
4 —	Reserved
3 FRO2	FRO2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 FRO1	FRO1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 FRO0	FRO0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.2.10 Wakeup Enable 1 Set (WAKEUPEN1_SET)

Offset

Register	Offset
WAKEUPEN1_SET	A4h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN1. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	GPIO9_C...	GPIO8_C...	GPIO8_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	EDMA2_C...	EDMA2_C...	EDMA2_C...	0			FLEXIO	I3C3	PMIC_IR...	USBPHY0	WWDT3	WWDT2	USB1	USB0	I3C2	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO9_CH0	GPIO9 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
30 GPIO8_CH1	GPIO8 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
29 GPIO8_CH0	GPIO8 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
28 EDMA3_CH7	eDMA3 Channel 7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
27 EDMA3_CH6	eDMA3 Channel 6 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
26 EDMA3_CH5	eDMA3 Channel 5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
25 EDMA3_CH4	eDMA3 Channel 4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
24 EDMA3_CH3	eDMA3 Channel 3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
23 EDMA3_CH2	eDMA3 Channel 2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 EDMA3_CH1	eDMA3 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21 EDMA3_CH0	eDMA3 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20 EDMA2_CH7	eDMA2 Channel 7 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19	eDMA2 Channel 6 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA2_CH6	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
18 EDMA2_CH5	eDMA2 Channel 5 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 EDMA2_CH4	eDMA2 Channel 4 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16 EDMA2_CH3	eDMA2 Channel 3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15 EDMA2_CH2	eDMA2 Channel 2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
14 EDMA2_CH1	eDMA2 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
13 EDMA2_CH0	eDMA2 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
12-10 —	Reserved
9 FLEXIO	FLEXIO Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
8 I3C3	I3C3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7 PMIC_IRQN	PMIC_IRQN Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 USBPHY0	USBPHY0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 WWDT3	WWDT3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
4 WWDT2	WWDT2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3 USB1	USB1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 USB0	USB0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1	I3C2 Wake-up Enable Set

Table continues on the next page...

Table continued from the previous page...

Field	Function
I3C2	Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.2.11 Wakeup Enable 2 Set (WAKEUPEN2_SET)

Offset

Register	Offset
WAKEUPEN2_SET	A8h

Function

Writing a 1 to a bit position sets the corresponding position in WAKEUPEN2. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved				0			Reserved	0			Reserved	0			Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0						SAI3	PINT1_I...	PINT1_I...	PINT1_I...	PINT1_I...	Reserved		GPIO10_...	GPIO10_...	GPIO9_C...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28-25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 —	Reserved
15-10 —	Reserved
9 SAI3	SAI3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
8 PINT1_INTR3	PINT1_INTR3 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
7 PINT1_INTR2	PINT1_INTR2 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
6 PINT1_INTR1	PINT1_INTR1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 PINT1_INTR0	PINT1_INTR0 Wake-up Enable Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
4-3 —	Reserved
2 GPIO10_CH1	GPIO10 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 GPIO10_CH0	GPIO10 Channel 0 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 GPIO9_CH1	GPIO9 Channel 1 Wake-up Enable Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.2.12 Wakeup Enable 0 Clear (WAKEUPEN0_CLR)

Offset

Register	Offset
WAKEUPEN0_CLR	100h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN0 register. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0	OSTIMER	PMC1	MU3_MUA	MU2_MUB	MU1_MUB	0	RTC1_WK...	RTC1_AL...	MICFIL_...	0		MICFIL_...	ACMP0	0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	ADC0	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LPI2C15	CTIMER7	CTIMER6	CTIMER5	0	UTICK1	0	FRO2	FRO1	FRO0	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 OSTIMER	OSTIMER Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
29 PMC1	PMC1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28 MU3_MUA	MU3_MUA Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
27 MU2_MUB	MU2_MUB Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
26 MU1_MUB	MU1_MUB Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
25 —	Reserved
24 RTC1_WKUP	RTC1_WKUP Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
23 RTC1_ALARM	RTC1_ALARM Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 MICFIL_WKUP	MICFIL Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21-20 —	Reserved
19 MICFIL_RD_ERR	MICFIL Read Data Error Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 ACMP0	ACMP0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17-16 —	Reserved
15 ADC0	ADC0 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
14 LP_FLEXCOMM20	LP_FLEXCOMM20 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 LP_FLEXCOMM19	LP_FLEXCOMM19 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12 LP_FLEXCOMM18	LP_FLEXCOMM18 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
11 LP_FLEXCOMM17	LP_FLEXCOMM17 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
10 LPI2C15	LPI2C15 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
9 CTIMER7	CTIMER7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 CTIMER6	CTIMER6 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7	CTIMER5 Wake-up Enable Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTIMER5	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 —	Reserved
5 UTICK1	UTICK1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 —	Reserved
3 FRO2	FRO2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 FRO1	FRO1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 FRO0	FRO0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.2.13 Wakeup Enable 1 Clear (WAKEUPEN1_CLR)

Offset

Register	Offset
WAKEUPEN1_CLR	104h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN1 register. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	GPIO9_C...	GPIO8_C...	GPIO8_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA3_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...	EDMA2_C...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	EDMA2_C...	EDMA2_C...	EDMA2_C...	0			FLEXIO	I3C3	PMIC_IR...	USBPHY0	WWDT3	WWDT2	USB1	USB0	I3C2	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 GPIO9_CH0	GPIO9 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
30 GPIO8_CH1	GPIO8 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
29 GPIO8_CH0	GPIO8 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
28 EDMA3_CH7	eDMA3_INTR7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
27 EDMA3_CH6	eDMA3 Channel 6 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
26 EDMA3_CH5	eDMA3 Channel 5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
25 EDMA3_CH4	eDMA3 Channel 4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
24 EDMA3_CH3	eDMA3 Channel 3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
23 EDMA3_CH2	eDMA3 Channel 2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 EDMA3_CH1	eDMA3 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21 EDMA3_CH0	eDMA3 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 EDMA2_CH7	eDMA2 Channel 7 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19	eDMA2 Channel 6 Wake-up Enable Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
EDMA2_CH6	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 EDMA2_CH5	eDMA2 Channel 5 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 EDMA2_CH4	eDMA2 Channel 4 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16 EDMA2_CH3	eDMA2 Channel 3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15 EDMA2_CH2	eDMA2 Channel 2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
14 EDMA2_CH1	eDMA2 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
13 EDMA2_CH0	eDMA2 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
12-10 —	Reserved
9 FLEXIO	FLEXIO Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
8 I3C3	I3C3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 PMIC_IRQN	PMIC_IRQN Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 USBPHY0	USBPHY0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 WWDT3	WWDT3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
4 WWDT2	WWDT2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 USB1	USB1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 USB0	USB0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1	I3C2 Wake-up Enable Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
I3C2	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.2.14 Wakeup Enable 2 Clear (WAKEUPEN2_CLR)

Offset

Register	Offset
WAKEUPEN2_CLR	108h

Function

Writing a 1 to a bit position clears the corresponding position in the WAKEUPEN2 register. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved				0			Reserved	0			Reserved	0			Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0						SAI3	PINT1_I...	PINT1_I...	PINT1_I...	PINT1_I...	Reserved		GPIO10_...	GPIO10_...	GPIO9_C...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28-25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 —	Reserved
15-10 —	Reserved
9 SAI3	SAI3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
8 PINT1_INTR3	PINT1_INTR3 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
7 PINT1_INTR2	PINT1_INTR2 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
6 PINT1_INTR1	PINT1_INTR1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 PINT1_INTR0	PINT1_INTR0 Wake-up Enable Clear Changes the corresponding bit to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the bit
4-3 —	Reserved
2 GPIO10_CH1	GPIO10 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 GPIO10_CH0	GPIO10 Channel 0 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 GPIO9_CH1	GPIO9 Channel 1 Wake-up Enable Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.2.15 LPOSC Delay (LPOSC_DELAY)

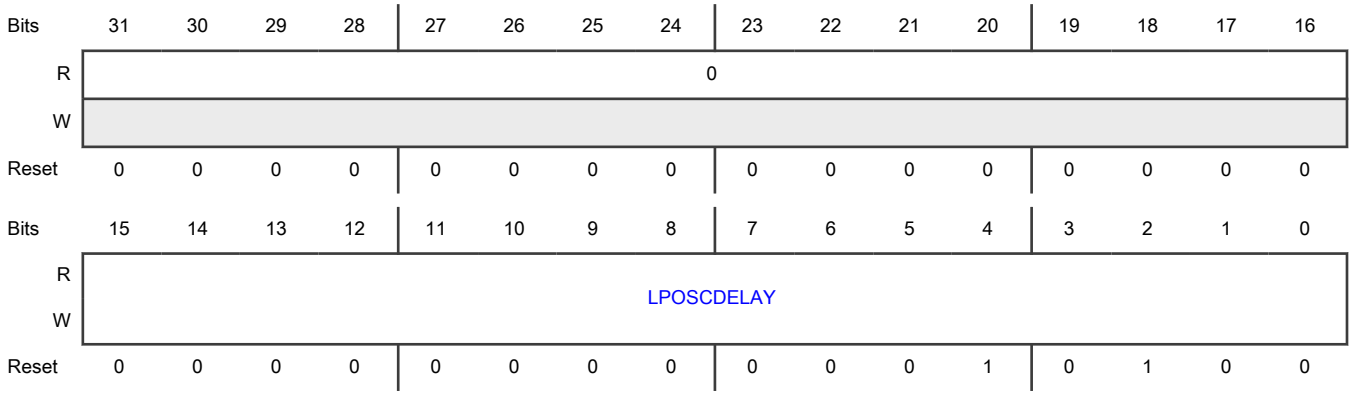
Offset

Register	Offset
LPOSC_DELAY	15Ch

Function

Controls the delay for LPOSC to output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 LPOSCDELAY	Delay value for LPOSC output Controls the delay in units of LPOSC clock cycles. 0000h - No delay 0001h - 1 LPOSC clock cycle 0002h - 2 LPOSC clock cycles ..... FFFFh - 2^{16}-1 LPOSC clock cycles

30.4.2.16 Power Down Wait (PWRDOWN_WAIT)

Offset

Register	Offset
PWRDOWN_WAIT	164h

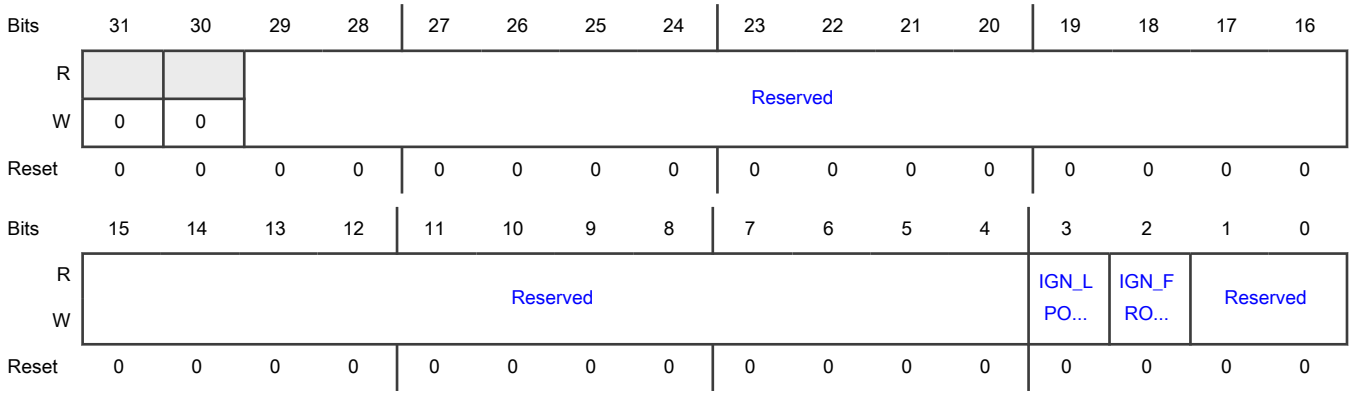
Function

Controls the power down wait of FRO2 and LPOSC. Setting the ignore bits below will allow the CPU1 to continue entering a low-power mode without waiting for the corresponding clock to shut down first.

NOTE

If SLEEPCON needs to wait power down ready from FRO, then CSR[CLKGATE] of FRO must enable FRO divider 6 clock. Otherwise SLEEPCON will be blocked and PMC cannot entry low-power mode.

Diagram



Fields

Field	Function
31 —	Reserved
30 —	Reserved
29-4 —	Reserved
3 IGN_LPOSCPDR	Ignore LPOSC Power Down Ready Signal Ignores power-down ready signal from LPOSC. 0b - Waits 1b - Ignores
2 IGN_FRO2PDR	Ignore FRO2 Power Down Ready Signal Ignores power-down ready signal from FRO2. 0b - Waits 1b - Ignores
1-0 —	Reserved

30.4.2.17 eDMA Wakeup Enable (HW_WAKE)

Offset

Register	Offset
HW_WAKE	190h

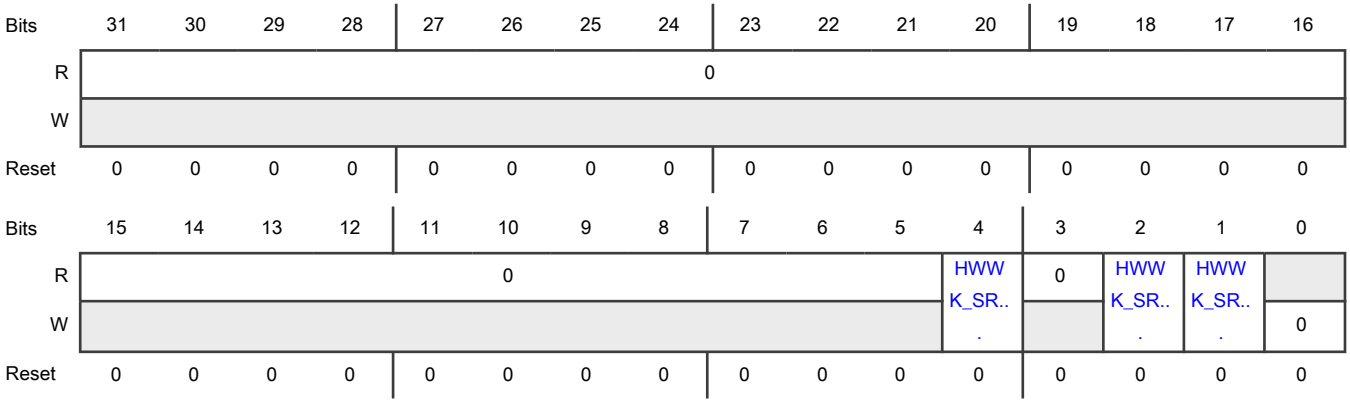
Function

Enables eDMA (eDMA2 and eDMA3) wakeup sources. To modify an HW_WAKE bit, use the HW_WAKE_CLR and HW_WAKE_SET registers:

- To make a bit 0, write a 1 to the corresponding bit position in HW_WAKE_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in HW_WAKE_SET.

Using HW_WAKE_CLR and HW_WAKE_SET prevents unintentionally setting or clearing other bits.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4	eDMA Wakeup Source 4 Enables eDMA to wakeup by SAI3 DMA request. 0b - Disables 1b - Enables
3 —	Reserved
2 HWWK_SRC2	eDMA Wakeup Source 2 Enables eDMA to wakeup by LP_FLEXCOMM17-20 DMA request. 0b - Disables 1b - Enables
1 HWWK_SRC1	eDMA Wakeup Source 1 Enables eDMA to wakeup by FLEXIO DMA request. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 —	Reserved

30.4.2.18 eDMA Wakeup Set (HW_WAKE_SET)

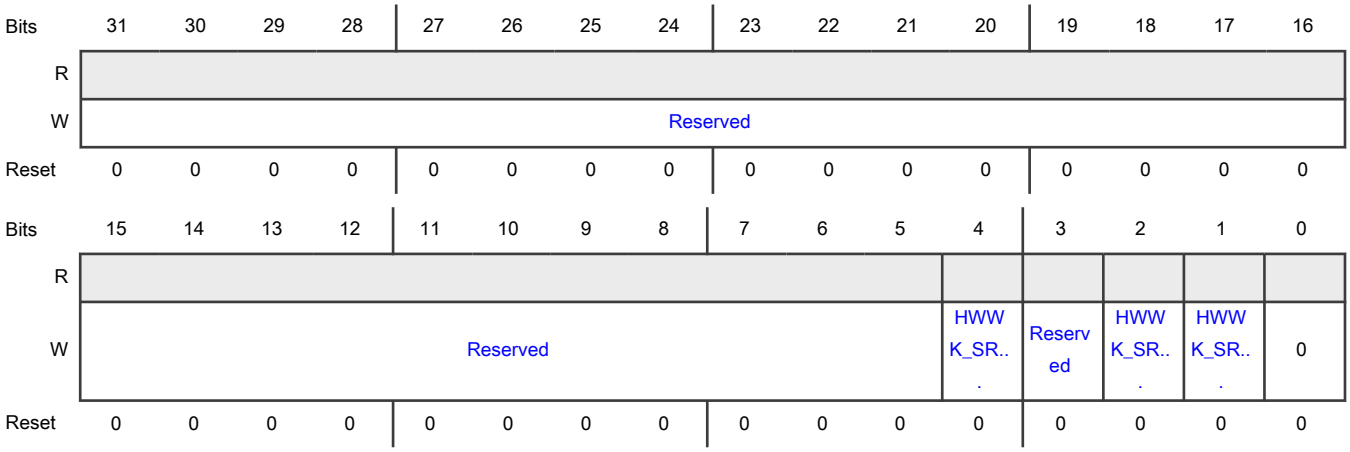
Offset

Register	Offset
HW_WAKE_SET	194h

Function

Writing a 1 to a bit position sets the corresponding position in HW_WAKE. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4_SET	eDMA Wakeup Source 4 Enable Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 HWWK_SRC2_SET	eDMA Wakeup Source 2 Enable Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
1 HWWK_SRC1_SET	eDMA Wakeup Source 1 Enable Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 —	Reserved

30.4.2.19 eDMA Wakeup Clear (HW_WAKE_CLR)

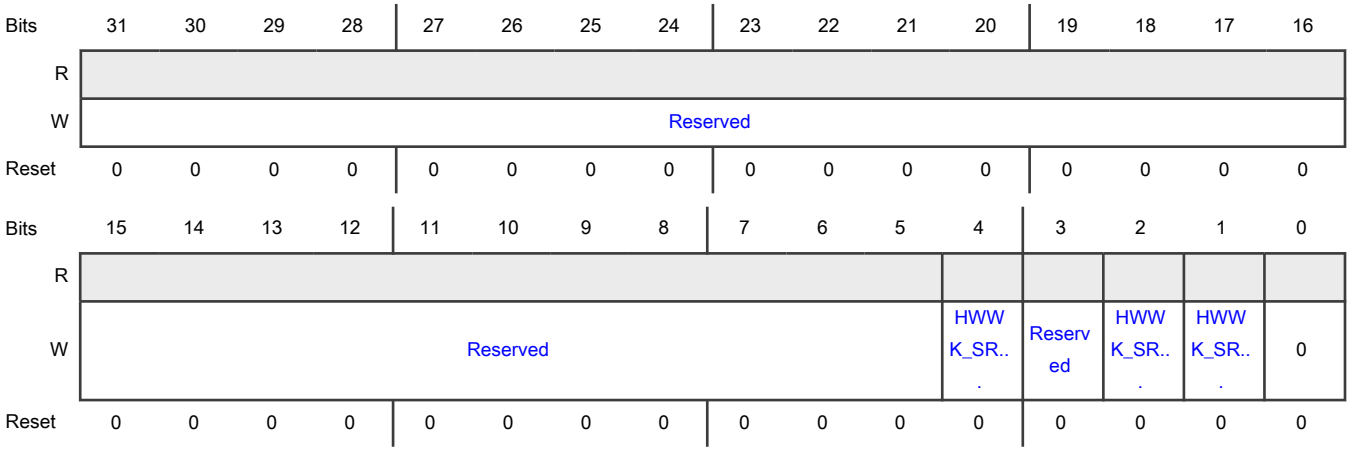
Offset

Register	Offset
HW_WAKE_CLR	198h

Function

Writing a 1 to a bit position clears the corresponding position in HW_WAKE. Writing a 0 to a bit position has no effect.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 HWWK_SRC4_CLR	eDMA Wakeup Source 4 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
3 —	Reserved
2 HWWK_SRC2_CLR	eDMA Wakeup Source 2 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 HWWK_SRC1_CLR	eDMA Wakeup Source 1 Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 —	Reserved

30.4.2.20 Shared Resources Mask (SHARED_MASK0)

Offset

Register	Offset
SHARED_MASK0	1A0h

Function

Enables the masking of SLEEPCON1 control for shared modules. To modify a SHARED_MASK0 bit, use the corresponding SHARED_MASK0_CLR and SHARED_MASK0_SET registers:

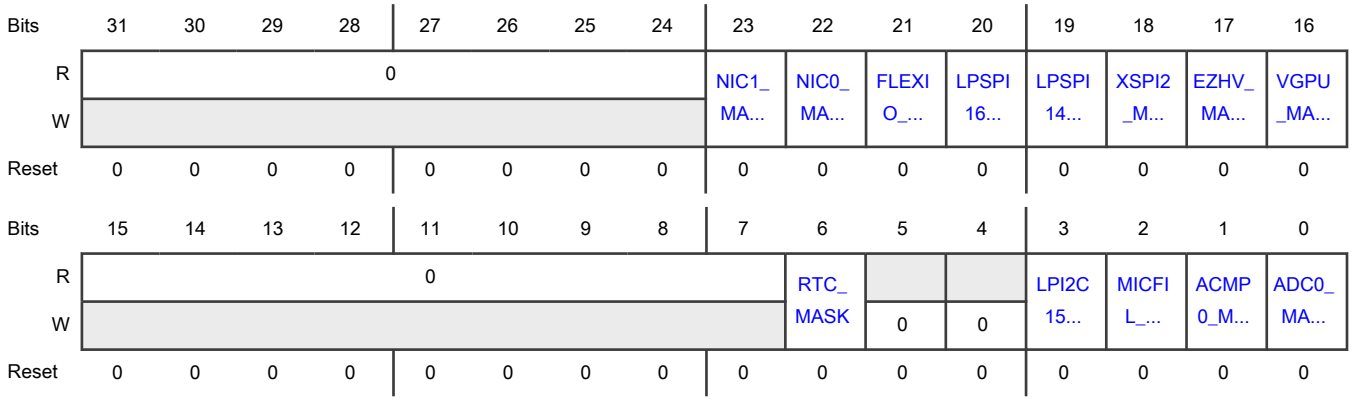
- To make a bit 0, write a 1 to the corresponding bit position in SHARED_MASK0_CLR.
- To make a bit 1, write a 1 to the corresponding bit position in SHARED_MASK0_SET.

Using SHARED_MASK0_CLR and SHARED_MASK0_SET prevents unintentionally setting or clearing other bits.

NOTE

The masking of a shared module must not be disabled in both SLEEPCON0 and SLEEPCON1.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 NIC1_MASK	Mask Bit of NIC_MEDIA1 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
22 NIC0_MASK	Mask Bit of NIC_MEDIA0 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
21 FLEXIO_MASK	Mask Bit of FLEXIO Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
20 LPSPI16_MASK	Mask Bit of LPSPI16 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
19	Mask Bit of LPSPI14

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPSPI14_MASK	Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
18 XSPI2_MASK	Mask Bit of XSPI2 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
17 EZHV_MASK	Mask Bit of EZH-V Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
16 VGPU_MASK	Mask Bit of VGPU Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
15-7 —	Reserved
6 RTC_MASK	Mask Bit of RTC0 and RTC1 Writing a 1 enables the masking of this SLEEPCON1 control for these shared modules. Reading returns the current mask value. 0b - Disables 1b - Enables
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK	Mask Bit of LPI2C15 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
2 MICFIL_MASK	Mask Bit of MICFIL Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
1 ACMP0_MASK	Mask Bit of ACMP0 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables
0 ADC0_MASK	Mask Bit of ADC0 Writing a 1 enables the masking of this SLEEPCON1 control for this shared module. Reading returns the current mask value. 0b - Disables 1b - Enables

30.4.2.21 Shared Resources Mask Set (SHARED_MASK0_SET)

Offset

Register	Offset
SHARED_MASK0_SET	1A4h

Function

Writing a 1 to a bit position sets the corresponding position in SHARED_MASK0. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved								NIC1_MA...	NIC0_MA...	FLEXIO_...	LPSPI16...	LPSPI14...	XSPI2_M...	EZHV_MA...	VGPU_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	Reserved									RTC_MAS...	0	0	LPI2C15...	MICFIL_...	ACMP0_M...	ADC0_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 NIC1_MASK_SET	NIC_MEDIA1 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
22 NIC0_MASK_SET	NIC_MEDIA0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
21 FLEXIO_MASK_SET	FLEXIO Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
20 LPSPI16_MASK_SET	LPSPI16 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
19 LPSPI14_MASK_SET	LPSPI14 Mask Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
18 XSPI2_MASK_SET	XSPI2 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
17 EZHV_MASK_SET	EZH-V Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
16 VGPU_MASK_SET	VGPU Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
15-7 —	Reserved
6 RTC_MASK_SET	RTC0 and RTC1 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK_SET	LPI2C15 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
2 MICFIL_MASK_SET	MICFIL Mask Set Changes the corresponding bit to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets the bit
1 ACMP0_MASK_SET	ACMP0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit
0 ADC0_MASK_SET	ADC0 Mask Set Changes the corresponding bit to 1. 0b - No effect 1b - Sets the bit

30.4.2.22 Shared Resources Mask Clear (SHARED_MASK0_CLR)

Offset

Register	Offset
SHARED_MASK0_CLR	1A8h

Function

Writing a 1 to a bit position clears the corresponding position in SHARED_MASK0. Writing a 0 to a bit position has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved								NIC1_MA...	NIC0_MA...	FLEXIO_...	LPSPI16...	LPSPI14...	XSPI2_M...	EZHV_MA...	VGPU_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	Reserved									RTC_MAS...	0	0	LPI2C15...	MICFIL_...	ACMP0_M...	ADC0_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 NIC1_MASK_CLR	NIC_MEDIA1 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
22 NIC0_MASK_CLR	NIC_MEDIAO Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
21 FLEXIO_MASK_CLR	FLEXIO Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
20 LPSPI16_MASK_CLR	LPSPI16 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
19 LPSPI14_MASK_CLR	LPSPI14 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
18 XSPI2_MASK_CLR	XSPI2 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
17 EZHV_MASK_CLR	EZH-V Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
16	VGPU Mask Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
VGPU_MASK_CLR	Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
15-7 —	Reserved
6 RTC_MASK_CLR	RTC0 and RTC1 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
5 —	Reserved
4 —	Reserved
3 LPI2C15_MASK_CLR	LPI2C15 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
2 MICFIL_MASK_CLR	MICFIL Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
1 ACMP0_MASK_CLR	ACMP0 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit
0 ADC0_MASK_CLR	ADC0 Mask Clear Changes the corresponding bit to 0. 0b - No effect 1b - Clears the bit

30.4.2.23 Media Domain Shared Controller Low-power Control (SHA_MED_CCTRL0)

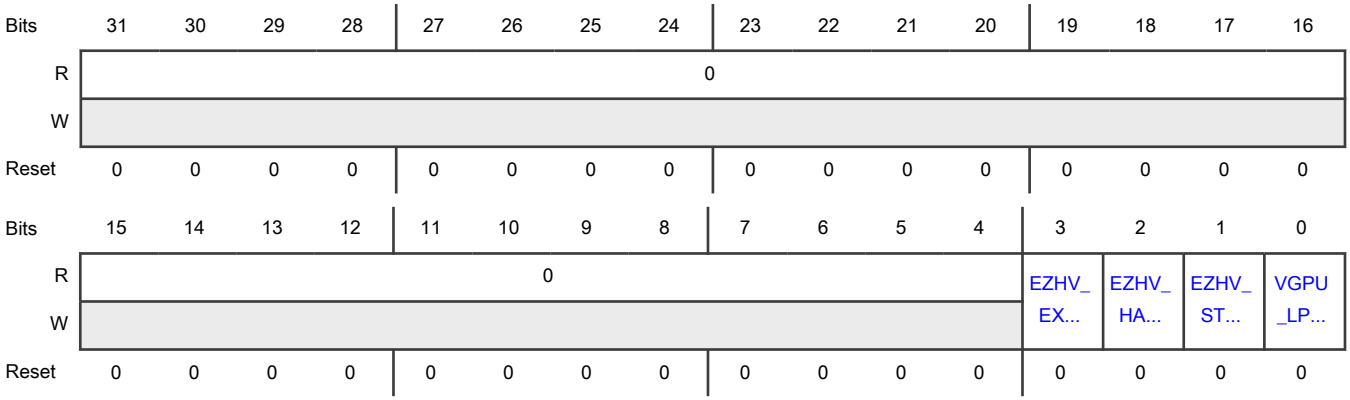
Offset

Register	Offset
SHA_MED_CCTRL0	1B0h

Function

Sends low-power request to shared bus controller if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 EZHV_EXIT_WAIT	EZH-V Exit Wait Mode Sends exit wait mode request to EZH-V. 0b - No request 1b - Sends
2 EZHV_HALT	EZH-V Halt Control Sends Halt request to EZH-V. 0b - No request 1b - Sends request
1 EZHV_STOP	EZH-V Stop Control Sends Stop request to EZH-V. 0b - No request 1b - Sends request

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 VGPU_LPREQ	VGPU Low-Power Control Sends low-power request to VGPU 0b - No request 1b - Sends request

30.4.2.24 Media Domain Shared Resources Target Low-power Control (SHA_MED_TCTRL0)

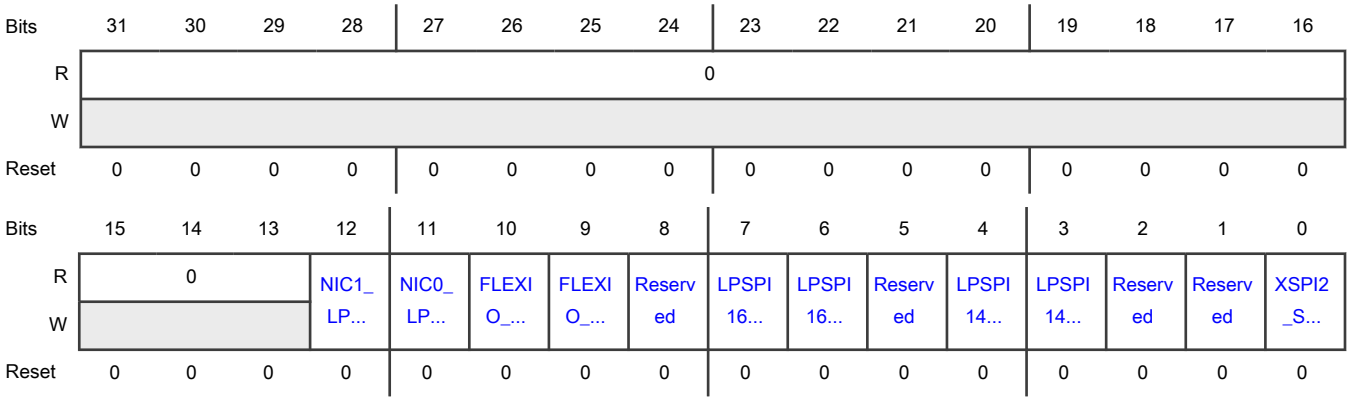
Offset

Register	Offset
SHA_MED_TCTRL0	1B4h

Function

Sends low-power request to shared bus target in the Media domain if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 NIC1_LPREQ	NIC_MEDIA1 Low-power Request Sends low-power request to NIC_MEDIA1. 0b - No request 1b - Sends request

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 NIC0_LPREQ	NIC_MEDIA0 Low-power Request Sends low-power request to NIC_MEDIA0. 0b - No request 1b - Sends request
10 FLEXIO_F_LPREQ	FLEXIO Function Low-power Request Sends function low-power request to FLEXIO. 0b - No request 1b - Sends request
9 FLEXIO_B_LPREQ	FLEXIO Bus Low-power Request Sends bus low-power request to FLEXIO. 0b - No request 1b - Sends request
8 —	Reserved
7 LPSPI16_F_LPREQ	LPSPI16 Function Low-power Request Sends function low-power request to LPSPI16. 0b - No request 1b - Sends request
6 LPSPI16_B_LPREQ	LPSPI16 Bus Low-power Request Sends bus low-power request to LPSPI16. 0b - No request 1b - Sends request
5 —	Reserved
4 LPSPI14_F_LPREQ	LPSPI14 Function Low-power Request Sends function low-power request to LPSPI14. 0b - No request 1b - Sends request
3 LPSPI14_B_LPREQ	LPSPI14 Bus Low-power Request Sends bus low-power request to LPSPI14.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No request 1b - Sends request
2 —	Reserved
1 —	Reserved
0 XSPI2_STOP	XSPI2 Stop Control Sends Stop request to XSPI2. 0b - No request 1b - Sends request

30.4.2.25 Shared VDD1_SENSE Domain Targets Low-power Control (SHA_SEN_TCTRL0)

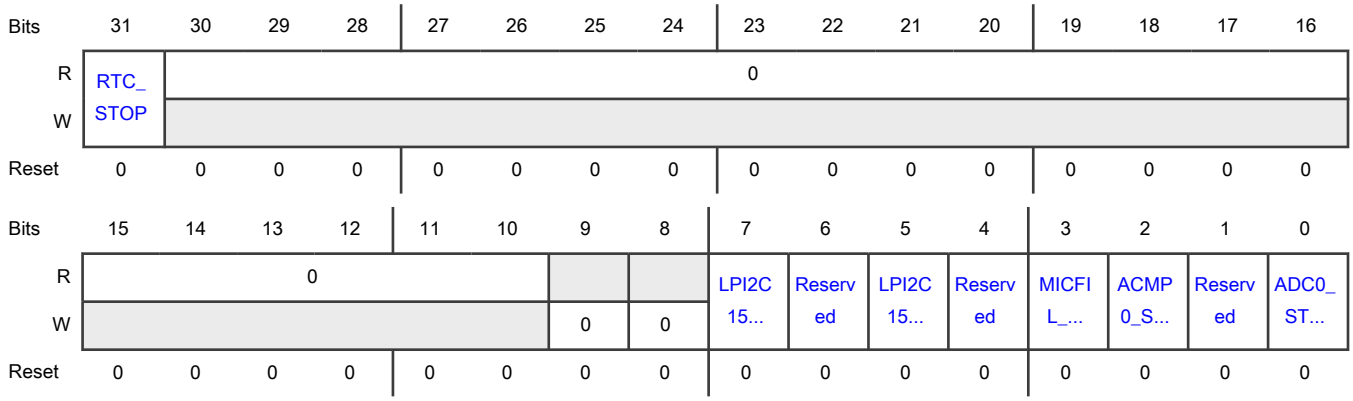
Offset

Register	Offset
SHA_SEN_TCTRL0	1BCh

Function

Sends low-power request to shared bus target in the VDD1_SENSE domain if its masking is disabled in SHARED_MASK0.

Diagram



Fields

Field	Function
31 RTC_STOP	RTC0 and RTC1 Stop Sends Stop request to RTC0 and RTC1. **NOTE** The RTC modules enter low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related SHA_MEDSEN_TSTAT0[RTC_STOP] bit will always read as 1. 0b - No request 1b - Sends request
30-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 LPI2C15_F_LPREQ	LPI2C15 Function Low-power Request Sends function low-power request to LPI2C15. 0b - No request 1b - Sends request
6 —	Reserved
5 LPI2C15_B_LPREQ	LPI2C15 Bus Low-power Request Sends bus low-power request to LPI2C15. 0b - No request 1b - Sends request
4 —	Reserved
3 MICFIL_STOP	MICFIL Stop Sends Stop request to MICFIL. 0b - No request 1b - Sends request
2	ACMP0 Stop

Table continues on the next page...

Table continued from the previous page...

Field	Function
ACMP0_STOP	Sends Stop request to ACMP0. **NOTE** The ACMP0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related SHA_MEDSEN_TSTAT0[ACMP0_STOP] bit will always read as 1. 0b - No request 1b - Sends request
1 —	Reserved
0 ADC0_STOP	ADC0 Stop Sends Stop request to ADC0. 0b - No request 1b - Sends request

30.4.2.26 Private Resources Controllers Low-power Control 0 (PRIVATE_CCTRL0)

Offset

Register	Offset
PRIVATE_CCTRL0	1C0h

Function

Sends low-power request to private bus controllers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EDMA 3_S...	EDMA 2_S...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1 EDMA3_STOP	eDMA3 Stop Sends Stop request to eDMA3 0b - No request 1b - Sends request
0 EDMA2_STOP	eDMA2 Stop Sends Stop request to eDMA2 0b - No request 1b - Sends request

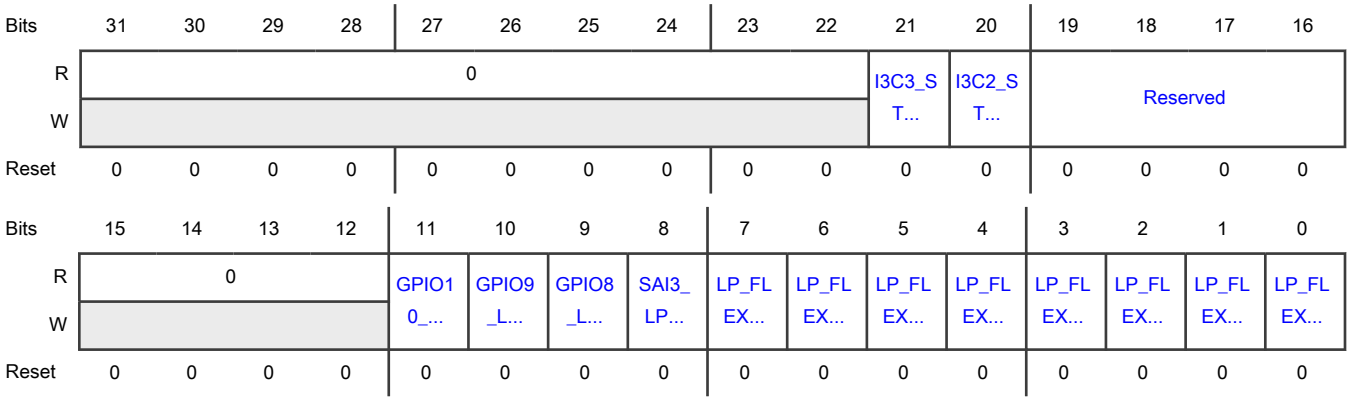
30.4.2.27 Private Resources Target Low-power Control (PRIVATE_TCTRL0)

Offset

Register	Offset
PRIVATE_TCTRL0	1C4h

Function
Sends low-power request to private bus targets.

Diagram



Fields

Field	Function
31-22 —	Reserved
21 I3C3_STOP	I3C3 Stop Sends Stop request to I3C3. **NOTE** The I3C3 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related PRIVATE_TSTAT0[I3C3_STOP] bit will always read as 1. 0b - No request 1b - Sends request
20 I3C2_STOP	I3C2 Stop Sends Stop request to I3C2. **NOTE** The I3C2 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The related PRIVATE_TSTAT0[I3C2_STOP] bit will always read as 1. 0b - No request 1b - Sends request
19-16 —	Reserved
15-12 —	Reserved
11 GPIO10_LPREQ	GPIO10 Low-power Request Sends low-power request to GPIO10. 0b - No request 1b - Sends request
10 GPIO9_LPREQ	GPIO9 Low-power Request Sends low-power request to GPIO9. 0b - No request 1b - Sends request
9 GPIO8_LPREQ	GPIO8 Low-power Request Sends low-power request to GPIO8.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No request 1b - Sends request
8 SAI3_LPREQ	SAI3 Low-power Request Sends low-power request to SAI3. 0b - No request 1b - Sends request
7 LP_FLEXCOMM20_F_LPREQ	LP_FLEXCOMM20 Function Low-power Request Sends function low-power request to LP_FLEXCOMM20. 0b - No request 1b - Sends request
6 LP_FLEXCOMM19_F_LPREQ	LP_FLEXCOMM19 Function Low-power Request Sends function low-power request to LP_FLEXCOMM19. 0b - No request 1b - Sends request
5 LP_FLEXCOMM18_F_LPREQ	LP_FLEXCOMM18 Function Low-power Request Sends function low-power request to LP_FLEXCOMM18. 0b - No request 1b - Sends request
4 LP_FLEXCOMM17_F_LPREQ	LP_FLEXCOMM17 Function Low-power Request Sends function low-power request to LP_FLEXCOMM17. 0b - No request 1b - Sends request
3 LP_FLEXCOMM20__B_LPREQ	LP_FLEXCOMM20 Bus Low-power Request Sends bus low-power request to LP_FLEXCOMM20. 0b - No request 1b - Sends request
2 LP_FLEXCOMM19__B_LPREQ	LP_FLEXCOMM19 Bus Low-power Request Sends bus low-power request to LP_FLEXCOMM19. 0b - No request 1b - Sends request
1	LP_FLEXCOMM18 Bus Low-power Request

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM18__B_LPREQ	Sends bus low-power request to LP_FLEXCOMM18. 0b - No request 1b - Sends request
0 LP_FLEXCOMM17__B_LPREQ	LP_FLEXCOMM17 Bus Low-power Request Sends bus low-power request to LP_FLEXCOMM17. 0b - No request 1b - Sends request

30.4.2.28 Media Domain Shared Controllers Low-power Status (SHA_MED_CSTAT0)

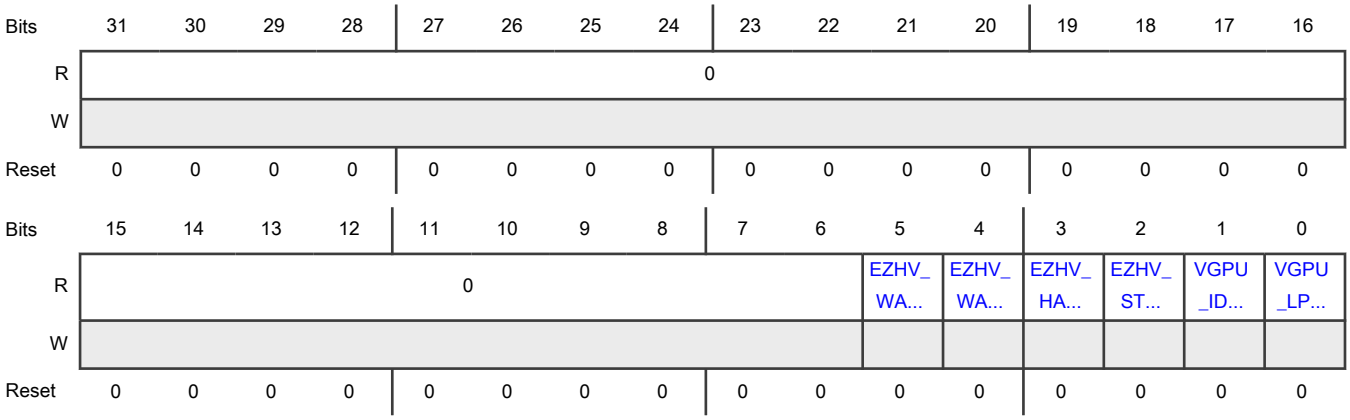
Offset

Register	Offset
SHA_MED_CSTAT0	1D0h

Function

Indicates the low-power ready status for bus controllers in Media domain.

Diagram



Fields

Field	Function
31-6 —	Reserved
5	Wakeup Request from EZH-V

Table continues on the next page...

Table continued from the previous page...

Field	Function
EZHV_WAKEUP	Indicates whether EZH-V needs clock running to wakeup 0b - Not needs 1b - Needs
4 EZHV_WAITING	EZH-V Waiting Status Indicates waiting status of EZH-V 0b - Not in waiting status 1b - In waiting status
3 EZHV_HALT	EZH-V Halt Status Indicates Halt status of EZH-V. 0b - Not in Halt status 1b - In Halt status
2 EZHV_STOP	EZH-V Stop Status Indicates the Stop status of EZH-V. 0b - Not in Stop status 1b - In Stop status
1 VGPU_IDLE	VGPU Idle Status Indicates VGPU Idle status. 0b - Not in Idle status, VGPU has operation to perform. 1b - In Idle status, VGPU might accept low-power request.
0 VGPU_LPACCEPT	VGPU Low-power Accept Status Indicates accept status to low-power request of VGPU. 0b - Not accept 1b - Accept

30.4.2.29 Media and Sense Domain Shared Targets Low-power Status (SHA_MEDSEN_TSTAT0)

Offset

Register	Offset
SHA_MEDSEN_TSTAT0	1D4h

Function

Indicates the low-power status for bus targets in Media and Sense domains.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RTC_STOP	0								0	0	LPI2C15...	LPI2C15...	MICFIL_...	ACMP0_S...	ADC0_ST...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							NIC1_LP...	NIC0_LP...	FLEXIO_...	FLEXIO_...	LPSPI16...	LPSPI16...	LPSPI14...	LPSPI14...	XSPI2_S...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 RTC_STOP	RTC0 and RTC1 Stop Status Indicates the Stop status of RTC0 and RTC1. **NOTE** The RTC modules enter low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. The bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
30-23 —	Reserved
22 —	Reserved
21 —	Reserved
20 LPI2C15_F_LP ACCEPT	LPI2C15 Function Low-power Accept Status Indicates the accept status to the function low-power request of LPI2C15. 0b - Not accept 1b - Accept
19 LPI2C15_B_LP ACCEPT	LPI2C15 Bus Low-power Accept Status Indicates the accept status to the bus low-power request of LPI2C15. 0b - Not accept 1b - Accept

Table continues on the next page...

Table continued from the previous page...

Field	Function
18 MICFIL_STOP	MICFIL Stop Status Indicates the Stop status of MICFIL. 0b - Not in Stop status 1b - In Stop status
17 ACMP0_STOP	ACMP0 Stop Status Indicates the Stop status of ACMP0. **NOTE** The ACMP0 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
16 ADC0_STOP	ADC0 Stop Status Indicates the Stop status of ADC0. 0b - Not in Stop status 1b - In Stop status
15-9 —	Reserved
8 NIC1_LPACCEPT	NIC_MEDIA1 Low-power Accept Status Indicates the accept status to low-power request of NIC_MEDIA1. 0b - Not accept 1b - Accept
7 NIC0_LPACCEPT	NIC_MEDIA0 Low-power Accept Status Indicates the accept status to low-power request of NIC_MEDIA0. 0b - Not accept 1b - Accept
6 FLEXIO_F_LPACCEPT	FLEXIO Function Low-power Accept Status Indicates the accept status to function low-power request of FLEXIO 0b - Not accept 1b - Accept
5 FLEXIO_B_LPACCEPT	FLEXIO Bus Low-power Accept Status Indicates the accept status to bus low-power request of FLEXIO

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not accept 1b - Accept
4 LPSPI16_F_LP ACCEPT	LPSPI16 Function Low-power Accept Status Indicates the accept status to function low-power request of LPSPI16. 0b - Not accept 1b - Accept
3 LPSPI16_B_LP ACCEPT	LPSPI16 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LPSPI16. 0b - Not accept 1b - Accept
2 LPSPI14_F_LP ACCEPT	LPSPI14 Function Low-power Accept Status Indicates the accept status to function low-power request of LPSPI14 0b - Not ready 1b - Ready
1 LPSPI14_B_LP ACCEPT	LPSPI14 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LPSPI14. 0b - Not accept 1b - Accept
0 XSPI2_STOP	XSPI2 Stop Status Indicates Stop status of XSPI2 0b - Not in Stop status 1b - In Stop status

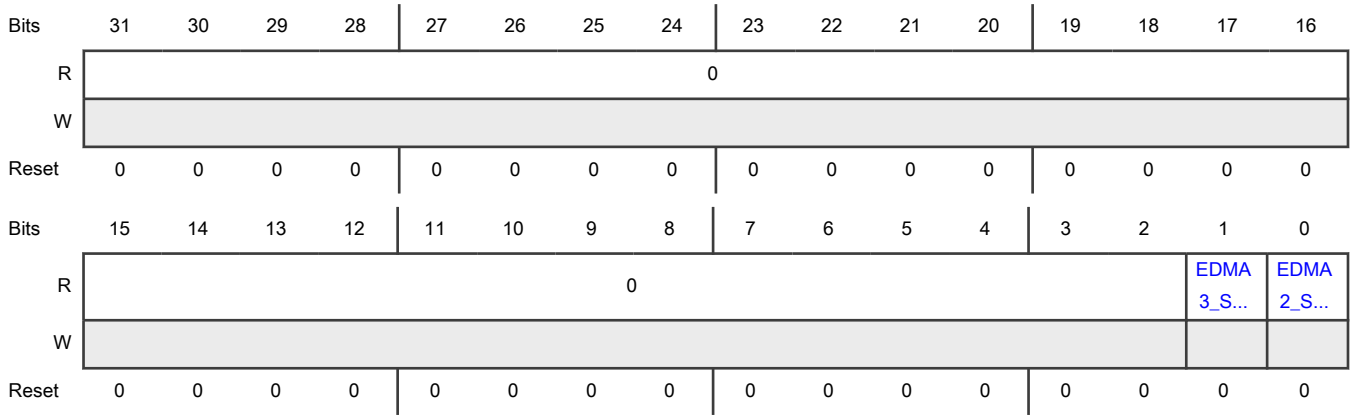
30.4.2.30 Private Resources Controllers Low-power Status (PRIVATE_CSTAT0)

Offset

Register	Offset
PRIVATE_CSTAT0	1E0h

Function
Indicates the low-power status of bus controllers that can only be controlled by SLEEPCON1.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 EDMA3_STOP	eDMA3 Stop Status Indicates Stop status of eDMA3. 0b - Not in Stop status 1b - In Stop status
0 EDMA2_STOP	eDMA2 Stop Status Indicates Stop status of eDMA2. 0b - Not in Stop status 1b - In Stop status

30.4.2.31 Private Resources Targets Low-power Status 0 (PRIVATE_TSTAT0)

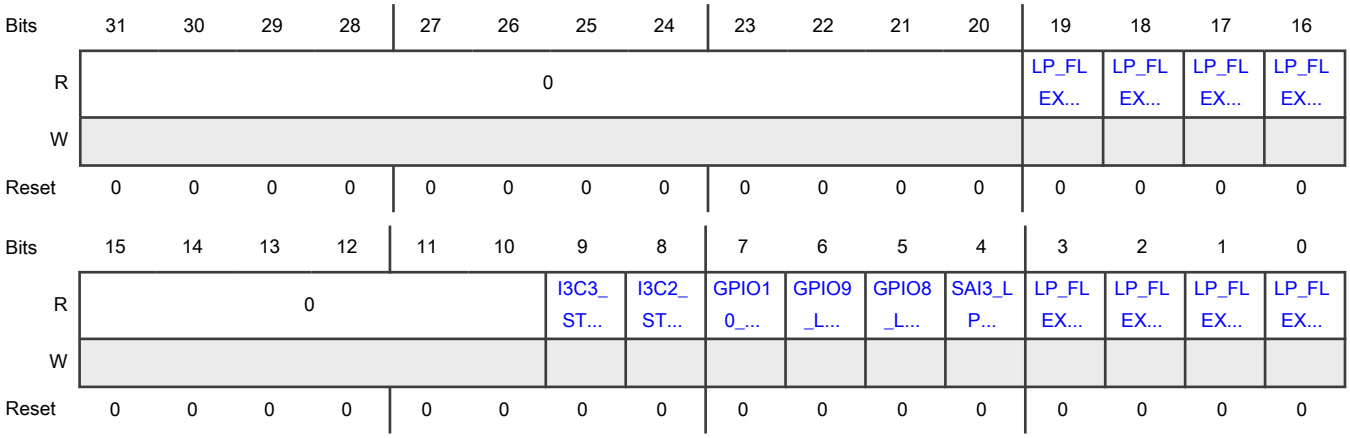
Offset

Register	Offset
PRIVATE_TSTAT0	1E4h

Function

Indicates low-power ready status for bus targets that can only be controlled by SLEEPCON1.

Diagram



Fields

Field	Function
31-20 —	Reserved
19 LP_FLEXCOMM20_F_LPACCEPT	LP_FLEXCOMM20 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM20. 0b - Not accept 1b - Accept
18 LP_FLEXCOMM19_F_LPACCEPT	LP_FLEXCOMM19 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM19. 0b - Not accept 1b - Accept
17 LP_FLEXCOMM18_F_LPACCEPT	LP_FLEXCOMM18 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM18. 0b - Not accept 1b - Accept
16 LP_FLEXCOMM17_F_LPACCEPT	LP_FLEXCOMM17 Function Low-power Accept Status Indicates the accept status to function low-power request of LP_FLEXCOMM17. 0b - Not accept 1b - Accept
15-10 —	Reserved
9	I3C3 Stop Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
I3C3_STOP	Indicates Stop status of I3C3. **NOTE** The I3C3 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
8 I3C2_STOP	I3C2 Stop Status Indicates Stop status of I3C2. **NOTE** The I3C2 module enters low-power mode on receiving the Stop request and will not send the acknowledge to SLEEPCON. This bit will always read as 1. 0b - Not in Stop status 1b - In Stop status
7 GPIO10_LPACCEPT	GPIO10 Low-power Accept Status Indicates the accept status to low-power request of GPIO10. 0b - Not accept 1b - Accept
6 GPIO9_LPACCEPT	GPIO Low-power Accept Status Indicates the accept status to low-power request of GPIO9. 0b - Not accept 1b - Accept
5 GPIO8_LPACCEPT	GPIO8 Low-power Accept Status Indicates the accept status to low-power request of GPIO8. 0b - Not accept 1b - Accept
4 SAI3_LPACCEPT	SAI3 Low-power Accept Status Indicates the accept status to low-power request of SAI3. 0b - Not accept 1b - Accept
3	LP_FLEXCOMM20 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM20.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM20_B_LPACCEPT	0b - Not accept 1b - Accept
2 LP_FLEXCOMM19_B_LPACCEPT	LP_FLEXCOMM19 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM19. 0b - Not accept 1b - Accept
1 LP_FLEXCOMM18_B_LPACCEPT	LP_FLEXCOMM18 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM18. 0b - Not accept 1b - Accept
0 LP_FLEXCOMM17_B_LPACCEPT	LP_FLEXCOMM17 Bus Low-power Accept Status Indicates the accept status to bus low-power request of LP_FLEXCOMM17. 0b - Not accept 1b - Accept

30.4.2.32 Low-Power Hint Status 0 (LP_HINT0)

Offset

Register	Offset
LP_HINT0	1ECh

Function

Indicates the active or inactive status of modules. If a module is inactive, you can use xxxxCTRLn register to send low-power request to it.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			NIC1_ID...	NIC1_ID...	NIC0_ID...	NIC0_ID...	FLEXIO_...	LPSPI16...	LPSPI14...	LPI2C15...	SAI3_F_...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				FLEXIO_...	LPSPI16...	LPSPI14...	LPI2C15...	GPIO10_...	GPIO9_L...	GPIO8_L...	SAI3_B_...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28 NIC1_IDLE1	NIC_MEDIA1 Idle Status 1 Indicates status of NIC_MEDIA1. 0b - Active 1b - Idle
27 NIC1_IDLE0	NIC_MEDIA1 Idle Status 0 Indicates status of NIC_MEDIA1. 0b - Active 1b - Idle
26 NIC0_IDLE1	NIC_MEDIA0 Idle Status 1 Indicates status of NIC_MEDIA0. 0b - Active 1b - Idle
25 NIC0_IDLE0	NIC_MEDIA0 Idle Status 0 Indicates status of NIC_MEDIA0. 0b - Active 1b - Idle
24 FLEXIO_F_LPHINT	FLEXIO Function Hint Status Indicates status of logic in FLEXIO that driven by function clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inactive 1b - Active
23 LPSPI16_F_LPHINT	LPSPI16 Function Hint Status Indicates status of logic in LPSPI16 that driven by function clock. 0b - Inactive 1b - Active
22 LPSPI14_F_LPHINT	LPSPI14 Function Hint Status Indicates status of logic in LPSPI14 that driven by function clock. 0b - Inactive 1b - Active
21 LPI2C15_F_LPHINT	LPI2C15 Function Hint Status Indicates status of logic in LPI2C15 that driven by function clock. 0b - Inactive 1b - Active
20 SAI3_F_LPHINT	SAI3 Function Hint Status Indicates status of logic in SAI3 that driven by function clock. 0b - Inactive 1b - Active
19 LP_FLEXCOMM20_F_LPHINT	LP_FLEXCOMM20 Function Hint Status Indicates status of logic in LP_FLEXCOMM20 that driven by function clock. 0b - Inactive 1b - Active
18 LP_FLEXCOMM19_F_LPHINT	LP_FLEXCOMM19 Function Hint Status Indicates status of logic in LP_FLEXCOMM19 that driven by function clock. 0b - Inactive 1b - Active
17 LP_FLEXCOMM18_F_LPHINT	LP_FLEXCOMM18 Function Hint Status Indicates status of logic in LP_FLEXCOMM18 that driven by function clock. 0b - Inactive 1b - Active
16	LP_FLEXCOMM17 Function Hint Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM17_F_LPHINT	Indicates status of logic in LP_FLEXCOMM17 that driven by function clock. 0b - Inactive 1b - Active
15-12 —	Reserved
11 FLEXIO_B_LPHINT	FLEXIO Bus Hint Status Indicates status of logic in FLEXIO that driven by bus clock. 0b - Inactive 1b - Active
10 LPSPI16_B_LPHINT	LPSPI16 Bus Hint Status Indicates status of logic in LPSPI16 that driven by bus clock. 0b - Inactive 1b - Active
9 LPSPI14_B_LPHINT	LPSPI14 Bus Hint Status Indicates status of logic in LPSPI14 that driven by bus clock. 0b - Inactive 1b - Active
8 LPI2C15_B_LPHINT	LPI2C15 Bus Hint Status Indicates status of logic in LPI2C15 that driven by bus clock. 0b - Inactive 1b - Active
7 GPIO10_LPHINT	GPIO10 Hint Status Indicates status of GPIO10. 0b - Inactive 1b - Active
6 GPIO9_LPHINT	GPIO9 Hint Status Indicates status of GPIO9. 0b - Inactive 1b - Active
5 GPIO8_LPHINT	GPIO8 Hint Status Indicates status of GPIO8.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inactive 1b - Active
4 SAI3_B_LPHINT	SAI3 Bus Hint Status Indicates status of logic in SAI3 that driven by bus clock. 0b - Inactive 1b - Active
3 LP_FLEXCOMM20_B_LPHINT	LP_FLEXCOMM20 Bus Hint Status Indicates status of logic in LP_FLEXCOMM20 that driven by bus clock. 0b - Inactive 1b - Active
2 LP_FLEXCOMM19_B_LPHINT	LP_FLEXCOMM19 Bus Hint Status Indicates status of logic in LP_FLEXCOMM19 that driven by bus clock. 0b - Inactive 1b - Active
1 LP_FLEXCOMM18_B_LPHINT	LP_FLEXCOMM18 Bus Hint Status Indicates status of logic in LP_FLEXCOMM18 that driven by bus clock. 0b - Inactive 1b - Active
0 LP_FLEXCOMM17_B_LPHINT	LP_FLEXCOMM17 Bus Hint Status Indicates status of logic in LP_FLEXCOMM17 that driven by bus clock. 0b - Inactive 1b - Active

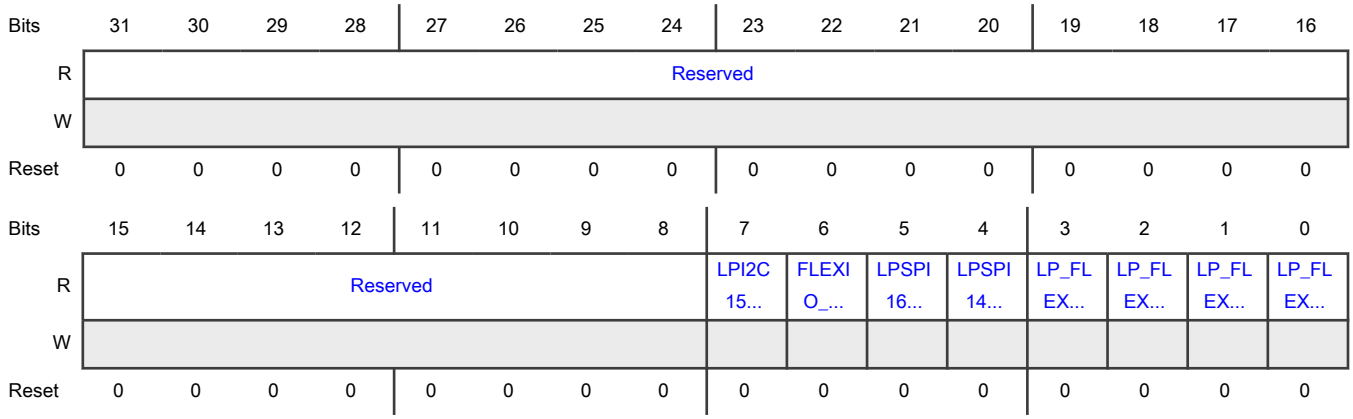
30.4.2.33 Low-Power Deny Status 0 (LP_DENY0)

Offset

Register	Offset
LP_DENY0	1F4h

Function
Indicates the module status to low-power request. If a module denied the low-power request, you need configure to send the low-power request again.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 LPI2C15_LPDENY	LPI2C15 LPDENY Status Indicates status of LPI2C15 to low-power request. 0b - Not deny 1b - Deny
6 FLEXIO_LPDENY	FLEXIO LPDENY Status Indicates status of FLEXIO to low-power request. 0b - Not deny 1b - Deny
5 LPSPI16_LPDENY	LPSPI16 LPDENY Status Indicates status of LPSPI16 to low-power request. 0b - Not deny 1b - Deny
4 LPSPI14_LPDENY	LPSPI14 LPDENY Status Indicates status of LPSPI14 to low-power request. 0b - Not deny 1b - Deny
3 LP_FLEXCOMM20_LPDENY	LP_FLEXCOMM20 LPDENY Status Indicates status of LP_FLEXCOMM20 to low-power request. 0b - Not deny

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Deny
2 LP_FLEXCOMM19_LPDENY	LP_FLEXCOMM19 LPDENY Status Indicates status of LP_FLEXCOMM19 to low-power request. 0b - Not deny 1b - Deny
1 LP_FLEXCOMM18_LPDENY	LP_FLEXCOMM18 LPDENY Status Indicates status of LP_FLEXCOMM18 to low-power request. 0b - Not deny 1b - Deny
0 LP_FLEXCOMM17_LPDENY	LP_FLEXCOMM17 LPDENY Status Indicates status of LP_FLEXCOMM17 to low-power request. 0b - Not deny 1b - Deny

Chapter 31
Reset Controller (RSTCTL)

31.1 Chip-specific RSTCTL information

Table 258. Reference links to related information

Topic	Related module	Reference
Full description	RSTCTL	RSTCTL
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller

NOTE

The RSTCTL module includes some security functionality that is not described here. Refer to the i.MX RT700 Security Reference Manual for more details.

31.1.1 Module instances

This chip has five instances of the RSTCTL module.

Table 259. RSTCTL instances

	RSTCTL0	RSTCTL1	RSTCTL2	RSTCTL3	RSTCTL4
Located in	VDD2_COM	VDD1_SENSE	VDDN_COM	VDD1_SENSE	VDD2_MEDIA
Controls modules in	VDD2_COMP, VDD2_COM, VDD2_DSP	VDD1_SENSE (private region)	VDDN_COM	VDD1_SENSE (shared region)	VDD2_MEDIA, VDDN_MEDIA
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

31.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

31.2 Overview

The RSTCTL module manages resets for the chip. It also provides registers for software to view the source of the latest reset event, and to reset individual peripheral modules. RSTCTL uses distributed structure to manage the reset function. The RSTCTL instances are illustrated and described in System diagram and Modules.

31.2.1 System diagram

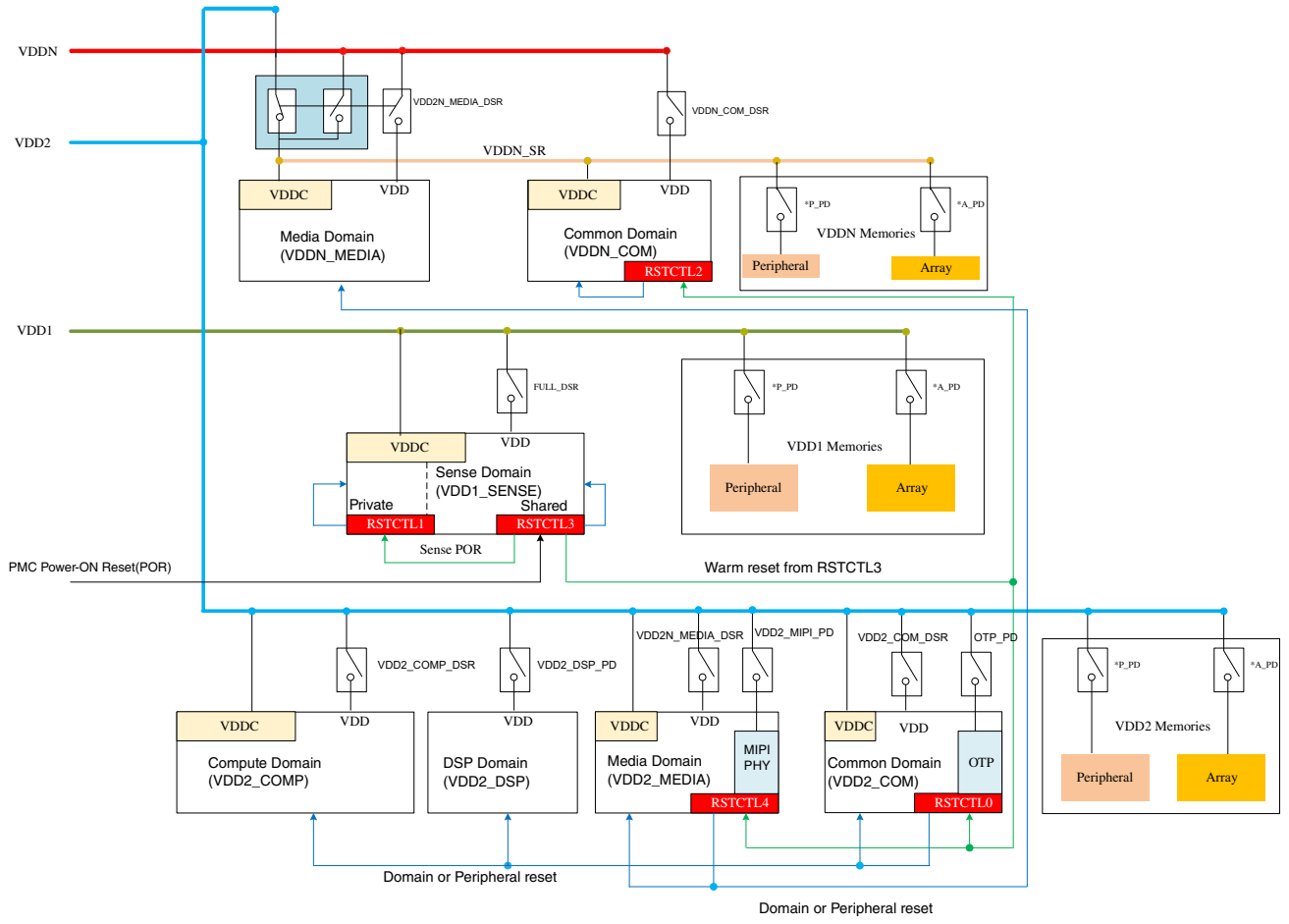


Figure 105. Reset controller system diagram

31.2.2 Features

The RSTCTL has the following key features:

- Cold Reset generation from PMC POR, triggered by POR, LVD, and HVD
- Warm Reset generation
  - RESETN
  - Software reset request (CPU0, CPU1)
  - Windowed Watchdog (WWDT0 - 3)
    - WWDT0 assigned to CPU0 (Compute Domain CM33 core), resets everything
    - WWDT1 assigned to HiFi4, resets everything
    - WWDT2 assigned to CPU1 (Sense Domain CM33 core), resets everything
    - WWDT3 assigned to HiFi1, resets DSP HiFi1 core
- Generate OCOTP reset and clock enable to trigger OCOTP FUSE reload after each Cold Reset.
- Generate domain power up reset, domain reset, peripheral reset, and domain clock enable to the Cold Reset, Warm Reset, and Deep Sleep Retention (DSR) mode.

- Reset individual peripheral modules based on software configuration.

31.2.3 Modules

The RSTCTL instances are described in Modules.

Table 260. Modules

Reset controller	Power Domain	Description
RSTCTL0	VDD2_COM	The RSTCTL0 is physically in the VDD2_COM power domain. It sends domain power-up reset, domain reset, and peripheral resets to VDD2_COMP, VDD2_DSP, and VDD2_COM power domains. RSTCTL0 and the domain peripherals can only be accessed by the Compute Domain CM33 (CPU0).
RSTCTL1	VDD1_SENSE (private)	The RSTCTL1 is physically in the VDD1_SENSE power domain. It sends peripheral resets to private peripherals in the VDD1_SENSE (private) power domain. RSTCTL1 and the domain peripherals can only be accessed by the Sense Domain CM33 (CPU1).
RSTCTL2	VDDN_COM	The RSTCTL2 is physically in the VDDN (Common) power domain. It sends domain power-up reset, domain reset, and peripheral resets to VDDN (Common) power domain. RSTCTL2 and the domain peripherals can be accessed by both CPUs.
RSTCTL3	VDD1_SENSE	The RSTCTL3 is physically in the VDD1_SENSE power domain. It sends power-up reset, domain reset, and peripheral resets to RSTCTL3 and shared peripherals in the VDD1_SENSE power domain. RSTCTL3 and the domain shared peripherals can be accessed by both CPUs.
RSTCTL4	VDD2_MEDIA	The RSTCTL4 is physically in the VDD2_MEDIA power domain. It sends domain power-up reset, domain reset, and peripheral resets to the VDD2_MEDIA and VDDN_MEDIA power domains. RSTCTL4 and the domain peripherals can be accessed by both CPUs. **NOTE** The VDD2_MEDIA and VDDN_MEDIA power domains will always power ON and OFF at the same time, although they are on different power rails (VDD2, VDDN).

31.3 Functional description

This section describes the key functionalities of RSTCTL.

31.3.1 Operation

After initial power up, the sequential order of reset is noted below:

1. Cold Reset is released.
2. Domain power-up reset.
3. Warm Reset is released.
4. OCOTP reload.
5. RESETN is released.
6. Domain reset, bus clock, and default peripheral clock are released.

NOTE

After initial power-up, VDD2_MEDIA and VDDN_MEDIA power domains are powered off by default, and VDDN_COM, VDD2_COM, VDD2_COMP, and VDD1_SENSE power domains are powered on by default.

You can configure software reset for modules in the RSTCTL registers. Write 1 to assert reset or write 0 to de-assert reset in the RSTCTL registers. You can also set and clear RSTCTL registers. This means each reset bit in RSTCL registers can also be controlled by the set and clear registers, which are at 2 different addresses. After initial power-up reset, most modules will be in reset state until software clears the corresponding bits in RSTCTL registers.

NOTE

Do not access the module instance whose related PRSTCTL register field is configured as 1b (sets reset) because the module instance under reset may not response.

In Deep Sleep Retention (DSR) mode, if a domain's power is off, its power-up reset, domain reset, and peripheral reset will be asserted for non-retention logic, and will not be asserted for retention logic. After wakeup from DSR mode, the peripheral's reset value will come from previous values in RSTCTL registers, before entering into DSR mode. This value may not be the hardware value if software changed it before entering into DSR mode.

31.3.2 Reset Controller 0 (RSTCTL0) in VDD2_COM Domain

RSTCTL0 has functions as below:

- Get Cold Reset and power switch to generate the domain power-up reset for VDD2_COMP, VDD2_DSP, and VDD2_COM power domains.
- Get Warm Reset and this domain's power-up reset to generate the domain reset for VDD2_COMP, VDD2_DSP, and VDD2_COM power domains.
- Generate the domain power-up reset and the domain reset for non-retention logic in VDD2_COMP, VDD2_DSP, and VDD2_COM power domains. Meanwhile keep domain power up and domain reset deasserted for retention logic in DSR mode.
- Configure individual peripheral resets by software.
- Only Compute Domain CM33 (CPU0) can access RSTCTL0 registers.

RSTCTL0 is in the VDD2_COM Domain, powered by the VDD2 power rail. It should be the last to power off amongst the other domains connected to VDD2, so VDD2 can supply power-up reset and domain reset to the domains when it is powered on.

31.3.3 Reset controller 1 (RSTCTL1) in VDD1_SENSE Domain (private)

RSTCTL1 has functions as below:

- Use RSTCTL3's domain power-up reset and domain reset to generate reset for Sense Domain CM33 (CPU1) private peripherals.
- Configure individual peripheral resets by software.

- Only Sense Domain CM33 (CPU1) can access RSTCTL1 registers.

RSTCTL1 is the reset controller for the CPU1 private peripherals. Its domain power up reset and domain reset come from RSTCTL3.

31.3.4 Reset controller 2 (RSTCTL2) in VDDN (Common) Domain

RSTCTL2 has functions as below:

- Enable Cold Reset and power switch to generate the domain power-up reset for VDDN_COM power domain's peripherals.
- Enable Warm Reset and this domain power-up reset to generate the domain reset for VDDN_COM power domain's peripherals.
- Configure individual peripheral resets by software.
- CPU0 and CPU1 can access RSTCTL2 registers.

RSTCTL2 is the reset controller for VDDN_COM domain's peripherals.

31.3.5 Reset controller 3 (RSTCTL3) in VDD1_SENSE Domain

RSTCTL3 has functions as below:

- Use PMC power-up reset to generate the Cold Reset for RSTCTL0, RSTCTL2 and RSTCTL4.
- Collect Warm Reset sources to generate the Warm Reset for RSTCTL0, RSTCTL2 and RSTCTL4.
- Generate the domain power-up reset for RSTCTL1 and RSTCTL3.
- Generate the domain reset for RSTCTL1 and RSTCTL3.
- Generate the domain power-up reset and the domain reset for VDD1_SENSE power domain while in DSR mode.
- Configure individual peripheral resets by software.
- CPU0 and CPU1 can access RSTCTL3 registers.

If PMC's power-up reset releases, RSTCTL3 starts to work. It is also the main reset controller for RSTCTL1. RSTCTL1 uses RSTCTL3's domain power-up reset and domain reset as its reset input.

Because VDD1_SENSE power domain should be the last one to power-off compared with other domains on the VDD1, VDD2 and VDDN power rails, VDD1_SENSE power domain can continue work as long as it is powered on.

The Warm Reset is released after all of Warm Reset sources are de-asserted.

31.3.6 Reset controller 4 (RSTCTL4) in VDD2_MEDIA Domain

RSTCTL4 has functions as below:

- Enable Cold Reset and power switch to generate the domain power-up reset for the RSTCTL4's peripherals.
- Enable Warm Reset and this domain power-up reset to generate the domain reset for the RSTCTL4's peripherals.
- Configure individual peripheral resets by software.
- CPU0 and CPU1 can access RSTCTL4 registers.

RSTCTL4 is a target reset controller for the VDD2_MEDIA and VDDN_MEDIA power domains.

31.3.7 Clocking

Each RSTCTL has its own always running clock shown in Table 261.

Table 261. Clocking

Clock	Description
1m_lposc	The clock source for RSTCTL is generated by CLKCTL. All synchronizers use this clock as sync clock.

31.3.8 Reset

31.3.8.1 Cold Reset

The Cold Reset will be triggered by any of the Cold Reset source assertion.

Table 262. Main cold reset source

Cold reset source	Description
PMC power-on reset (POR)	PMC power-on reset will only be released when all power rails (VDD1, VDD2 and VDDN) are ready.
PMC Low Voltage Detect (LVD) reset	PMC will assert this reset when any LVD happens in any power rails (VDD1, VDD2 or VDDN).
PMC High Voltage Detect (HVD) reset	PMC will assert this reset when any HVD happens in any power rails (VDD1, VDD2 or VDDN).

The Cold Reset will trigger the following:

- Resets all logic in VDD1_SENSE, VDD2_COM, VDD2_MEDIA, and VDDN_COM domains.
- Triggers Warm Reset.
- Resets debug and trace relative components.
- OCOTP FUSE reloads.
- Wakes up from low power mode without recovery to old state before entering into low power mode. After wakeup, the chip follows power up sequence.

31.3.8.2 Warm Reset

The Warm Reset will be triggered by the following:

- Cold Reset
- Any Warm Reset source assertion

Warm Reset will trigger the following:

- Resets all modules in the VDD1_SENSE, VDD2_COM, VDD2_MEDIA, and VDDN_COM domains except for debug and trace modules.
- Resets all individual peripheral reset bits in RSTCTL0 - 4.
- After reset releases, individual peripheral reset status will come from the default values of RSTCTL0 - 4. Software configurable reset bits default value is asserted, and software un-configurable reset bit value is deasserted by default.
- If device was in a low power mode and reset source was not a cold reset, then wakes up PMC from low power mode. Power switch and LDO status will be recovered to the previous state before entering into the new low-power mode.

NOTE

This may be different with power up state depending on how software configures PDRUNCFG.

The Warm Reset sources are listed below.

- RESETN
- Software reset request from CPU0
- Software reset request from CPU1
- WWDT0 (assigned to CPU0)
- WWDT1 (assigned to DSP HiFi4 Core)
- WWDT2 (assigned to CPU1)
- WWDT3 (assigned to DSP HiFi1 Core)

31.3.8.3 Peripheral reset

Individual peripheral reset will be triggered by the following:

- Writes 1 to the bit fields in reset controller registers.

NOTE

Software needs to write 0 to release the module reset.

- The peripheral modules' power domain is OFF. After peripheral modules' power domain is powered up, its reset state comes from the current value of its reset controller.

31.3.8.4 Special reset case(s)

WWDT3 resets HiFi1. In case of system reset or WWDT3 timeout, HiFi1 is held in reset until released by a software command issued by the CM33 Processor. Nothing else in the system should be directly impacted by WWDT3. Once WWDT3 is enabled, you can enable or disable WWDT3 reset and generate an interrupt to the Sense CM33 Processor, depending on the configuration of WWDT3. See Windowed Watchdog Timer (WWDT) section for configuration details of WWDT3.

31.3.9 Interrupts (RESETN)

RESET PAD (RESETN) is a warm reset source, and is in the AON 1.8V power domain. RESETN can wakeup the chip from any low-power mode, even if VDD1, VDD2, VDDN and PMC are OFF. After wakeup, RESETN will trigger a Warm Reset.

31.3.10 Low-power modes

If Cold Reset happens when the chip is in low-power mode, it wakes up from low-power mode and triggers a Cold Reset.

When VDD1 is ON, Warm Reset can wakeup the chip from low-power mode. PMC will wake up the DSR domain and trigger a Warm Reset. Only power switch restores to the old value before entering low power mode. Other configuration will go to reset default state.

NOTE

This may be different with power up state depending on how SW configure PDRUNCFG.

31.3.11 Debug mode

- Cold Reset and Domain POR can reset debug and trace components.
- Warm Reset, Domain Reset, and software cannot reset debug and trace components.

31.4 External signals

The module signals are shown as below:

Signal	I/O	Description
RESET_PAD (RESETN) **NOTE** RESETN only connects to RSTCTL3.	I/O	Bi-directional PAD. A LOW on this signal will trigger a main warm reset, cause I/O ports and peripherals to take on their default states, and execute boot code without FUSE reload. If the chip is in any low power mode, a LOW on this PAD wakes up the chip from the low power mode, and triggers a main warm reset at the same time. When RESETN is driven low externally, the device will stay in reset the entire time. When the external RESETN source releases, the chip will driver RESETN low through the main warm reset process to indicate the chip is still in the reset state internally. RESETN will be released (HiZ) when the warm reset process completes internally.

31.5 RSTCTL register descriptions

This section includes the RSTCTL module memory map and detailed descriptions of all registers.

31.5.1 RSTCTL0 register descriptions

This section includes the memory map and detailed descriptions of registers for RSTCTL0. This reset controller controls peripherals in the VDD2_COMP, VDD2_DSP, and VDD2_COM power domains on the VDD2 power rail.

31.5.1.1 RSTCTL0 memory map

RSTCTL0 base address: 4000_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Domain Reset Status (DOMRSTSTAT)	32	R	0000_000Bh
10h	Common Domain Peripheral Reset Control 0 (PRSTCTL0)	32	RW	0000_0040h
18h	Compute Domain Peripheral Reset Control 2 (PRSTCTL2)	32	RW	C7FC_0D60h
1Ch	Compute Domain Peripheral Reset Control 3 (PRSTCTL3)	32	RW	CFE7_EFFFh
20h	Compute Domain Peripheral Reset Control 4 (PRSTCTL4)	32	RW	0000_1DA7h
24h	DSP Domain Peripheral Reset Control 5 (PRSTCTL5)	32	RW	0000_0005h
40h	Common Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)	32	W	0000_0000h
48h	Compute Domain Peripheral Reset Control 2 SET (PRSTCTL2_SET)	32	W	0000_0000h
4Ch	Compute Domain Peripheral Reset Control 3 SET (PRSTCTL3_SET)	32	W	0000_0000h
50h	Compute Domain Peripheral Reset Control 4 SET (PRSTCTL4_SET)	32	W	0000_0000h
54h	DSP Domain Peripheral Reset Control 5 SET (PRSTCTL5_SET)	32	W	0000_0000h
70h	Common Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)	32	W	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
78h	Compute Domain Peripheral Reset Control 2 CLR (PRSTCTL2_CLR)	32	W	0000_0000h
7Ch	Compute Domain Peripheral Reset Control 3 CLR (PRSTCTL3_CLR)	32	W	0000_0000h
80h	Compute Domain Peripheral Reset Control 4 CLR (PRSTCTL4_CLR)	32	W	0000_0000h
84h	DSP Domain Peripheral Reset Control 5 CLR (PRSTCTL5_CLR)	32	W	0000_0000h

31.5.1.2 Domain Reset Status (DOMRSTSTAT)

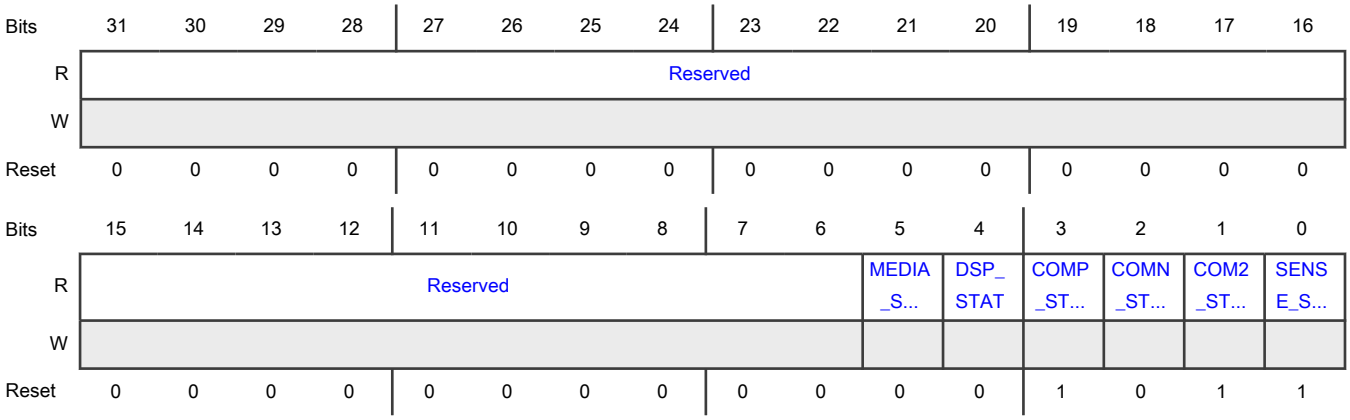
Offset

Register	Offset
DOMRSTSTAT	4h

Function

Shows the reset status of domains.

Diagram



Fields

Field	Function
31-6 —	Reserved
5	VDD2_MEDIA Domain Reset State

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEDIA_STAT	Shows the reset status of VDD2_MEDIA domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
4 DSP_STAT	VDD2_DSP Domain Reset State Shows the reset status of VDD2_DSP domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
3 COMP_STAT	VDD2_COMP Domain Reset State Shows the reset status of VDD2_COMP domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
2 COMN_STAT	VDDN_COM Domain Reset State Shows the reset status of VDDN_COM domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
1 COM2_STAT	VDD2_COM (Main) Domain Reset State Shows the reset status of VDD2_COM domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
0 SENSE_STAT	VDD1_SENSE Domain Reset State Shows the reset status of VDD1_SENSE domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.

31.5.1.3 Common Domain Peripheral Reset Control 0 (PRSTCTL0)

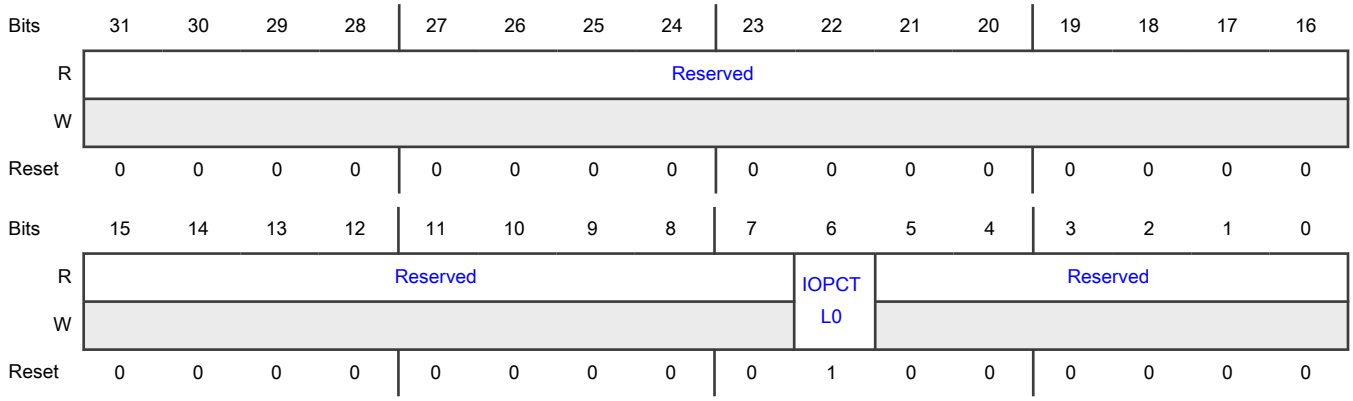
Offset

Register	Offset
PRSTCTL0	10h

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold and warm reset. Because RSTCTL is in the DSR list, so even wakeup from DSR mode this register's value will retain, except wakeup by RESETN.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 IOPCTL0	IOPCTL0 Reset Sets IOPCTL0 reset. 0b - Clears reset. 1b - Sets reset.
5-0 —	Reserved

31.5.1.4 Compute Domain Peripheral Reset Control 2 (PRSTCTL2)

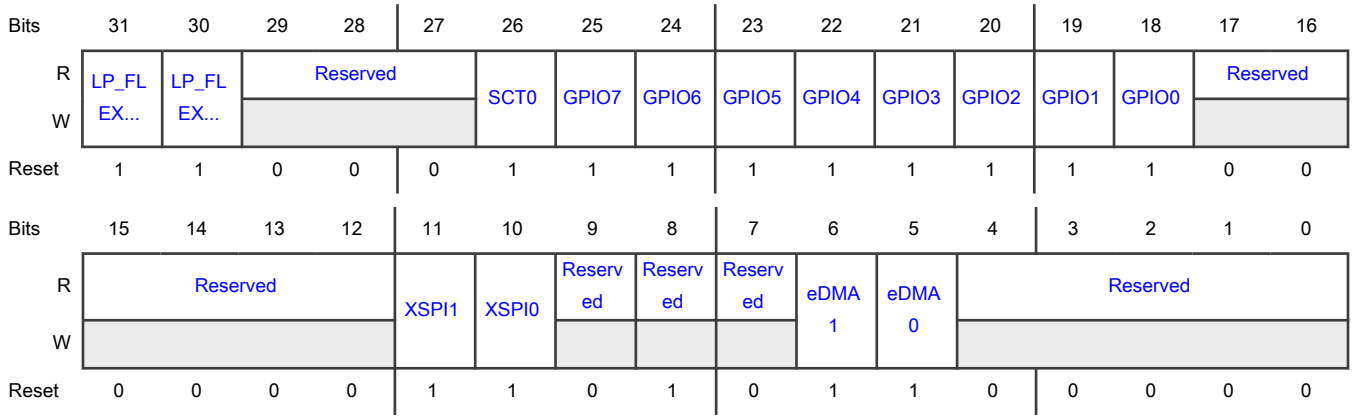
Offset

Register	Offset
PRSTCTL2	18h

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold and warm reset. Because RSTCTL is in the DSR list, so even wakeup from DSR mode this register's value will retain, except wakeup by RESETN.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Reset Enables LP_FLEXCOMM1 reset control. 0b - Clears reset. 1b - Sets reset.
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Reset Enables LP_FLEXCOMM0 reset control. 0b - Clears reset. 1b - Sets reset.
29-27 —	Reserved
26 SCT0	SCT0 Reset Enables SCT0 reset control. **NOTE** Writing 1 to this field will reset the SCT0 module and the SCT0_TRIG0 - SCT0_TRIG6 registers of INPUTMUX0 module. 0b - Clears reset. 1b - Sets reset.
25 GPIO7	GPIO7 Reset Enables GPIO7 reset control. 0b - Clears reset. 1b - Sets reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 GPIO6	GPIO6 Reset Enables GPIO6 reset control. 0b - Clears reset. 1b - Sets reset.
23 GPIO5	GPIO5 Reset Enables GPIO5 reset control. 0b - Clears reset. 1b - Sets reset.
22 GPIO4	GPIO4 Reset Enables GPIO4 reset control. 0b - Clears reset. 1b - Sets reset.
21 GPIO3	GPIO3 Reset Enables GPIO3 reset control. 0b - Clears reset. 1b - Sets reset.
20 GPIO2	GPIO2 Reset Enables GPIO2 reset control. 0b - Clears reset. 1b - Sets reset.
19 GPIO1	GPIO1 Reset Enables GPIO1 reset control. 0b - Clears reset. 1b - Sets reset.
18 GPIO0	GPIO0 Reset Enables GPIO0 reset control. 0b - Clears reset. 1b - Sets reset.
17-12 —	Reserved
11	XSPI1 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
XSPI1	Enables XSPI1, CACHE64_CTRL1 and MMU1 reset control. 0b - Clears reset. 1b - Sets reset.
10 XSPI0	XSPI0 Reset Enables XSPI0, CACHE64_CTRL0 and MMU0 reset control. 0b - Clears reset. 1b - Sets reset.
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 eDMA1	eDMA1 Reset Enables eDMA1 reset control. 0b - Clears reset. 1b - Sets reset.
5 eDMA0	eDMA0 Reset Enables eDMA0 reset control. 0b - Clears reset. 1b - Sets reset.
4-0 —	Reserved

31.5.1.5 Compute Domain Peripheral Reset Control 3 (PRSTCTL3)

Offset

Register	Offset
PRSTCTL3	1Ch

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold and warm reset. Because RSTCTL is in the DSR list, so even wakeup from DSR mode this register's value will retain, except wakeup by RESETN.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MU4	SEMA42_4	Reserved		UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0	Reserved		CRC0	I3C1	I3C0
W																
Reset	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAI2	SAI1	SAI0	Reserved	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
W																
Reset	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31 MU4	MU4 Reset Enables MU4 reset control. 0b - Clears reset. 1b - Sets reset.
30 SEMA42_4	SEMA42_4 Reset Enables SEMA42_4 reset control. 0b - Clears reset. 1b - Sets reset.
29-28 —	Reserved
27 UTICK0	UTICK0 Reset Enables UTICK0 reset control. 0b - Clears reset. 1b - Sets reset.
26 MRT0	MRT0 Reset Enables MRT0 reset control. 0b - Clears reset. 1b - Sets reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 CTIMER4	CTIMER4 Reset Enables CTIMER4 reset control. 0b - Clears reset. 1b - Sets reset.
24 CTIMER3	CTIMER3 Reset Enables CTIMER3 reset control. 0b - Clears reset. 1b - Sets reset.
23 CTIMER2	CTIMER2 Reset Enables CTIMER2 reset control. 0b - Clears reset. 1b - Sets reset.
22 CTIMER1	CTIMER1 Reset Enables CTIMER1 reset control. 0b - Clears reset. 1b - Sets reset.
21 CTIMER0	CTIMER0 Reset Enables CTIMER0 reset control. 0b - Clears reset. 1b - Sets reset.
20-19 —	Reserved
18 CRC0	CRC0 Reset Enables CRC0 reset control. 0b - Clears reset. 1b - Sets reset.
17 I3C1	I3C1 Reset Enables I3C1 reset control. 0b - Clears reset. 1b - Sets reset.
16	I3C0 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
I3C0	Enables I3C0 reset control. 0b - Clears reset. 1b - Sets reset.
15 SAI2	SAI2 Reset Enables SAI2 reset control. 0b - Clears reset. 1b - Sets reset.
14 SAI1	SAI1 Reset Enables SAI1 reset control. 0b - Clears reset. 1b - Sets reset.
13 SAI0	SAI0 Reset Enables SAI0 reset control. 0b - Clears reset. 1b - Sets reset.
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Reset Enables LP_FLEXCOMM13 reset control. 0b - Clears reset. 1b - Sets reset.
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Reset Enables LP_FLEXCOMM12 reset control. 0b - Clears reset. 1b - Sets reset.
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Reset Enables LP_FLEXCOMM11 reset control. 0b - Clears reset. 1b - Sets reset.
8	LP_FLEXCOMM10 Reset Enables LP_FLEXCOMM10 reset control.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM10	0b - Clears reset. 1b - Sets reset.
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Reset Enables LP_FLEXCOMM9 reset control. 0b - Clears reset. 1b - Sets reset.
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Reset Enables LP_FLEXCOMM8 reset control. 0b - Clears reset. 1b - Sets reset.
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Reset Enables LP_FLEXCOMM7 reset control. 0b - Clears reset. 1b - Sets reset.
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Reset Enables LP_FLEXCOMM6 reset control. 0b - Clears reset. 1b - Sets reset.
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Reset Enables LP_FLEXCOMM5 reset control. 0b - Clears reset. 1b - Sets reset.
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Reset Enables LP_FLEXCOMM4 reset control. 0b - Clears reset. 1b - Sets reset.
1 LP_FLEXCOMM3	LP_FLEXCOMM3 Reset Enables LP_FLEXCOMM3 reset control. 0b - Clears reset. 1b - Sets reset.
0	LP_FLEXCOMM2 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM2	Enables LP_FLEXCOMM2 reset control. 0b - Clears reset. 1b - Sets reset.

31.5.1.6 Compute Domain Peripheral Reset Control 4 (PRSTCTL4)

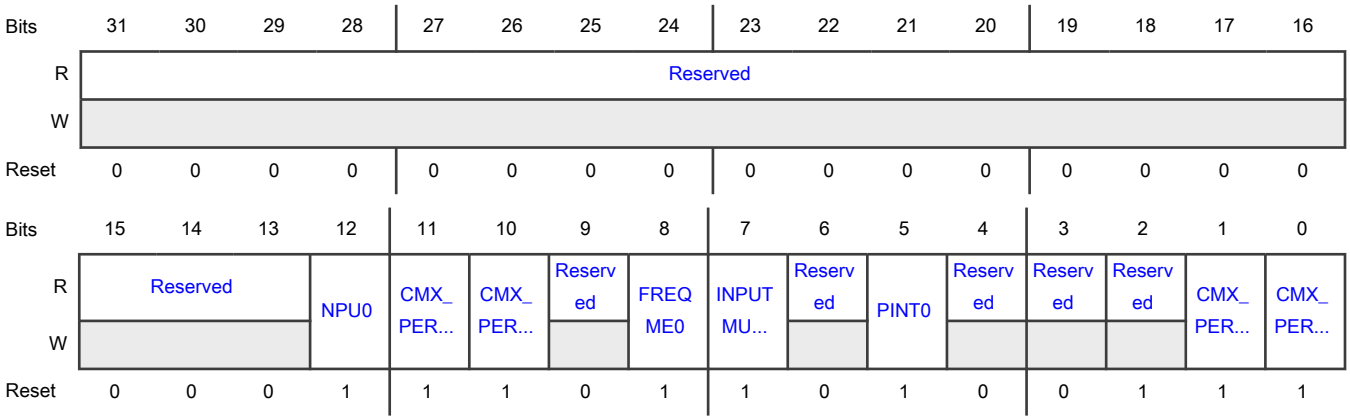
Offset

Register	Offset
PRSTCTL4	20h

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold and warm reset. Because RSTCTL is in the DSR list, so even wakeup from DSR mode this register's value will retain, except wakeup by RESETN.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 NPU0	NPU0 Reset Enables NPU0 reset control. 0b - Clears reset. 1b - Sets reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 CMX_PERFMON1	CMX_PERFMON1 Reset Enables CMX_PERFMON1 reset control. 0b - Clears reset. 1b - Sets reset.
10 CMX_PERFMON0	CMX_PERFMON0 Reset Enables CMX_PERFMON0 reset control. 0b - Clears reset. 1b - Sets reset.
9 —	Reserved
8 FREQME0	FREQME Reset Enables FREQME reset control. 0b - Clears reset. 1b - Sets reset.
7 INPUTMUX0	INPUTMUX0 Reset Enables INPUTMUX0 reset control. **NOTE** Writing 1 to this field will also reset the following registers of INPUTMUX0 module: • HIFI4_INTERRUPT0 - HIFI4_INTERRUPT26 • LP_FLEXCOMM0_TRIG - LP_FLEXCOMM13_TRIG • CTIMER0_CAP0 - CTIMER4_CAP3 • CTIMER0_TRIG - CTIMER4_TRIG • FREQME_TRIG0 - FREQME_TRIG1 • EZHV_TRIG0 - EZHV_TRIG7 • FLEXIO_TRIG0 - FLEXIO_TRIG3 0b - Clears reset. 1b - Sets reset.
6 —	Reserved
5 PINT0	PINT0 Reset Enables PINT0 reset control.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Writing 1 to this field will reset the PINT0 module and the PINT0_TRIG0 - PINT0_TRIG7 registers of INPUTMUX0 module. 0b - Clears reset. 1b - Sets reset.
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 CMX_PERFMON3	CMX_PERFMON3 Reset Enables CMX_PERFMON3 reset control. 0b - Clears reset. 1b - Sets reset.
0 CMX_PERFMON2	CMX_PERFMON2 Reset Enables CMX_PERFMON2 reset control. 0b - Clears reset. 1b - Sets reset.

31.5.1.7 DSP Domain Peripheral Reset Control 5 (PRSTCTL5)

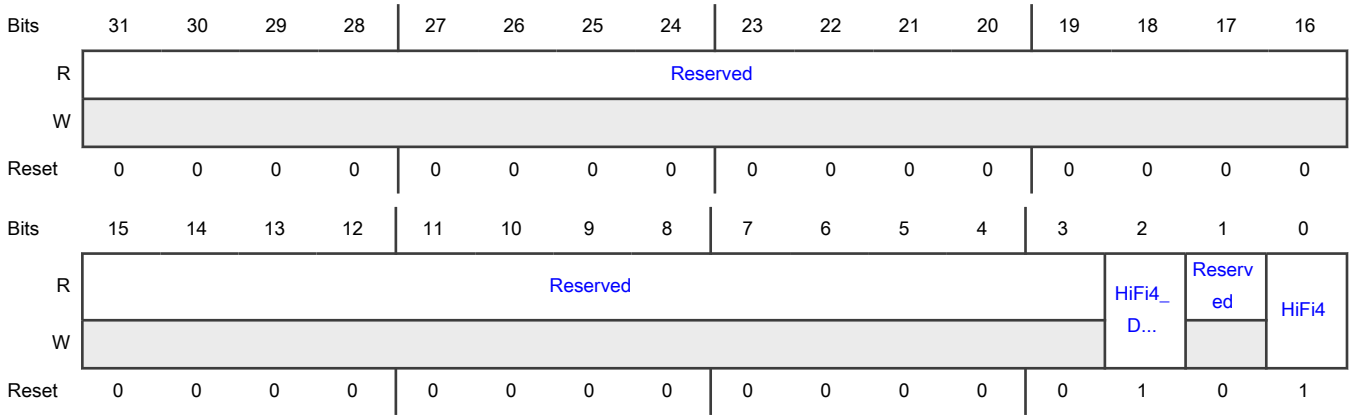
Offset

Register	Offset
PRSTCTL5	24h

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold and warm reset. Because RSTCTL is in the DSR list, so even wakeup from DSR mode this register's value will retain, except wakeup by RESETN.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 HiFi4_DEBUG	HiFi4 Debug Reset Enables HiFi4 debug reset control. 0b - Clears reset. 1b - Sets reset.
1 —	Reserved
0 HiFi4	HiFi4 Reset Enables HiFi4 reset control. 0b - Clears reset. 1b - Sets reset.

31.5.1.8 Common Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)

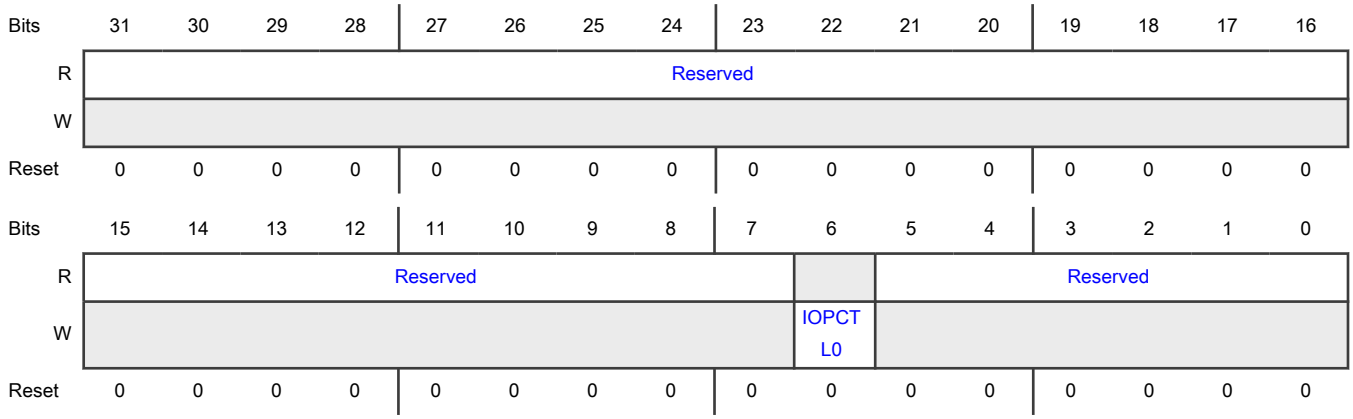
Offset

Register	Offset
PRSTCTL0_SET	40h

Function

Writing one to a bit position in PRSTCTL0_SET sets the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 IOPCTL0	IOPCTL0 Reset Sets IOPCTL0 reset. 0b - No effect 1b - Sets reset.
5-0 —	Reserved

31.5.1.9 Compute Domain Peripheral Reset Control 2 SET (PRSTCTL2_SET)

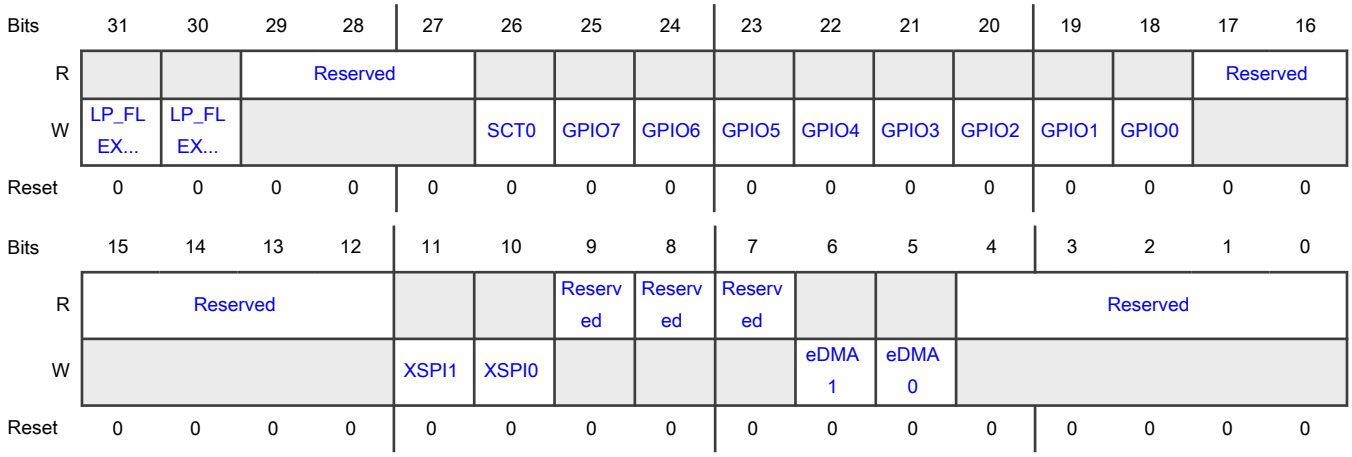
Offset

Register	Offset
PRSTCTL2_SET	48h

Function

Writing one to a bit position in PRSTCTL2_SET sets the corresponding position in PRSTCTL2.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Reset Sets LP_FLEXCOMM1 reset. 0b - No effect 1b - Sets reset.
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Reset Sets LP_FLEXCOMM0 reset. 0b - No effect 1b - Sets reset.
29-27 —	Reserved
26 SCT0	SCT0 Reset Sets SCT0 reset. **NOTE** Writing 1 to this field will reset the SCT0 module and the SCT0_TRIG0 - SCT0_TRIG6 registers of INPUTMUX0 module. 0b - No effect 1b - Sets reset.
25 GPIO7	GPIO7 Reset Sets GPIO7 reset. 0b - No effect 1b - Sets reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 GPIO6	GPIO6 Reset Sets GPIO6 reset. 0b - No effect 1b - Sets reset.
23 GPIO5	GPIO5 Reset Sets GPIO5 reset. 0b - No effect 1b - Sets reset.
22 GPIO4	GPIO4 Reset Sets GPIO4 reset. 0b - No effect 1b - Sets reset.
21 GPIO3	GPIO3 Reset Sets GPIO3 reset. 0b - No effect 1b - Sets reset.
20 GPIO2	GPIO2 Reset Sets GPIO2 reset. 0b - No effect 1b - Sets reset.
19 GPIO1	GPIO1 Reset Sets GPIO1 reset. 0b - No effect 1b - Sets reset.
18 GPIO0	GPIO0 Reset Sets GPIO0 reset. 0b - No effect 1b - Sets reset.
17-12 —	Reserved
11	XSPI1 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
XSPI1	Sets XSPI1, CACHE64_CTRL1 and MMU1 reset. 0b - No effect 1b - Sets reset.
10 XSPI0	XSPI0 Reset Sets XSPI0, CACHE64_CTRL0 and MMU0 reset. 0b - No effect 1b - Sets reset.
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 eDMA1	eDMA1 Reset Sets eDMA1 reset. 0b - No effect 1b - Sets reset.
5 eDMA0	eDMA0 Reset Sets eDMA0 reset. 0b - No effect 1b - Sets reset.
4-0 —	Reserved

31.5.1.10 Compute Domain Peripheral Reset Control 3 SET (PRSTCTL3_SET)

Offset

Register	Offset
PRSTCTL3_SET	4Ch

Function

Writing one to a bit position in PRSTCTL3_SET sets the corresponding position in PRSTCTL3.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			Reserved									Reserved				
W	MU4	SEMA42_4			UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0			CRC0	I3C1	I3C0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R				Reserved												
W	SAI2	SAI1	SAI0		LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MU4	MU4 Reset Sets MU4 reset. 0b - No effect 1b - Sets reset.
30 SEMA42_4	SEMA42_4 Reset Sets SEMA42_4 reset. 0b - No effect 1b - Sets reset.
29-28 —	Reserved
27 UTICK0	UTICK0 Reset Sets UTICK0 reset. 0b - No effect 1b - Sets reset.
26 MRT0	MRT0 Reset Sets MRT0 reset. 0b - No effect 1b - Sets reset.
25 CTIMER4	CTIMER4 Reset Sets CTIMER4 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
24 CTIMER3	CTIMER3 Reset Sets CTIMER3 reset. 0b - No effect 1b - Sets reset.
23 CTIMER2	CTIMER2 Reset Sets CTIMER2 reset. 0b - No effect 1b - Sets reset.
22 CTIMER1	CTIMER1 Reset Sets CTIMER1 reset. 0b - No effect 1b - Sets reset.
21 CTIMER0	CTIMER0 Reset Sets CTIMER0 reset. 0b - No effect 1b - Sets reset.
20-19 —	Reserved
18 CRC0	CRC0 Reset Sets CRC0 reset. 0b - No effect 1b - Sets reset.
17 I3C1	I3C1 Reset Sets I3C1 reset. 0b - No effect 1b - Sets reset.
16 I3C0	I3C0 Reset Sets I3C0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
15 SAI2	SAI2 Reset Sets SAI2 reset. 0b - No effect 1b - Sets reset.
14 SAI1	SAI1 Reset Sets SAI1 reset. 0b - No effect 1b - Sets reset.
13 SAI0	SAI0 Reset Sets SAI0 reset. 0b - No effect 1b - Sets reset.
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Reset Sets LP_FLEXCOMM13 reset. 0b - No effect 1b - Sets reset.
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Reset Sets LP_FLEXCOMM12 reset. 0b - No effect 1b - Sets reset.
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Reset Sets LP_FLEXCOMM11 reset. 0b - No effect 1b - Sets reset.
8 LP_FLEXCOMM10	LP_FLEXCOMM10 Reset Sets LP_FLEXCOMM10 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Reset Sets LP_FLEXCOMM9 reset. 0b - No effect 1b - Sets reset.
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Reset Sets LP_FLEXCOMM8 reset. 0b - No effect 1b - Sets reset.
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Reset Sets LP_FLEXCOMM7 reset. 0b - No effect 1b - Sets reset.
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Reset Sets LP_FLEXCOMM6 reset. 0b - No effect 1b - Sets reset.
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Reset Sets LP_FLEXCOMM5 reset. 0b - No effect 1b - Sets reset.
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Reset Sets LP_FLEXCOMM4 reset. 0b - No effect 1b - Sets reset.
1 LP_FLEXCOMM3	LP_FLEXCOMM3 Reset Sets LP_FLEXCOMM3 reset. 0b - No effect 1b - Sets reset.
0	LP_FLEXCOMM2 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM2	Sets LP_FLEXCOMM2 reset. 0b - No effect 1b - Sets reset.

31.5.1.11 Compute Domain Peripheral Reset Control 4 SET (PRSTCTL4_SET)

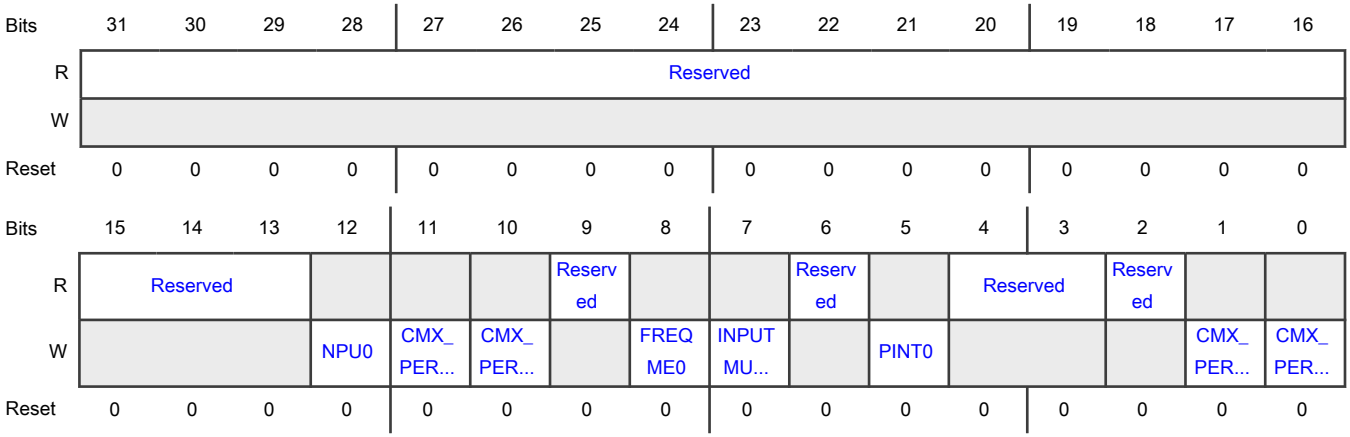
Offset

Register	Offset
PRSTCTL4_SET	50h

Function

Writing one to a bit position in PRSTCTL4_SET sets the corresponding position in PRSTCTL4.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 NPU0	NPU0 Reset Sets NPU0 reset. 0b - No effect 1b - Sets reset.
11	CMX_PERFMON1 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
CMX_PERFMON1	Sets CMX_PERFMON1 reset. 0b - No effect 1b - Sets reset.
10 CMX_PERFMON0	CMX_PERFMON0 Reset Sets CMX_PERFMON0 reset. 0b - No effect 1b - Sets reset.
9 —	Reserved
8 FREQME0	FREQME0 Reset Sets FREQME0 reset. 0b - No effect 1b - Sets reset.
7 INPUTMUX0	INPUTMUX0 Reset Sets INPUTMUX0 reset. **NOTE** Writing 1 to this field will also reset the following registers of INPUTMUX0 module: • HIFI4_INTERRUPT0 - HIFI4_INTERRUPT26 • LP_FLEXCOMM0_TRIG - LP_FLEXCOMM13_TRIG • CTIMER0_CAP0 - CTIMER4_CAP3 • CTIMER0_TRIG - CTIMER4_TRIG • FREQME_TRIG0 - FREQME_TRIG1 • EZHV_TRIG0 - EZHV_TRIG7 • FLEXIO_TRIG0 - FLEXIO_TRIG3 0b - No effect 1b - Sets reset.
6 —	Reserved
5 PINT0	PINT0 Reset Sets PINT0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Writing 1 to this field will reset the PINT0 module and the PINT0_TRIG0 - PINT0_TRIG7 registers of INPUTMUX0 module. 0b - No effect 1b - Sets reset.
4-3 —	Reserved
2 —	Reserved
1 CMX_PERFMON3	CMX_PERFMON3 Reset Sets CMX_PERFMON3 reset. 0b - No effect 1b - Sets reset.
0 CMX_PERFMON2	CMX_PERFMON2 Reset Sets CMX_PERFMON2 reset. 0b - No effect 1b - Sets reset.

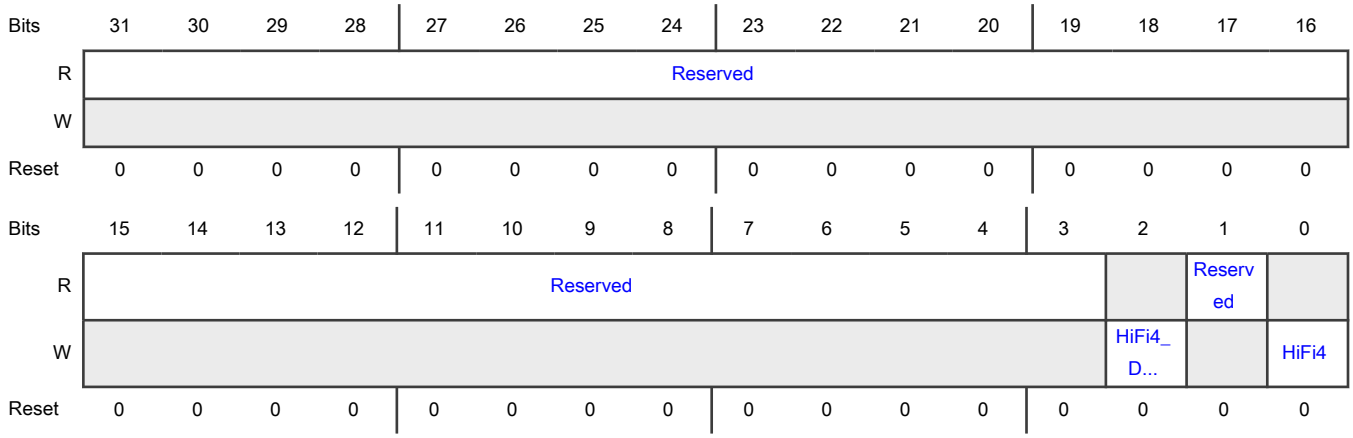
31.5.1.12 DSP Domain Peripheral Reset Control 5 SET (PRSTCTL5_SET)

Offset

Register	Offset
PRSTCTL5_SET	54h

Function
Writing one to a bit position in PRSTCTL5_SET sets the corresponding position in PRSTCTL5.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 HiFi4_DEBUG	HiFi4 Debug Reset Sets HiFi4 debug reset. 0b - No effect 1b - Sets reset.
1 —	Reserved
0 HiFi4	HiFi4 Reset Sets HiFi4 reset. 0b - No effect 1b - Sets reset.

31.5.1.13 Common Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)

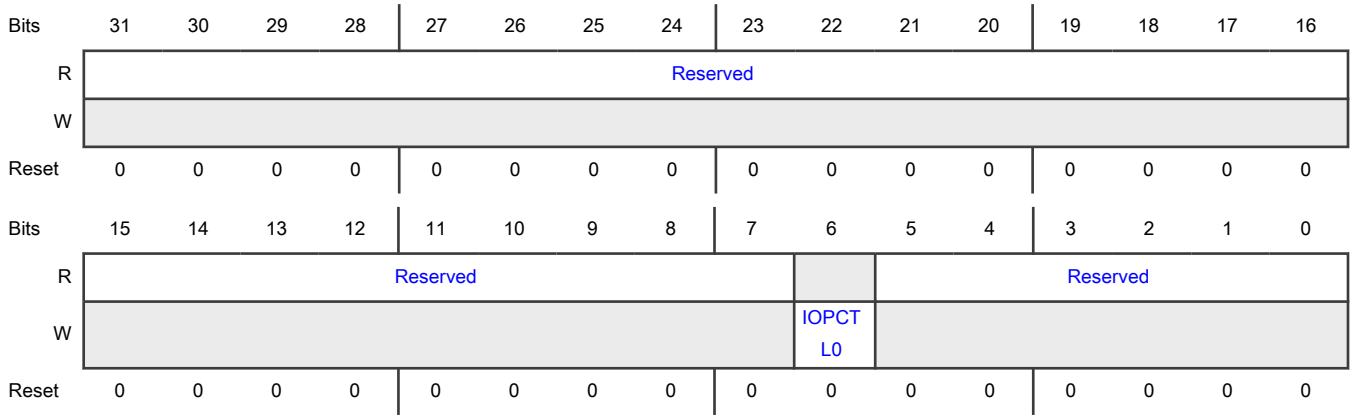
Offset

Register	Offset
PRSTCTL0_CLR	70h

Function

Writing one to a bit position in PRSTCTL0_CLR clears the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31-7 —	Reserved
6 IOPCTL0	IOPCTL0 Reset Clears IOPCTL0 reset. 0b - No effect 1b - Clears reset.
5-0 —	Reserved

31.5.1.14 Compute Domain Peripheral Reset Control 2 CLR (PRSTCTL2_CLR)

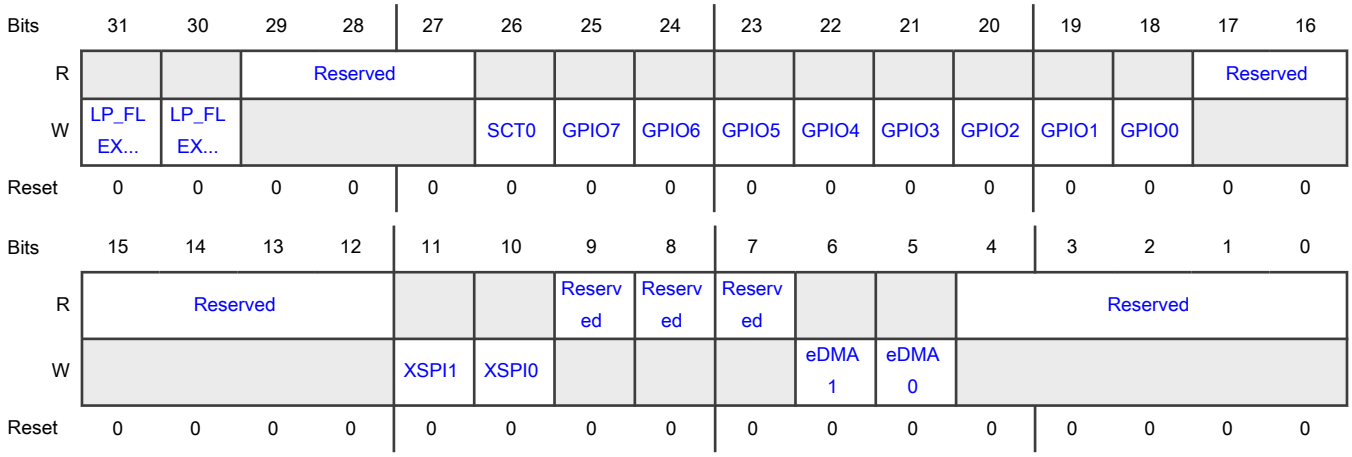
Offset

Register	Offset
PRSTCTL2_CLR	78h

Function

Writing one to a bit position in PRSTCTL2_CLR clears the corresponding position in PRSTCTL2.

Diagram



Fields

Field	Function
31 LP_FLEXCOMM1	LP_FLEXCOMM1 Reset Clears LP_FLEXCOMM1 reset. 0b - No effect 1b - Clears reset.
30 LP_FLEXCOMM0	LP_FLEXCOMM0 Reset Clears LP_FLEXCOMM0 reset. 0b - No effect 1b - Clears reset.
29-27 —	Reserved
26 SCT0	SCT0 Reset Clears SCT0 reset. 0b - No effect 1b - Clears reset.
25 GPIO7	GPIO7 Reset Clears GPIO7 reset. 0b - No effect 1b - Clears reset.
24 GPIO6	GPIO6 Reset Clears GPIO6 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
23 GPIO5	GPIO5 Reset Clears GPIO5 reset. 0b - No effect 1b - Clears reset.
22 GPIO4	GPIO4 Reset Clears GPIO4 reset. 0b - No effect 1b - Clears reset.
21 GPIO3	GPIO3 Reset Clears GPIO3 reset. 0b - No effect 1b - Clears reset.
20 GPIO2	GPIO2 Reset Clears GPIO2 reset. 0b - No effect 1b - Clears reset.
19 GPIO1	GPIO1 Reset Clears GPIO1 reset. 0b - No effect 1b - Clears reset.
18 GPIO0	GPIO0 Reset Clears GPIO0 reset. 0b - No effect 1b - Clears reset.
17-12 —	Reserved
11 XSPI1	XSPI1 Reset Clears XSPI1, CACHE64_CTRL1 and MMU1 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
10 XSPI0	XSPI0 Reset Clears XSPI0, CACHE64_CTRL0 and MMU0 reset. 0b - No effect 1b - Clears reset.
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 eDMA1	eDMA1 Reset Clears eDMA1 reset. 0b - No effect 1b - Clears reset.
5 eDMA0	eDMA0 Reset Clears eDMA0 reset. 0b - No effect 1b - Clears reset.
4-0 —	Reserved

31.5.1.15 Compute Domain Peripheral Reset Control 3 CLR (PRSTCTL3_CLR)

Offset

Register	Offset
PRSTCTL3_CLR	7Ch

Function

Writing one to a bit position in PRSTCTL3_CLR clears the corresponding position in PRSTCTL3.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			Reserved									Reserved				
W	MU4	SEMA42_4			UTICK0	MRT0	CTIMER4	CTIMER3	CTIMER2	CTIMER1	CTIMER0			CRC0	I3C1	I3C0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R				Reserved												
W	SAI2	SAI1	SAI0		LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MU4	MU4 Reset Clears MU4 reset. 0b - No effect 1b - Clears reset.
30 SEMA42_4	SEMA42_4 Reset Clears SEMA42_4 reset. 0b - No effect 1b - Clears reset.
29-28 —	Reserved
27 UTICK0	UTICK0 Reset Clears UTICK0 reset. 0b - No effect 1b - Clears reset.
26 MRT0	MRT0 Reset Clears MRT0 reset. 0b - No effect 1b - Clears reset.
25 CTIMER4	CTIMER4 Reset Clears CTIMER4 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
24 CTIMER3	CTIMER3 Reset Clears CTIMER3 reset. 0b - No effect 1b - Clears reset.
23 CTIMER2	CTIMER2 Reset Clears CTIMER2 reset. 0b - No effect 1b - Clears reset.
22 CTIMER1	CTIMER1 Reset Clears CTIMER1 reset. 0b - No effect 1b - Clears reset.
21 CTIMER0	CTIMER0 Reset Clears CTIMER0 reset. 0b - No effect 1b - Clears reset.
20-19 —	Reserved
18 CRC0	CRC0 Reset Clears CRC0 reset. 0b - No effect 1b - Clears reset.
17 I3C1	I3C1 Reset Clears I3C1 reset. 0b - No effect 1b - Clears reset.
16 I3C0	I3C0 Reset Clears I3C0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
15 SAI2	SAI2 Reset Clears SAI2 reset. 0b - No effect 1b - Clears reset.
14 SAI1	SAI1 Reset Clears SAI1 reset. 0b - No effect 1b - Clears reset.
13 SAI0	SAI0 Reset Clears SAI0 reset. 0b - No effect 1b - Clears reset.
12 —	Reserved
11 LP_FLEXCOMM13	LP_FLEXCOMM13 Reset Clears LP_FLEXCOMM13 reset. 0b - No effect 1b - Clears reset.
10 LP_FLEXCOMM12	LP_FLEXCOMM12 Reset Clears LP_FLEXCOMM12 reset. 0b - No effect 1b - Clears reset.
9 LP_FLEXCOMM11	LP_FLEXCOMM11 Reset Clears LP_FLEXCOMM11 reset. 0b - No effect 1b - Clears reset.
8 LP_FLEXCOMM10	LP_FLEXCOMM10 Reset Clears LP_FLEXCOMM10 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
7 LP_FLEXCOMM9	LP_FLEXCOMM9 Reset Clears LP_FLEXCOMM9 reset. 0b - No effect 1b - Clears reset.
6 LP_FLEXCOMM8	LP_FLEXCOMM8 Reset Clears LP_FLEXCOMM8 reset. 0b - No effect 1b - Clears reset.
5 LP_FLEXCOMM7	LP_FLEXCOMM7 Reset Clears LP_FLEXCOMM7 reset. 0b - No effect 1b - Clears reset.
4 LP_FLEXCOMM6	LP_FLEXCOMM6 Reset Clears LP_FLEXCOMM6 reset. 0b - No effect 1b - Clears reset.
3 LP_FLEXCOMM5	LP_FLEXCOMM5 Reset Clears LP_FLEXCOMM5 reset. 0b - No effect 1b - Clears reset.
2 LP_FLEXCOMM4	LP_FLEXCOMM4 Reset Clears LP_FLEXCOMM4 reset. 0b - No effect 1b - Clears reset.
1 LP_FLEXCOMM3	LP_FLEXCOMM3 Reset Clears LP_FLEXCOMM3 reset. 0b - No effect 1b - Clears reset.
0	LP_FLEXCOMM2 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM2	Clears LP_FLEXCOMM2 reset. 0b - No effect 1b - Clears reset.

31.5.1.16 Compute Domain Peripheral Reset Control 4 CLR (PRSTCTL4_CLR)

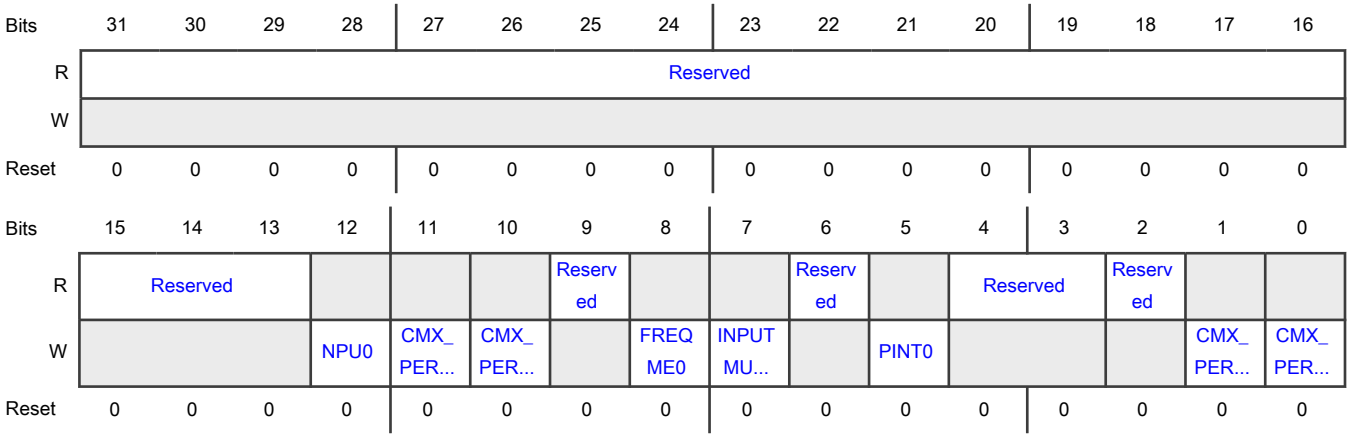
Offset

Register	Offset
PRSTCTL4_CLR	80h

Function

Writing one to a bit position in PRSTCTL4_CLR clears the corresponding position in PRSTCTL4.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 NPU0	NPU0 Reset Clears NPU0 reset. 0b - No effect 1b - Clears reset.
11	CMX_PERFMON1 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
CMX_PERFMON1	Clears CMX_PERFMON1 reset. 0b - No effect 1b - Clears reset.
10 CMX_PERFMON0	CMX_PERFMON0 Reset Clears CMX_PERFMON0 reset. 0b - No effect 1b - Clears reset.
9 —	Reserved
8 FREQME0	FREQME0 Reset Clears FREQME0 reset. 0b - No effect 1b - Clears reset.
7 INPUTMUX0	INPUTMUX0 Reset Clears INPUTMUX0 reset. 0b - No effect 1b - Clears reset.
6 —	Reserved
5 PINT0	PINT0 Reset Clears PINT0 reset. 0b - No effect 1b - Clears reset.
4-3 —	Reserved
2 —	Reserved
1 CMX_PERFMON3	CMX_PERFMON3 Reset Clears CMX_PERFMON3 reset. 0b - No effect 1b - Clears reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 CMX_PERFMON2	CMX_PERFMON2 Reset Clears CMX_PERFMON2 reset. 0b - No effect 1b - Clears reset.

31.5.1.17 DSP Domain Peripheral Reset Control 5 CLR (PRSTCTL5_CLR)

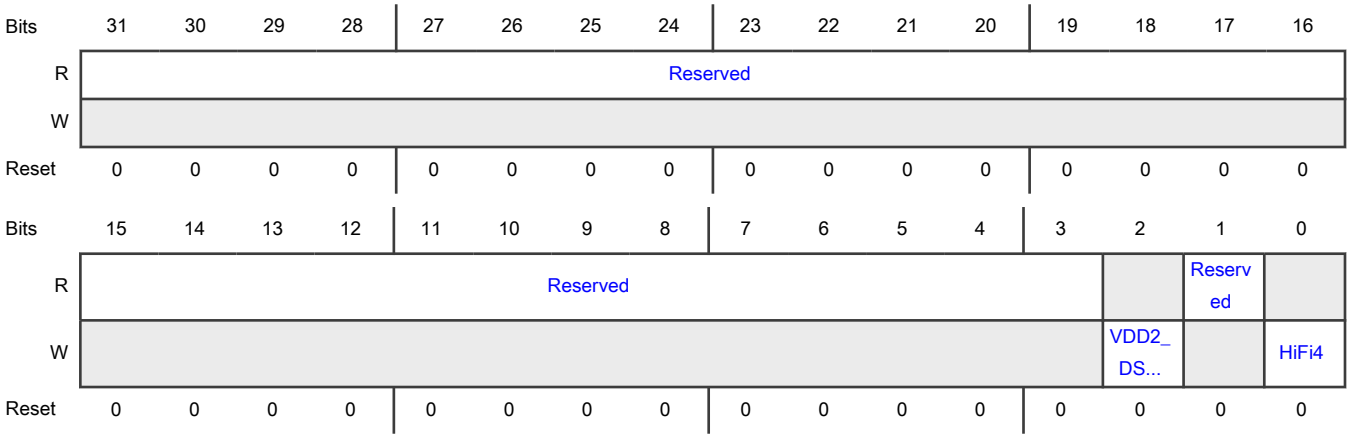
Offset

Register	Offset
PRSTCTL5_CLR	84h

Function

Writing one to a bit position in PRSTCTL5_CLR clears the corresponding position in PRSTCTL5.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 VDD2_DSP_DEBUG	HiFi4 Debug Reset Clears HiFi4 debug reset. 0b - No effect 1b - Clears reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 —	Reserved
0 HiFi4	HiFi4 Reset Clears HiFi4 reset. 0b - No effect 1b - Clears reset.

31.5.2 RSTCTL1 register descriptions

This section includes the memory map and detailed descriptions of registers for Reset Controllers 1. This reset controller controls private peripherals in the VDD1_SENSE power domain on the VDD1 power rail.

31.5.2.1 RSTCTL1 memory map

RSTCTL1 base address: 4004_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Sense Domain Private Peripheral Reset Control 0 (PRSTCTL0)	32	RW	133F_FFF7h
40h	Sense Domain Private Peripheral Reset Control 0 SET (PRSTCTL0_SET)	32	W	0000_0000h
70h	Sense Domain Private Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)	32	W	0000_0000h

31.5.2.2 Sense Domain Private Peripheral Reset Control 0 (PRSTCTL0)

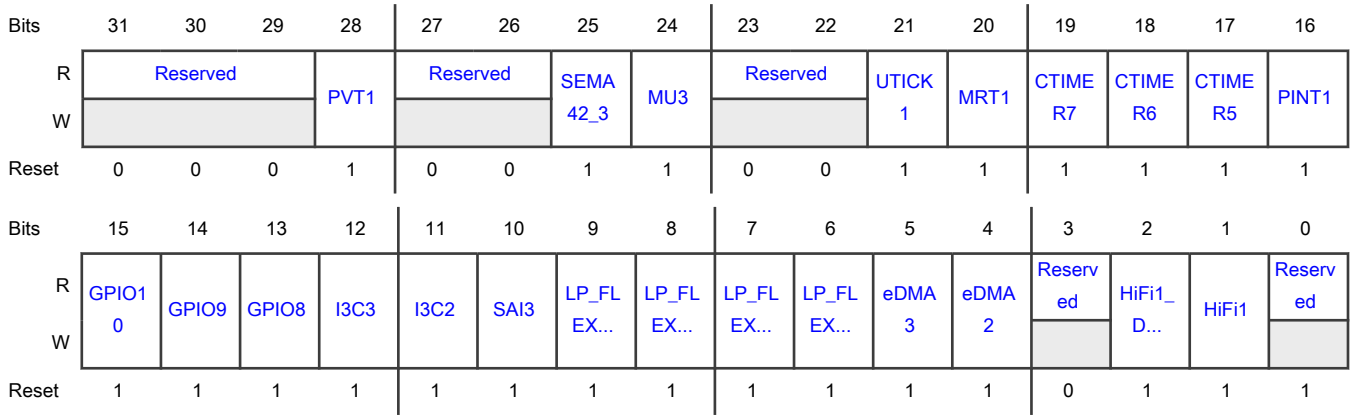
Offset

Register	Offset
PRSTCTL0	10h

Function

Allows software to reset individual peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

Diagram



Fields

Field	Function
31-29 —	Reserved
28 PVT1	PVT1 Reset Enables PVT1 reset control. 0b - Clears reset. 1b - Sets reset.
27-26 —	Reserved
25 SEMA42_3	SEMA42_3 Reset Enables SEMA42_3 reset control. 0b - Clears reset. 1b - Sets reset.
24 MU3	MU3 Reset Enables MU3 reset control. 0b - Clears reset. 1b - Sets reset.
23-22 —	Reserved
21 UTICK1	UTICK1 Reset Enables UTICK1 reset control.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Clears reset. 1b - Sets reset.
20 MRT1	MRT1 Reset Enables MRT1 reset control. 0b - Clears reset. 1b - Sets reset.
19 CTIMER7	CTIMER7 Reset Enables CTIMER7 reset control. 0b - Clears reset. 1b - Sets reset.
18 CTIMER6	CTIMER6 Reset Enables CTIMER6 reset control. 0b - Clears reset. 1b - Sets reset.
17 CTIMER5	CTIMER5 Reset Enables CTIMER5 reset control. 0b - Clears reset. 1b - Sets reset.
16 PINT1	PINT1 Reset Enables PINT1 reset control. **NOTE** Writing 1 to this field will reset the PINT1 module and the PINT1_TRIG0 - PINT1_TRIG3 registers of INPUTMUX1 module. 0b - Clears reset. 1b - Sets reset.
15 GPIO10	GPIO10 Reset Enables GPIO10 reset control. 0b - Clears reset. 1b - Sets reset.
14 GPIO9	GPIO9 Reset Enables GPIO9 reset control.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Clears reset. 1b - Sets reset.
13 GPIO8	GPIO8 Reset Enables GPIO8 reset control. 0b - Clears reset. 1b - Sets reset.
12 I3C3	I3C3 Reset Enables I3C3 reset control. 0b - Clears reset. 1b - Sets reset.
11 I3C2	I3C2 Reset Enables I3C2 reset control. 0b - Clears reset. 1b - Sets reset.
10 SAI3	SAI3 Reset Enables SAI3 reset control. 0b - Clears reset. 1b - Sets reset.
9 LP_FLEXCOMM20	LP_FLEXCOMM20 Reset Enables LP_FLEXCOMM20 reset control. 0b - Clears reset. 1b - Sets reset.
8 LP_FLEXCOMM19	LP_FLEXCOMM19 Reset Enables LP_FLEXCOMM19 reset control. 0b - Clears reset. 1b - Sets reset.
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Reset Enables LP_FLEXCOMM18 reset control. 0b - Clears reset. 1b - Sets reset.
6	LP_FLEXCOMM17 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM17	Enables LP_FLEXCOMM17 reset control. 0b - Clears reset. 1b - Sets reset.
5 eDMA3	eDMA3 Reset Enables eDMA3 reset control. 0b - Clears reset. 1b - Sets reset.
4 eDMA2	eDMA2 Reset Enables eDMA2 reset control. 0b - Clears reset. 1b - Sets reset.
3 —	Reserved
2 HiFi1_DEBUG	HiFi1 Debug Reset Enables HiFi1 Debug reset control. 0b - Clears reset. 1b - Sets reset.
1 HiFi1	HiFi1 Reset Enables HiFi1 reset control. 0b - Clears reset. 1b - Sets reset.
0 —	Reserved

31.5.2.3 Sense Domain Private Peripheral Reset Control 0 SET (PRSTCTL0_SET)

Offset

Register	Offset
PRSTCTL0_SET	40h

Function
Writing one to a bit position in PRSTCTL0_SET sets the corresponding position in PRSTCTL0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved							
W				PVT1			SEMA42_3	MU3			UTICK1	MRT1	CTIMER7	CTIMER6	CTIMER5	PINT1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													Reserved			Reserved
W	GPIO10	GPIO9	GPIO8	I3C3	I3C2	SAI3	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	eDMA3	eDMA2		HiFi1_D...	HiFi1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28 PVT1	PVT1 Reset Sets PVT1 reset. 0b - No effect 1b - Sets reset.
27-26 —	Reserved
25 SEMA42_3	SEMA42_3 Reset Sets SEMA42_3 reset. 0b - No effect 1b - Sets reset.
24 MU3	MU3 Reset Sets MU3 reset. 0b - No effect 1b - Sets reset.
23-22 —	Reserved
21 UTICK1	UTICK1 Reset Sets UTICK1 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
20 MRT1	MRT1 Reset Sets MRT1 reset. 0b - No effect 1b - Sets reset.
19 CTIMER7	CTIMER7 Reset Sets CTIMER7 reset. 0b - No effect 1b - Sets reset.
18 CTIMER6	CTIMER6 Reset Sets CTIMER6 reset. 0b - No effect 1b - Sets reset.
17 CTIMER5	CTIMER5 Reset Sets CTIMER5 reset. 0b - No effect 1b - Sets reset.
16 PINT1	PINT1 Reset Sets PINT1 reset. **NOTE** Writing 1 to this field will reset the PINT1 module and the PINT1_TRIG0 - PINT1_TRIG3 registers of INPUTMUX1 module. 0b - No effect 1b - Sets reset.
15 GPIO10	GPIO10 Reset Sets GPIO10 reset. 0b - No effect 1b - Sets reset.
14 GPIO9	GPIO9 Reset Sets GPIO9 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
13 GPIO8	GPIO8 Reset Sets GPIO8 reset. 0b - No effect 1b - Sets reset.
12 I3C3	I3C3 Reset Sets I3C3 reset. 0b - No effect 1b - Sets reset.
11 I3C2	I3C2 Reset Sets I3C2 reset. 0b - No effect 1b - Sets reset.
10 SAI3	SAI3 Reset Sets SAI3 reset. 0b - No effect 1b - Sets reset.
9 LP_FLEXCOMM20	LP_FLEXCOMM20 Reset Sets LP_FLEXCOMM20 reset. 0b - No effect 1b - Sets reset.
8 LP_FLEXCOMM19	LP_FLEXCOMM19 Reset Sets LP_FLEXCOMM19 reset. 0b - No effect 1b - Sets reset.
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Reset Sets LP_FLEXCOMM18 reset. 0b - No effect 1b - Sets reset.
6	LP_FLEXCOMM17 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
LP_FLEXCOMM17	Sets LP_FLEXCOMM17 reset. 0b - No effect 1b - Sets reset.
5 eDMA3	eDMA3 Reset Sets eDMA3 reset. 0b - No effect 1b - Sets reset.
4 eDMA2	eDMA2 Reset Sets eDMA2 reset. 0b - No effect 1b - Sets reset.
3 —	Reserved
2 HiFi1_DEBUG	HiFi1 Debug Reset Sets HiFi1 Debug reset. 0b - No effect. 1b - Sets reset.
1 HiFi1	HiFi1 Reset Sets HiFi1 reset. 0b - No effect 1b - Sets reset.
0 —	Reserved

31.5.2.4 Sense Domain Private Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)

Offset

Register	Offset
PRSTCTL0_CLR	70h

Function

Writing one to a bit position in PRSTCTL0_CLR clears the corresponding position in PRSTCTL0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved							
W				PVT1			SEMA42_3	MU3			UTICK1	MRT1	CTIMER7	CTIMER6	CTIMER5	PINT1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													Reserved			Reserved
W	GPIO10	GPIO9	GPIO8	I3C3	I3C2	SAI3	LP_FLEX...	LP_FLEX...	LP_FLEX...	LP_FLEX...	eDMA3	eDMA2		HiFi1_D...	HiFi1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28 PVT1	PVT1 Reset Clears PVT1 reset. 0b - No effect 1b - Clears reset.
27-26 —	Reserved
25 SEMA42_3	SEMA42_3 Reset Clears SEMA42_3 reset. 0b - No effect 1b - Clears reset.
24 MU3	MU3 Reset Clears MU3 reset. 0b - No effect 1b - Clears reset.
23-22 —	Reserved
21 UTICK1	UTICK1 Reset Clears UTICK1 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
20 MRT1	MRT1 Reset Clears MRT1 reset. 0b - No effect 1b - Clears reset.
19 CTIMER7	CTIMER7 Reset Clears CTIMER7 reset. 0b - No effect 1b - Clears reset.
18 CTIMER6	CTIMER6 Reset Clears CTIMER6 reset. 0b - No effect 1b - Clears reset.
17 CTIMER5	CTIMER5 Reset Clears CTIMER5 reset. 0b - No effect 1b - Clears reset.
16 PINT1	PINT1 Reset Clears PINT1 reset. 0b - No effect 1b - Clears reset.
15 GPIO10	GPIO10 Reset Clears GPIO10 reset. 0b - No effect 1b - Clears reset.
14 GPIO9	GPIO9 Reset Clears GPIO9 reset. 0b - No effect 1b - Clears reset.
13	GPIO8 Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
GPIO8	Clears GPIO8 reset. 0b - No effect 1b - Clears reset.
12 I3C3	I3C3 Reset Clears I3C3 reset. 0b - No effect 1b - Clears reset.
11 I3C2	I3C2 Reset Clears I3C2 reset. 0b - No effect 1b - Clears reset.
10 SAI3	SAI3 Reset Clears SAI3 reset. 0b - No effect 1b - Clears reset.
9 LP_FLEXCOMM20	LP_FLEXCOMM20 Reset Clears LP_FLEXCOMM20 reset. 0b - No effect 1b - Clears reset.
8 LP_FLEXCOMM19	LP_FLEXCOMM19 Reset Clears LP_FLEXCOMM19 reset. 0b - No effect 1b - Clears reset.
7 LP_FLEXCOMM18	LP_FLEXCOMM18 Reset Clears LP_FLEXCOMM18 reset. 0b - No effect 1b - Clears reset.
6 LP_FLEXCOMM17	LP_FLEXCOMM17 Reset Clears LP_FLEXCOMM17 reset. 0b - No effect 1b - Clears reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
5 eDMA3	eDMA3 Reset Clears eDMA3 reset. 0b - No effect 1b - Clears reset.
4 eDMA2	eDMA2 Reset Clears eDMA2 reset. 0b - No effect 1b - Clears reset.
3 —	Reserved
2 HiFi1_DEBUG	HiFi1 Debug Reset Clears HiFi1 Debug reset. 0b - No effect. 1b - Clears reset
1 HiFi1	HiFi1 Reset Clears HiFi1 reset. 0b - No effect 1b - Clears reset.
0 —	Reserved

31.5.3 RSTCTL2 register descriptions

This section includes the memory map and detailed descriptions of registers for Reset Controllers 2. This reset controller controls peripherals in the VDDN_COM power domain on the VDDN power rail.

31.5.3.1 RSTCTL2 memory map

RSTCTL2 base address: 4006_7000h

Offset	Register	Width (In bits)	Access	Reset value
10h	VDDN Domain Peripheral Reset Control 0 (PRSTCTL0)	32	RW	0000_0002h
40h	VDDN Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)	32	W	0000_0000h
70h	VDDN Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)	32	W	0000_0000h

31.5.3.2 VDDN Domain Peripheral Reset Control 0 (PRSTCTL0)

Offset

Register	Offset
PRSTCTL0	10h

Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														IOPCTL2	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-2 —	Reserved
1 IOPCTL2	IOPCTL2 Reset Enables IOPCTL2 reset control. 0b - Clears reset. 1b - Sets reset.
0 —	Reserved

31.5.3.3 VDDN Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)

Offset

Register	Offset
PRSTCTL0_SET	40h

Function
Writing one to a bit position in PRSTCTL0_SET sets the corresponding position in PRSTCTL0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															Reserved
W															IOPCTL2	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1 IOPCTL2	IOPCTL2 Reset Sets IOPCTL2 reset. 0b - No effect 1b - Sets reset.
0 —	Reserved

31.5.3.4 VDDN Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)

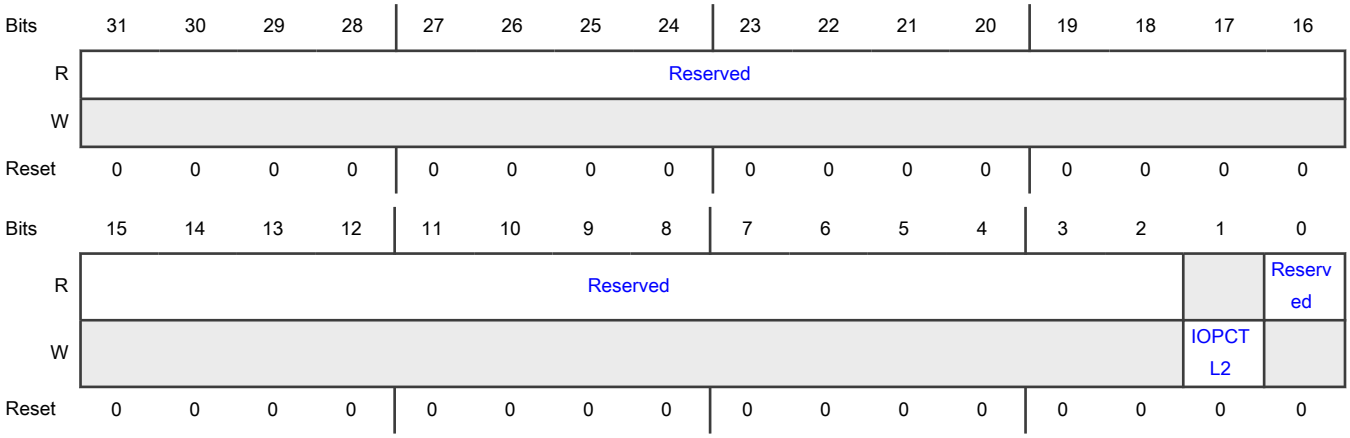
Offset

Register	Offset
PRSTCTL0_CLR	70h

Function

Writing one to a bit position in PRSTCTL0_CLR clears the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 IOPCTL2	IOPCTL2 Reset Clears IOPCTL2 reset. 0b - No effect 1b - Clears reset.
0 —	Reserved

31.5.4 RSTCTL3 register descriptions

This section includes the memory map and detailed descriptions of registers for Reset Controllers 3. This controller controls shared peripherals in the VDD1_SENSE power domain on the VDD1 power rail.

31.5.4.1 RSTCTL3 memory map

RSTCTL3 base address: 4006_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	System Reset Status (SYSRSTSTAT)	32	RW	0000_0001h
4h	Domain Reset Status (DOMRSTSTAT)	32	R	0000_000Bh

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
10h	Sense Domain Peripheral Reset Control 0 (PRSTCTL0)	32	RW	8000_0001h
14h	Sense Domain Peripheral Reset Control 1 (PRSTCTL1)	32	RW	0004_43EEh
40h	Sense Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)	32	W	0000_0000h
44h	Sense Domain Peripheral Reset Control 1 SET (PRSTCTL1_SET)	32	W	0000_0000h
70h	Sense Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)	32	W	0000_0000h
74h	Sense Peripheral Reset Control 1 CLR (PRSTCTL1_CLR)	32	W	0000_0000h

31.5.4.2 System Reset Status (SYSRSTSTAT)

Offset

Register	Offset
SYSRSTSTAT	0h

Function

Shows the status of the latest reset event. The bits are cleared by writing one to any of the bits in this register. The POR event clears all other bits in this register. If another reset signal, for example the external RESET pin, remains asserted after the POR signal is negated, the reset events are set to be detected.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved					WWDT3_R...	WWDT2_R...	WWDT1_R...	WWDT0_R...	CPU1_RE...	CPU0_RE...	Reserved	Reserved	RESETN_...	VDD_POR
W							W1C	W1C	W1C	W1C	W1C	W1C			W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
14-10 —	Reserved
9 WWDT3_RESET	WWDT3 Reset WWDT3 reset is detected. Suggest to cohort with DSP HiFi1 core. **NOTE** This bit is cleared by POR. 0b - No WWDT3 reset event is detected. 1b - WWDT3 reset event is detected.
8 WWDT2_RESET	WWDT2 Reset WWDT2 reset is detected. Suggest to cohort with CPU1. **NOTE** This bit is cleared by POR. **NOTE** During the ROM boot, the ITRC.OUT4_SEL0[IN11_SELn] is configured as 01b. With this configuration, ITRC will clear this bit each time it receives the WWDT2 interrupt which resets the WWDT2. In order to show the correct reset status, you need to configure ITRC.OUT4_SEL0[IN11_SELn] as 10b. See i.MX RT700 Security Reference Manual for more details about ITRC. 0b - No WWDT2 reset event is detected. 1b - WWDT2 reset event is detected.
7 WWDT1_RESET	WWDT1 Reset WWDT1 reset was detected. Suggest to cohort with DSP HiFi4 core. **NOTE** This bit is cleared by POR. **NOTE** During the ROM boot, the ITRC.OUT4_SEL0[IN10_SELn] is configured as 01b. With this configuration, ITRC will clear this bit each time it receives the WWDT1 interrupt which resets the WWDT1. In order to show the correct reset status, you need to configure ITRC.OUT4_SEL0[IN10_SELn] as 10b. See i.MX RT700 Security Reference Manual for more details about ITRC. 0b - No WWDT1 reset event is detected 1b - WWDT1 reset event is detected.
6	WWDT0 Reset WWDT0 reset is detected. Suggest to cohort with CPU0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
WWDT0_RESET	**NOTE** This bit is cleared by POR. **NOTE** During the ROM boot, the ITRC.OUT4_SEL0[IN9_SELn] is configured as 01b. With this configuration, ITRC will clear this bit each time it receives the WWDT0 interrupt which resets the WWDT0. In order to show the correct reset status, you need to configure ITRC.OUT4_SEL0[IN9_SELn] as 10b. See i.MX RT700 Security Reference Manual for more details about ITRC. 0b - No WWDT0 reset event is detected. 1b - WWDT0 reset event is detected.
5 CPU1_RESET	CPU1 Reset CPU1 reset is detected. **NOTE** This bit is cleared by POR. 0b - No CPU1 reset event is detected. 1b - CPU1 reset is detected.
4 CPU0_RESET	CPU0 Reset CPU0 reset is detected. **NOTE** This bit is cleared by POR. 0b - No CPU0 reset event is detected. 1b - CPU0 reset is detected.
3 —	Reserved
2 —	Reserved
1 RESETN_RESET	RESETN Reset RESETN reset is detected. **NOTE** This bit is cleared by POR. 0b - No RESETN event is detected 1b - RESETN event is detected.

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 VDD_POR	VDD Power-On Reset (POR) VDD Power-On Reset (POR) is detected. Any POR for VDD1, VDD2, VDDN, LVD, HVD HW reset will trigger VDD POR. **NOTE** This bit will be set to 1. 0b - No VDD POR event is detected. 1b - VDD POR event is detected.

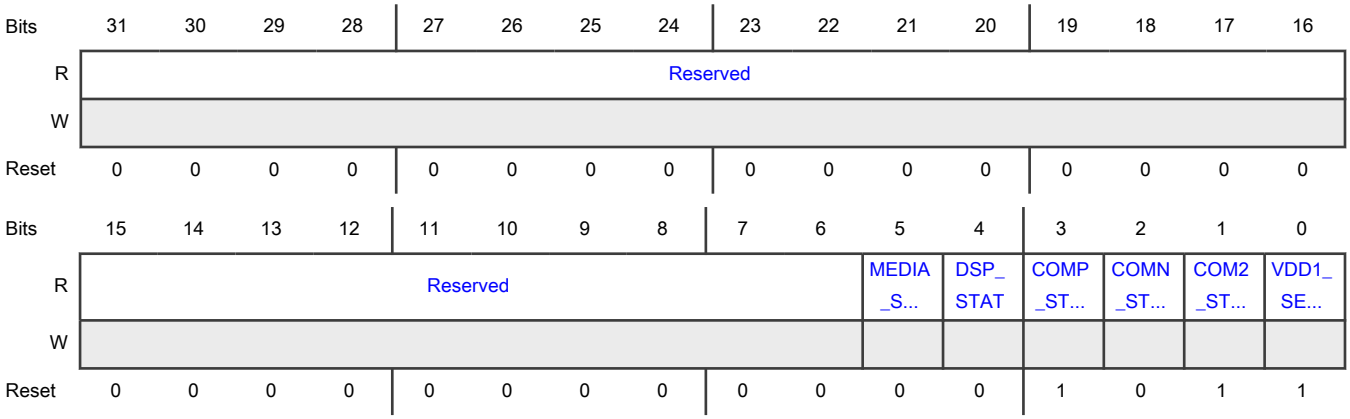
31.5.4.3 Domain Reset Status (DOMRSTSTAT)

Offset

Register	Offset
DOMRSTSTAT	4h

Function
Shows the reset status of domains.

Diagram



Fields

Field	Function
31-6 —	Reserved
5	VDD2_MEDIA Domain Reset State

Table continues on the next page...

Table continued from the previous page...

Field	Function
MEDIA_STAT	Shows the reset status of VDD2_MEDIA domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
4 DSP_STAT	VDD2_DSP Domain Reset State Shows the reset status of VDD2_DSP domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
3 COMP_STAT	VDD2_COMP Domain Reset State Shows the reset status of VDD2_COMP domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
2 COMN_STAT	VDDN_COM Domain Reset State Shows the reset status of VDDN_COM domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
1 COM2_STAT	VDD2_COM (Main) Domain Reset State Shows the reset status of VDD2_COM (Main) domain. 0b - The domain is in reset state. 1b - The domain is in normal run state.
0 VDD1_SENSE_STAT	VDD1_SENSE Domain Reset State Shows the reset status of VDD1_SENSE domain . 0b - The domain is in reset state. 1b - The domain is in normal run state.

31.5.4.4 Sense Domain Peripheral Reset Control 0 (PRSTCTL0)

Offset

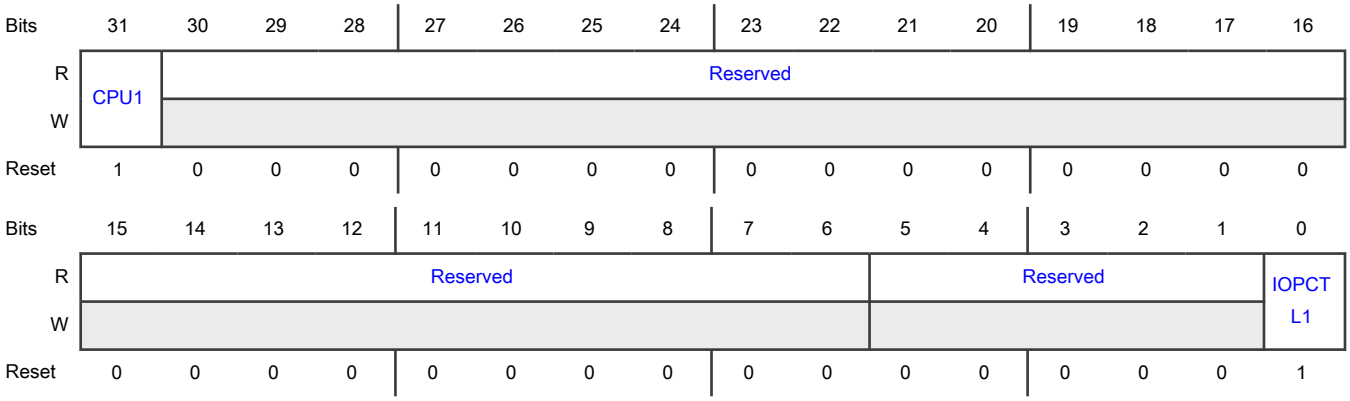
Register	Offset
PRSTCTL0	10h

Function

Allows software to reset individual peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm

reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

Diagram



Fields

Field	Function
31 CPU1	CPU1 Reset Enables CPU1 reset control. 0b - Clears reset. 1b - Sets reset.
30-6 —	Reserved
5-1 —	Reserved
0 IOPCTL1	IOPCTL1 Reset Enables IOPCTL1 reset control. 0b - Clears reset. 1b - Sets reset.

31.5.4.5 Sense Domain Peripheral Reset Control 1 (PRSTCTL1)

Offset

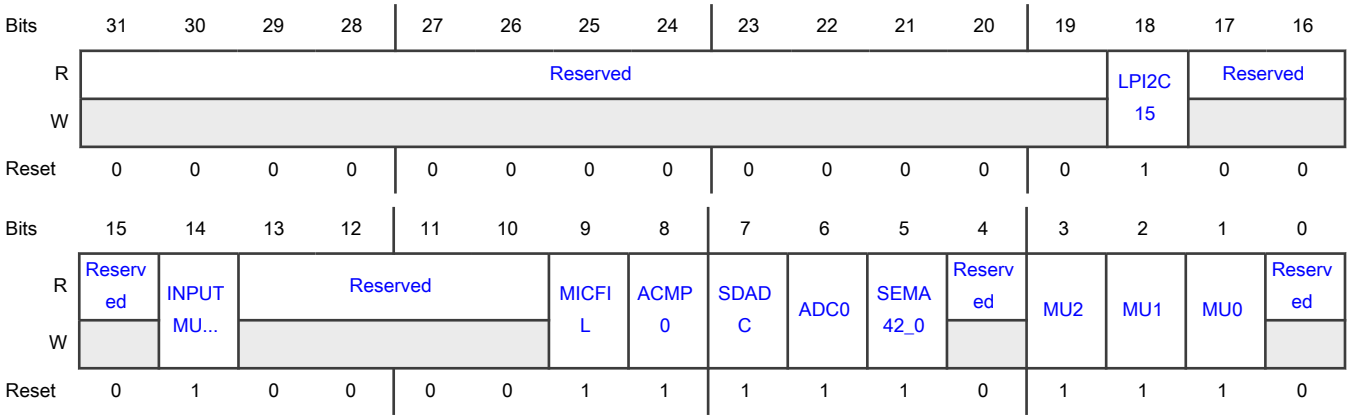
Register	Offset
PRSTCTL1	14h

Function

Allows software to reset individual peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm

reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

Diagram



Fields

Field	Function
31-19 —	Reserved
18 LPI2C15	LPI2C15 Reset Enables LPI2C15 reset control. 0b - Clears reset. 1b - Sets reset.
17-15 —	Reserved
14 INPUTMUX1	INPUTMUX1 Reset Enables INPUTMUX1 reset control. **NOTE** Writing 1 to this field will also reset the following registers of INPUTMUX1 module: • HIFI1_INTERRUPT0 - HIFI1_INTERRUPT26 • LP_FLEXCOMM17_TRIG - LP_FLEXCOMM20_TRIG • ADC0_TRIG0 - ADC0_TRIG1 • CTIMER5_CAP0 - CTIMER7_CAP3 • CTIMER5_TRIG - CTIMER7_TRIG 0b - Clears reset. 1b - Sets reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
13-10 —	Reserved
9 MICFIL	MICFIL Reset Enables reset control over MICFIL.DATACHn registers. 0b - Clears reset. 1b - Sets reset.
8 ACMP0	ACMP0 Reset Enables ACMP0 reset control. 0b - Clears reset. 1b - Sets reset.
7 SDADC	SDADC Reset Enables SDADC reset control. 0b - Clears reset. 1b - Sets reset.
6 ADC0	ADC0 Reset Enables ADC0 reset control. 0b - Clears reset. 1b - Sets reset.
5 SEMA42_0	SEMA42_0 Reset Enables SEMA42_0 reset control. 0b - Clears reset. 1b - Sets reset.
4 —	Reserved
3 MU2	MU2 Reset Enables MU2 reset control. 0b - Clears reset. 1b - Sets reset.
2 MU1	MU1 Reset Enables MU1 reset control.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Clears reset. 1b - Sets reset.
1 MU0	MU0 Reset Enables MU0 reset control. 0b - Clears reset. 1b - Sets reset.
0 —	Reserved

31.5.4.6 Sense Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)

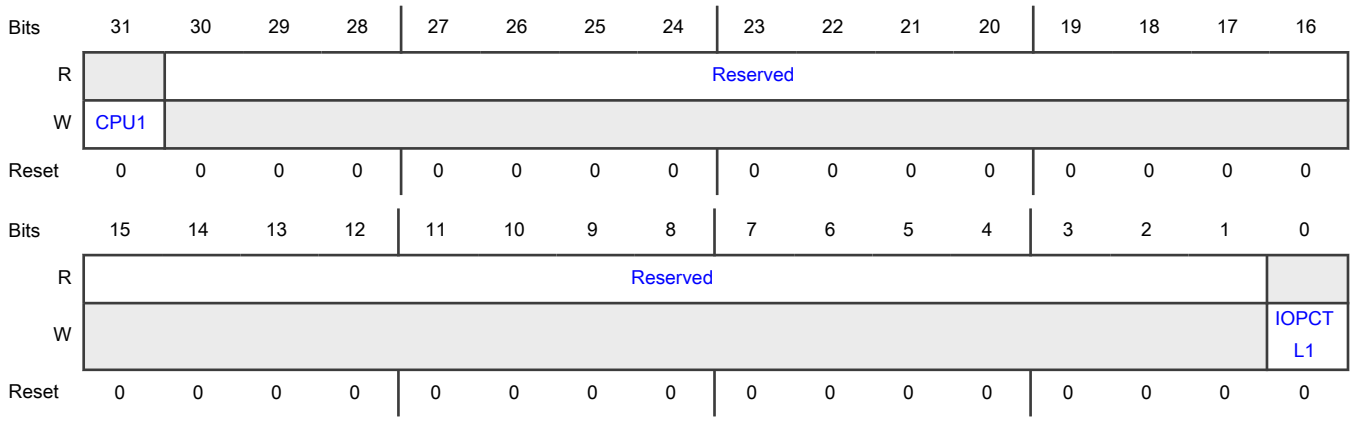
Offset

Register	Offset
PRSTCTL0_SET	40h

Function

Writing one to a bit position in PRSTCTL0_SET sets the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31 CPU1	CPU1 Reset Sets CPU1 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
30-1 —	Reserved
0 IOPCTL1	IOPCTL1 Reset Sets IOPCTL1 reset. 0b - No effect 1b - Sets reset.

31.5.4.7 Sense Domain Peripheral Reset Control 1 SET (PRSTCTL1_SET)

Offset

Register	Offset
PRSTCTL1_SET	44h

Function

Writing one to a bit position in PRSTCTL1_SET sets the corresponding position in PRSTCTL1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														Reserved	
W														LPI2C15		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Reserved									Reserved				Reserved
W		INPUTMU...					MICFIL	ACMP0	SDADC	ADC0	SEMA42_0		MU2	MU1	MU0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-19	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
18 LPI2C15	LPI2C15 Reset Sets LPI2C15 reset. 0b - No effect 1b - Sets reset.
17-15 —	Reserved
14 INPUTMUX1	INPUTMUX1 Reset Sets INPUTMUX1 reset. **NOTE** Writing 1 to this field will also reset the following registers of INPUTMUX1 module: • HIFI1_INTERRUPT0 - HIFI1_INTERRUPT26 • LP_FLEXCOMM17_TRIG - LP_FLEXCOMM20_TRIG • ADC0_TRIG0 - ADC0_TRIG1 • CTIMER5_CAP0 - CTIMER7_CAP3 • CTIMER5_TRIG - CTIMER7_TRIG 0b - No effect 1b - Sets reset.
13-10 —	Reserved
9 MICFIL	MICFIL Reset Sets reset over MICFIL.DATACHn registers. 0b - No effect 1b - Sets reset.
8 ACMP0	ACMP0 Reset Sets ACMP0 reset. 0b - No effect 1b - Sets reset.
7 SDADC	SDADC Reset Sets SDADC reset. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets reset.
6 ADC0	ADC0 Reset Sets ADC0 reset. 0b - No effect 1b - Sets reset.
5 SEMA42_0	SEMA42_0 Reset Sets SEMA42_0 reset. 0b - No effect 1b - Sets reset.
4 —	Reserved
3 MU2	MU2 Reset Sets MU2 reset. 0b - No effect 1b - Sets reset.
2 MU1	MU1 Reset Sets MU1 reset. 0b - No effect 1b - Sets reset.
1 MU0	MU0 Reset Sets MU0 reset. 0b - No effect 1b - Sets reset.
0 —	Reserved

31.5.4.8 Sense Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)

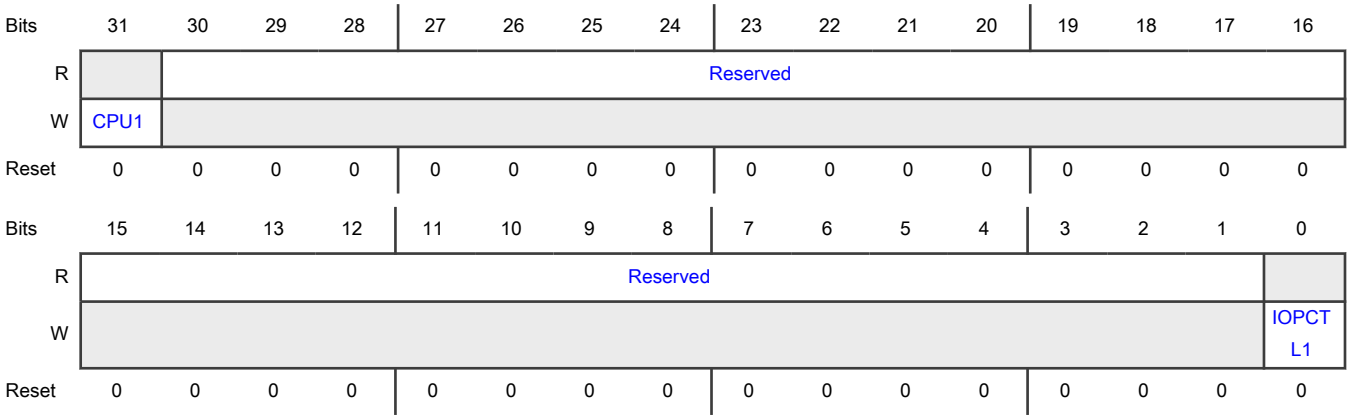
Offset

Register	Offset
PRSTCTL0_CLR	70h

Function

Writing one to a bit position in PRSTCTL0_CLR clears the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31 CPU1	CPU1 Reset Clears CPU1 reset. 0b - No effect 1b - Clears reset.
30-1 —	Reserved
0 IOPCTL1	IOPCTL1 Reset Clears IOPCTL1 reset. 0b - No effect 1b - Clears reset.

31.5.4.9 Sense Peripheral Reset Control 1 CLR (PRSTCTL1_CLR)

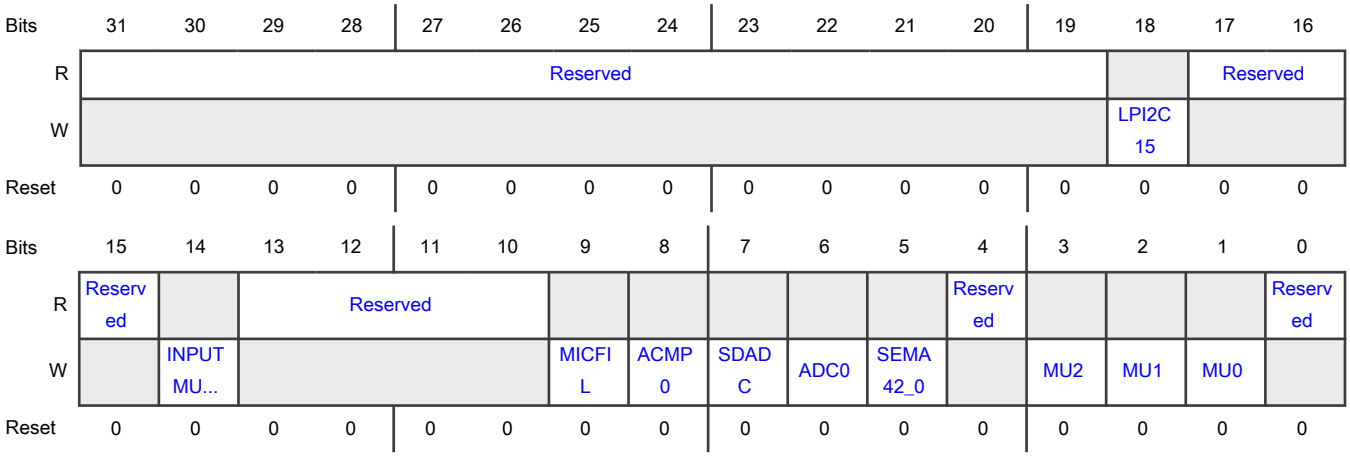
Offset

Register	Offset
PRSTCTL1_CLR	74h

Function

Writing one to a bit position in PRSTCTL1_CLR clears the corresponding position in PRSTCTL1.

Diagram



Fields

Field	Function
31-19 —	Reserved
18 LPI2C15	LPI2C15 Reset Clears LPI2C15 reset. 0b - No effect 1b - Clears reset.
17-15 —	Reserved
14 INPUTMUX1	INPUTMUX1 Reset Clears INPUTMUX1 reset. 0b - No effect 1b - Clears reset.
13-10 —	Reserved
9 MICFIL	MICFIL Reset Clears reset over MICFIL.DATACHn registers. 0b - No effect 1b - Clears reset.
8 ACMP0	ACMP0 Reset Clears ACMP0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
7 SDADC	SDADC Reset Clears SDADC reset. 0b - No effect 1b - Clears reset.
6 ADC0	ADC0 Reset Clears ADC0 reset. 0b - No effect 1b - Clears reset.
5 SEMA42_0	SEMA42_0 Reset Clears SEMA42_0 reset. 0b - No effect 1b - Clears reset.
4 —	Reserved
3 MU2	MU2 Reset Clears MU2 reset. 0b - No effect 1b - Clears reset.
2 MU1	MU1 Reset Clears MU1 reset. 0b - No effect 1b - Clears reset.
1 MU0	MU0 Reset Clears MU0 reset. 0b - No effect 1b - Clears reset.
0 —	Reserved

31.5.5 RSTCTL4 register descriptions

This section includes the memory map and detailed descriptions of registers for Reset Controllers 4, which controls the peripherals in the VDD2_MEDIA power domain on the VDD2 power rail and the VDDN_MEDIA power domain on the VDDN power rail.

31.5.5.1 RSTCTL4 memory map

RSTCTL4 base address: 400A_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Media Domain Peripheral Reset Control 0 (PRSTCTL0)	32	RW	0002_F1FCh
14h	Media Domain (Interface) Peripheral Reset Control 1 (PRSTCTL1)	32	RW	0000_003Fh
40h	Media Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)	32	W	0000_0000h
44h	Media Domain (Interface) Peripheral Reset Control 1 SET (PRSTCTL1_SET)	32	W	0000_0000h
70h	Media Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)	32	W	0000_0000h
74h	Media Domain (Interface) Peripheral Reset Control 1 CLR (PRSTCTL1_CLR)	32	W	0000_0000h

31.5.5.2 Media Domain Peripheral Reset Control 0 (PRSTCTL0)

Offset

Register	Offset
PRSTCTL0	10h

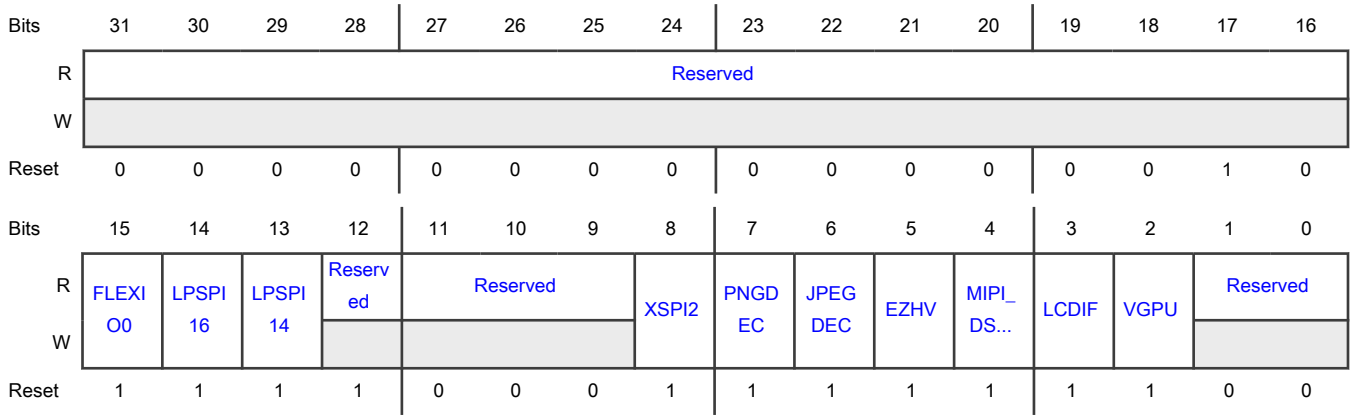
Function

Allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

NOTE

VDD2_MEDIA and VDDN_MEDIA will use same power switch. This way, they are always power ON or power OFF together.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 FLEXIO0	FLEXIO0 Reset Enables FLEXIO0 reset control. 0b - Clears reset. 1b - Sets reset.
14 LPSPI16	LPSPI16 Reset Enables LPSPI16 reset control. 0b - Clears reset. 1b - Sets reset.
13 LPSPI14	LPSPI14 Reset Enables LPSPI14 reset control. 0b - Clears reset. 1b - Sets reset.
12 —	Reserved
11-9 —	Reserved
8 XSPI2	XSPI2 Reset Enables reset control for XSPI2 and MMU2.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Clears reset. 1b - Sets reset.
7 PNGDEC	PNGDEC Reset Enables PNGDEC reset control. 0b - Clears reset. 1b - Sets reset.
6 JPEGDEC	JPEGDEC Reset Enables JPEGDEC reset control. 0b - Clears reset. 1b - Sets reset.
5 EZHV	EZHV Reset Enables EZHV reset control. 0b - Clears reset. 1b - Sets reset.
4 MIPI_DSI_HOST	MIPI_DSI_HOST Reset Enables MIPI_DSI_HOST reset control. 0b - Clears reset. 1b - Sets reset.
3 LCDIF	LCDIF Reset Enables LCDIF reset control. 0b - Clears reset. 1b - Sets reset.
2 VGPU	VGPU Reset Enables VGPU reset control. 0b - Clears reset. 1b - Sets reset.
1-0 —	Reserved

31.5.5.3 Media Domain (Interface) Peripheral Reset Control 1 (PRSTCTL1)

Offset

Register	Offset
PRSTCTL1	14h

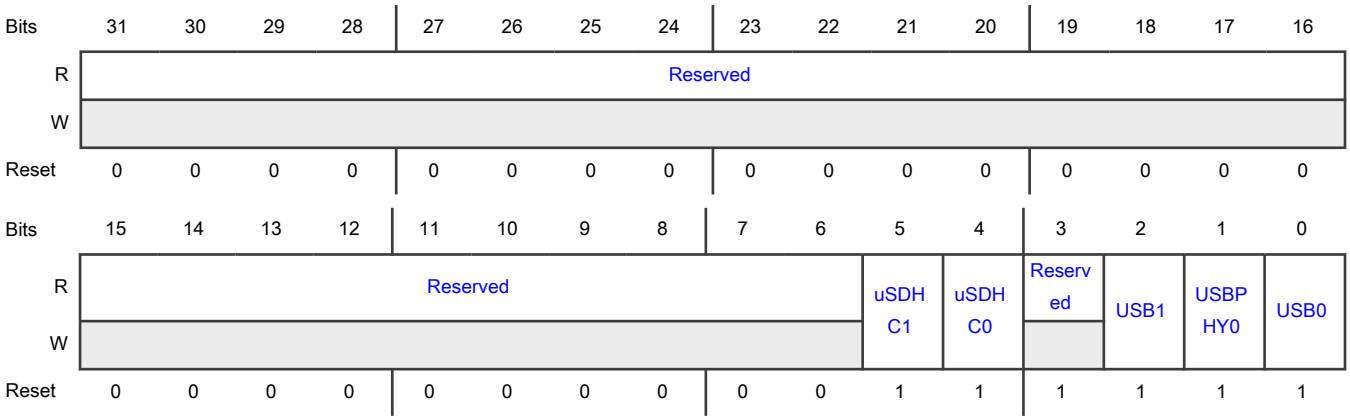
Function

This register allows software to reset specific peripherals. Writing zero to any assigned bit in this register clears the reset and allows the specified peripheral to operate. Writing one asserts the reset. This register will be reset by main cold reset and main warm reset. Because the reset controller module is in the DSR list, the values in this register are retained, even the reset controller module is waken up from the DSR mode, except for wakeup by RESET PAD.

NOTE

VDD2_MEDIA and VDDN_MEDIA will use same power switch. This way, they are always ON/OFF together.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Reset Enables uSDHC1 reset control. 0b - Clears reset. 1b - Sets reset.
4 uSDHC0	uSDHC0 Reset Enables uSDHC0 reset control. 0b - Clears reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Sets reset.
3 —	Reserved
2 USB1	USB1 Reset Enables USB1 reset control. 0b - Clears reset. 1b - Sets reset.
1 USBPHY0	USBPHY0 Reset Enables USBPHY0 reset control. 0b - Clears reset. 1b - Sets reset.
0 USB0	USB0 Reset Enables USB0 reset control. 0b - Clears reset. 1b - Sets reset.

31.5.5.4 Media Domain Peripheral Reset Control 0 SET (PRSTCTL0_SET)

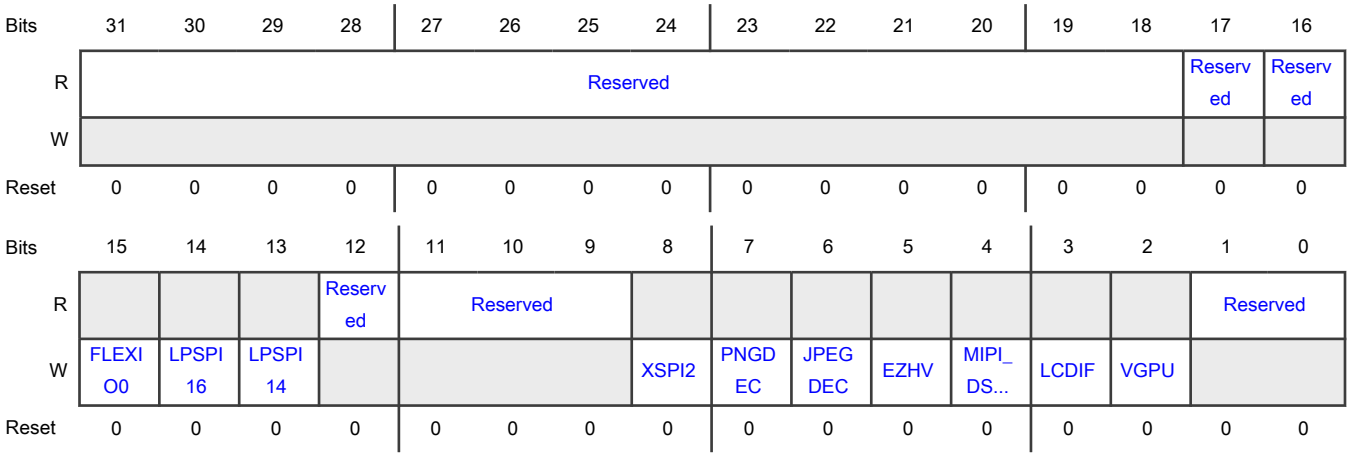
Offset

Register	Offset
PRSTCTL0_SET	40h

Function

Writing one to a bit position in PRSTCTL0_SET sets the corresponding position in PRSTCTL0.

Diagram



Fields

Field	Function
31-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 FLEXIO0	FLEXIO0 Reset Sets FLEXIO0 reset. 0b - No effect 1b - Sets reset.
14 LPSPI16	LPSPI16 Reset Sets LPSPI16 reset. 0b - No effect 1b - Sets reset.
13 LPSPI14	LPSPI14 Reset Sets LPSPI14 reset. 0b - No effect 1b - Sets reset.
12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-9 —	Reserved
8 XSPI2	XSPI2 Reset Sets reset for XSPI2 and MMU2. 0b - No effect 1b - Sets reset.
7 PNGDEC	PNGDEC Reset Sets PNGDEC reset. 0b - No effect 1b - Sets reset.
6 JPEGDEC	JPEGDEC Reset Sets JPEGDEC reset. 0b - No effect 1b - Sets reset.
5 EZHV	EZHV Reset Sets EZHV reset. 0b - No effect 1b - Sets reset.
4 MIPI_DSI_HOST	MIPI_DSI_HOST Reset Sets MIPI_DSI_HOST reset. 0b - No effect 1b - Sets reset.
3 LCDIF	LCDIF Reset Sets LCDIF reset. 0b - No effect 1b - Sets reset.
2 VGPU	VGPU Reset Sets VGPU reset. 0b - No effect 1b - Sets reset.
1-0	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	

31.5.5.5 Media Domain (Interface) Peripheral Reset Control 1 SET (PRSTCTL1_SET)

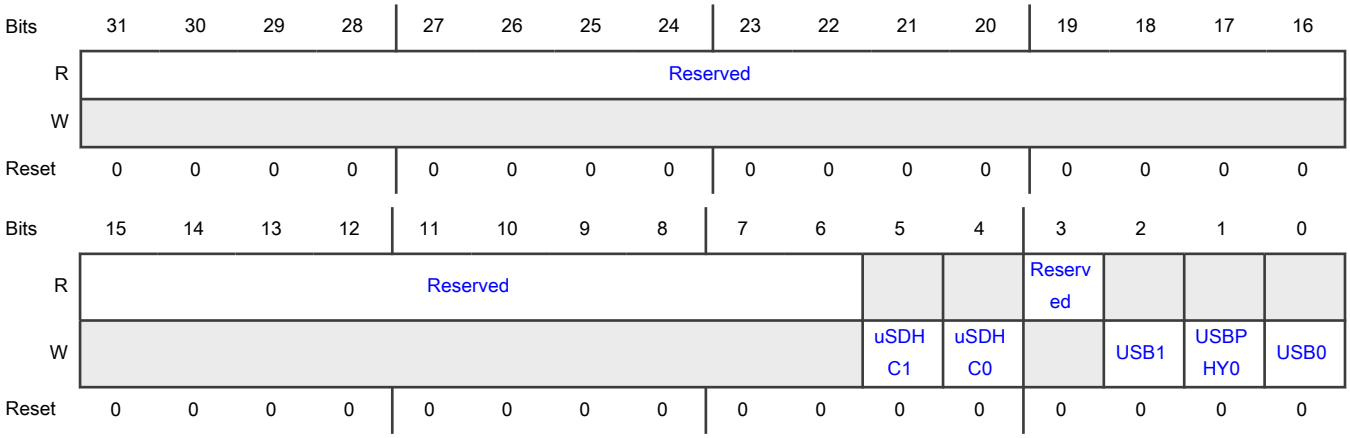
Offset

Register	Offset
PRSTCTL1_SET	44h

Function

Writing one to a bit position in PRSTCTL1_SET sets the corresponding position in PRSTCTL1.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Reset Sets uSDHC1 reset. 0b - No effect 1b - Sets reset.
4 uSDHC0	uSDHC0 Reset Sets uSDHC0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Sets reset.
3 —	Reserved
2 USB1	USB1 Reset Sets USB1 reset. 0b - No effect 1b - Sets reset.
1 USBPHY0	USBPHY0 Reset Sets USBPHY0 reset. 0b - No effect 1b - Sets reset.
0 USB0	USB0 Reset Sets USB0 reset. 0b - No effect 1b - Sets reset.

31.5.5.6 Media Domain Peripheral Reset Control 0 CLR (PRSTCTL0_CLR)

Offset

Register	Offset
PRSTCTL0_CLR	70h

Function

Writing one to a bit position in PRSTCTL0_CLR clears the corresponding position in PRSTCTL0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R				Reserved	Reserved										Reserved	
W	FLEXIO0	LPSPI16	LPSPI14					XSPI2	PNGDEC	JPEGDEC	EZHV	MIPI_DS...	LCDIF	VGPU		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 FLEXIO0	FLEXIO0 Reset Clears FLEXIO0 reset. 0b - No effect 1b - Clears reset.
14 LPSPI16	LPSPI16 Reset Clears LPSPI16 reset. 0b - No effect 1b - Clears reset.
13 LPSPI14	LPSPI14 Reset Clears LPSPI14 reset. 0b - No effect 1b - Clears reset.
12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-9 —	Reserved
8 XSPI2	XSPI2 Reset Clears XSPI2 and MMU2 reset. 0b - No effect 1b - Clears reset.
7 PNGDEC	PNGDEC Reset Clears PNGDEC reset. 0b - No effect 1b - Clears reset.
6 JPEGDEC	JPEGDEC Reset Clears JPEGDEC reset. 0b - No effect 1b - Clears reset.
5 EZHV	EZHV Reset Clears EZHV reset. 0b - No effect 1b - Clears reset.
4 MIPI_DSI_HOST	MIPI_DSI_HOST Reset Clears MIPI_DSI_HOST reset. 0b - No effect 1b - Clears reset.
3 LCDIF	LCDIF Reset Clears LCDIF reset. 0b - No effect 1b - Clears reset.
2 VGPU	VGPU Reset Clears VGPU reset. 0b - No effect 1b - Clears reset.
1-0	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	

31.5.5.7 Media Domain (Interface) Peripheral Reset Control 1 CLR (PRSTCTL1_CLR)

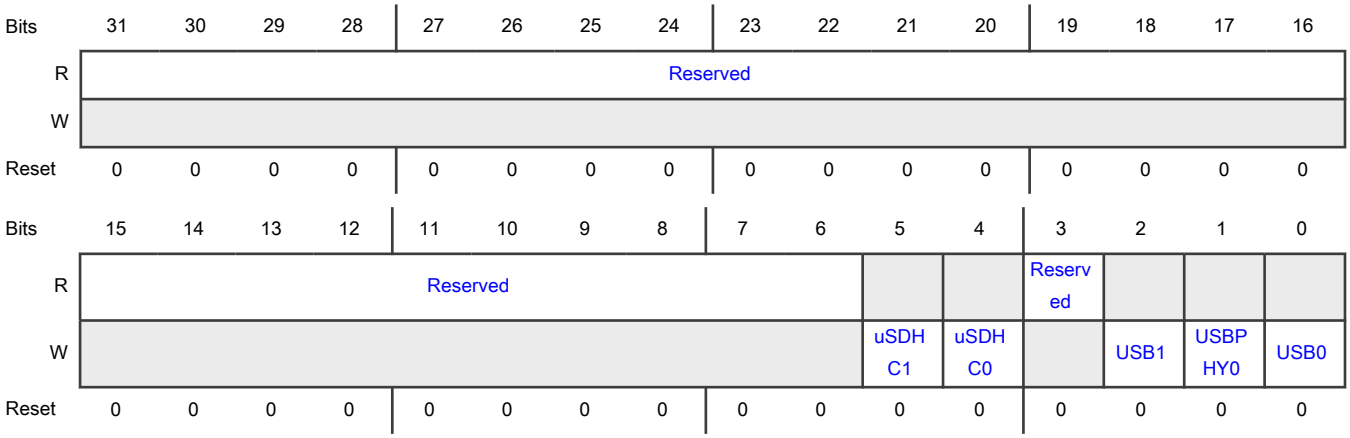
Offset

Register	Offset
PRSTCTL1_CLR	74h

Function

Writing one to a bit position in PRSTCTL1_CLR clears the corresponding position in PRSTCTL1.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 uSDHC1	uSDHC1 Reset Clears uSDHC1 reset. 0b - No effect 1b - Clears reset.
4 uSDHC0	uSDHC0 Reset Clears uSDHC0 reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears reset.
3 —	Reserved
2 USB1	USB1 Reset Clears USB1 reset. 0b - No effect 1b - Clears reset.
1 USBPHY0	USBPHY0 Reset Clears USBPHY0 reset. 0b - No effect 1b - Clears reset.
0 USB0	USB0 Reset Clears USB0 reset. 0b - No effect 1b - Clears reset.

Chapter 32
FRO Clock Generator (FRO)

32.1 Chip-specific FRO information

Table 263. Reference links to related information

Topic	Related module	Reference
Full description	FRO	FRO
Clocking		Clock overview Clock controller
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Sleep Controller	SLEEPCON	SLEEPCON

32.1.1 Module instances

This device contains three instances of the FRO module, FRO0, FRO1, and FRO2.

Table 264. FRO instances

	FRO0	FRO1	FRO2
Located	VDD2_COM	VDD2_COM	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
FRO_CLK should be configured (MHz)	150-325	192	150-325

32.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

32.1.3 FRO_PD

FRO_PD signal is from SLEEPCON. It works together with CSR[FROEN] to control the power on (enable) and power down (disable) of the FRO module.

You can power down (disable) the FRO module by:

- Configure FRO_PD as power down in SLEEPCON
- and configure CSR[FROEN] as disable in FRO

You can power on the FRO module by:

- Configure FRO_PD as power on in SLEEPCON

- or configure CSR[FROEN] as enable in FRO

The following table shows the register fields in SLEEPCON to control the FRO_PD.

Table 265. FRO instances

	FRO_PD control field in SLEEPCON
FRO0	• SLEEPCON0_SLEEPCFG[FRO0_PD] • SLEEPCON0_RUNCFG[FRO0_PD] • SLEEPCON0_RUNCFG_SET[FRO0_PD] • SLEEPCON0_RUNCFG_CLR[FRO0_PD]
FRO1	• SLEEPCON0_SLEEPCFG[FRO1_PD] • SLEEPCON0_RUNCFG[FRO1_PD] • SLEEPCON0_RUNCFG_SET[FRO1_PD] • SLEEPCON0_RUNCFG_CLR[FRO1_PD]
FRO2	• SLEEPCON0_SLEEPCFG[FRO2_PD] • SLEEPCON0_RUNCFG[FRO2_PD] • SLEEPCON0_RUNCFG_SET[FRO2_PD] • SLEEPCON0_RUNCFG_CLR[FRO2_PD] • SLEEPCON1_SLEEPCFG[FRO2_PD] • SLEEPCON1_RUNCFG[FRO2_PD] • SLEEPCON1_RUNCFG_SET[FRO2_PD] • SLEEPCON1_RUNCFG_CLR[FRO2_PD]

32.2 Overview

FRO provides a trimmable system clock, which consists of an internal free-running oscillator (FRO_INT) and an autotuner (FRO_TUNER).

32.2.1 Block diagram

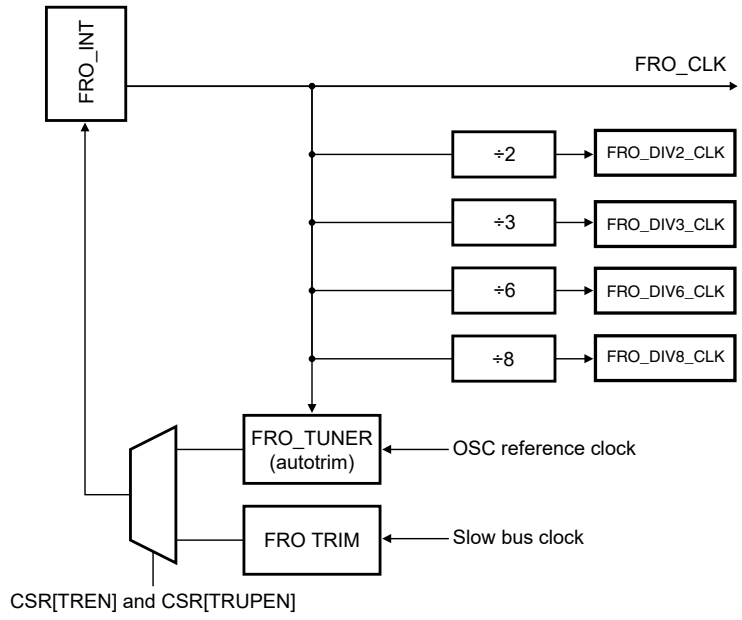


Figure 106. Block diagram

32.2.2 Features

- Trimmable FRO output
- Programmable FRO post divider
- Several interrupt requests
- Provides tune-in logic
- Fast startup option

32.3 Functional description

FRO generates system clock outputs that are based on a trimmable FRO clock source. It operates in the following modes:

- Open Loop (default): In this mode, the trim value (configured either via metal-option or a programmable trim value written to FRO Trim (FROTRIM)) specifies the FRO output clock frequency. See Modes for settings related to this mode.
- Closed Loop: In this mode, FRO_TUNER provides the option for a built-in hardware to adjust the FRO output clock automatically until the output clock locks to a target frequency that you configure in FRO Expected Trim Count (TEXPCNT). FRO_TUNER enters this mode when both CSR[TREN] and CSR[TRUPEN] are 1. See Modes for settings related to this mode.

  The autotuner provides CSR[TRIM_LOCK] to indicate whether the target frequency has been achieved. It continues to monitor the FRO clock source to detect a loss of the target lock frequency when the FRO output clock source deviates from the allowable range specified that TEXPCNT[TEXPCNT_RANGE] specifies. To accomplish this autotuning process, FRO_TUNER uses an external OSC clock source or reference clock to generate an FRO sample window that CNFG1[RFCLKCNT] controls. FRO_TUNER also provides a reference clock divider to support the external reference clocks of multiple chips. The output of this divider is what FRO_TUNER uses to sample the FRO output clock source. You program the correct value in CNFG1[REFDIV] to specify the divide ratio.

32.3.1 Modes

Table 266. Mode settings

Mode	Entry	Exit	Description
Off	POR disabled	POR exit and disabled deassertion	FRO is off and disabled.
Open Loop (on)	CSR[TREN] = 0 CSR[TRUPEN] = 0	CSR[TREN] = 1 CSR[TRUPEN] = 1	FRO runs in Open Loop mode. FRO Trim (FROTRIM) controls the FRO output frequency.
Closed Loop (on)	CSR[TREN] = 1 CSR[TRUPEN] = 1	CSR[TREN] = 0 CSR[TRUPEN] = 0 Warm reset POR	FRO runs in Closed Loop mode. The FRO output frequency is automatically trimmed to the target frequency that you configure by using FRO Expected Trim Count (TEXPCNT) with the use of an accurate clock source.

32.3.2 Interrupts

FRO includes several sources for interrupt requests. The following table summarizes these sources.

Each of these sources includes a mask control field (such as CNFG1[LOL_ERR_IE], CNFG1[TUNE_ERR_IE], and CNFG1[TRUPREQ_IE]).

Table 267. Interrupts

Flag	Functionality	Can generate interrupt?
CSR[LOL_ERR]	Indicates whether FRO is running in Closed Loop mode and the target frequency has deviated more than the expected frequency range after a lock condition is detected.	Yes, if CNFG1[LOL_ERR_IE] = 1
CSR[TUNE_ERR]	Indicates whether FRO is running in Closed Loop mode and has saturated FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM] without achieving the target frequency that you configure in FRO Expected Trim Count (TEXPCNT).	Yes, if CNFG1[TUNE_ERR_IE] = 1
CSR[TRUPREQ]	When configured with CSR[TREN] = 1 and CSR[TRUPEN] = 0, this flag asserts when FRO runs outside the expected frequency range that you configure in TEXPCNT.	Yes, if CNFG1[TRUPREQ_IE] = 1

32.3.3 Clocking

Table 268. Clocking

Clock	Type	Description
Slow bus clock	Input	FRO R/W clock source
OSC reference clock	Input	Used to autotune FRO to the target frequency in Closed Loop mode
FRO_CLK	Output	Undivided FRO trimmed clock (see the chip-specific FRO information for the maximum trim frequency)
FRO_DIV2_CLK	Output	FRO_DIV2_CLK = FRO_CLK ÷ 2

Table continues on the next page...

Table 268. Clocking (continued)

Clock	Type	Description
FRO_DIV3_CLK	Output	FRO_DIV3_CLK = FRO_CLK ÷ 3
FRO_DIV6_CLK	Output	FRO_DIV6_CLK = FRO_CLK ÷ 6
FRO_DIV8_CLK	Output	FRO_DIV8_CLK = FRO_CLK ÷ 8

32.4 External signals

Table 269. External Signals

Signal	Description	Direction
FRO_PD	Signal to power down FRO. This signal can be used to control whether FRO is on or off by default. FRO_PD works together with CSR[FROEN] to determine whether FRO is enabled or disabled. Either FRO_PD or CSR[FROEN] can enable FRO. However, FRO will only be powered down when both FRO_PD and CSR[FROEN] are configured to disable FRO. **NOTE** See the chip-specific information for more details about this signal.	Input

32.5 Initialization

FRO is enabled either externally or by writing 1 to CSR[FROEN]. It supports:

- Open Loop mode (default mode of operation): In this mode, you can adjust the FRO output by writing to FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM].
- Closed Loop mode: In this mode, FRO automatically adjusts the FRO frequency using an external reference clock source.

32.5.1 Open Loop mode

FOPT trim bits initially configure the FRO output frequency, which can be adjusted after exiting chip reset via writes to FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM]. To configure FRO in Open Loop mode:

1. Write 0 to CSR[TREN] (default setting).
2. Write 0 to CSR[TRUPEN].
3. Configure FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM] to the desired frequency setting.

   NOTE

   See FRO Trim (FROTRIM) for more information about how to determine the FRO trim values.

4. Write 1 to CSR[FROEN] to enable FRO (when FRO is not on or enabled by default).

32.5.2 Closed Loop (Autotuner) mode

In Closed Loop mode (CSR[TREN] and CSR[TRUPEN]), FRO automatically adjusts the FRO output frequency by sampling the FRO output clock using an external reference clock, which can be further divided using CNFG1[REFDIV]. The divided reference clock is used to determine how long the FRO clock is sampled. You can configure the sample window using CNFG1[RFCLKCNT]. At the end of the FRO sampled window, a captured FRO clock cycle is compared to the expected count

stored in TEXPCNT[TEXPCNT]. Based on the output comparison result, the FRO autotuner adjusts FRO trim values and feeds them back to FRO to adjust the output frequency up or down.

Initially, the FRO autotuner starts with the trim values stored in FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM]. If CSR[COARSEN] = 1, the FRO autotuner first adjusts the COARSE_TRIM bits, followed by the FINE_TRIM bits. This produces the fastest lock settling but generates maximum frequency disturbance, hence it is not recommended for most applications. If CSR[COARSEN] = 0, the autotuner starts by adjusting the FINE_TRIM bits and only adjusts the COARSE_TRIM bits if saturation of FINE_TRIM is detected. This gives the least disturbance and is the generally recommended configuration to use.

CSR[TRIM_LOCK] sets when the FRO autotuner detects the FRO output frequency to be within TEXPCNT[TEXPCNT] ± TEXPCNT[TEXPCNT_RANGE]. When the autotuner is in locked state (CSR[TRIM_LOCK] = 1), it continues to monitor and adjust the FRO output frequency, and CSR[LOL_ERR] will set if the FRO output frequency drifts outside the TEXPCNT[TEXPCNT] ± TEXPCNT[TEXPCNT_RANGE] range. If the autotuner saturates all trim value options without reaching a locking state condition, then CSR[TUNE_ERR] will set. You can configure both CSR[LOL_ERR] and CSR[TUNE_ERR] to generate an output interrupt request by writing 1 to CNFG1[LOL_ERR_IE] and CNFG1[TUNE_ERR_IE], respectively.

To configure FRO in Closed Loop mode:

1. Write 0 to CSR[TREN] and CSR[TRUPEN].
2. Configure the FRO trim values (FROTRIM[COARSE_TRIM] and FROTRIM[FINE_TRIM]) to correct the start frequency, if applicable.

   NOTE

   Gating off output clocks is recommended when changing the FRO frequency. You can do this by disabling each of the clock gates in CSR[CLKGATE]. Writing 0 to CSR[CLKGATE] disables the output clock.

3. Configure CNFG1[REFDIV] to generate the desired FRO reference frequency.
4. Configure CNFG1[RFCLKCNT] to be large enough to sample the FRO for a minimum of 4000 FRO clock cycles.
5. Configure TEXPCNT[TEXPCNT] to the expected FRO count based on the FRO sample size configured in 4.
6. Configure TEXPCNT[TEXPCNT_RANGE] to the desired lock setting tolerance.
7. Configure TRIM*n*_DELAY (see CNFG2[TRIM1_DELAY] and CNFG2[TRIM2_DELAY]) to insert the minimum delay that is decided based on the divided reference clock.
8. Write 1 to the desired FRO monitor interrupt options, including CNFG1[LOL_ERR_IE] and CNFG1[TUNE_ERR_IE].
9. To enable FRO autotuner:
   a. Write 1 to CSR[FROEN] to enable FRO if it is not already enabled.
   b. Write 1 to CSR[COARSEN] if autotuning of the coarse trim is required.

      NOTE

      This could generate a faster settling to the final target frequency but generates coarser frequency adjustments.

   c. Write 1 to CSR[TREN] and CSR[TRUPEN].

32.6 FRO register descriptions

32.6.1 FRO memory map

FRO0 base address: 4000_1100h

FRO1 base address: 4000_1000h

FRO2 base address: 4006_1100h

Offset	Register	Width (In bits)	Access	Reset value
200h	Control Status (CSR)	32	RW	0000_1F00h
204h	Control Status (CSR_SET)	32	RW	0000_1F00h
208h	Control Status (CSR_CLR)	32	RW	0000_1F00h
20Ch	Control Status (CSR_TOG)	32	RW	0000_1F00h
210h	Trim Configuration 1 (CNFG1)	32	RW	0000_0000h
214h	Trim Configuration 1 (CNFG1_SET)	32	RW	0000_0000h
218h	Trim Configuration 1 (CNFG1_CLR)	32	RW	0000_0000h
21Ch	Trim Configuration 1 (CNFG1_TOG)	32	RW	0000_0000h
220h	Trim Configuration 2 (CNFG2)	32	RW	0000_0000h
224h	Trim Configuration 2 (CNFG2_SET)	32	RW	0000_0000h
228h	Trim Configuration 2 (CNFG2_CLR)	32	RW	0000_0000h
22Ch	Trim Configuration 2 (CNFG2_TOG)	32	RW	0000_0000h
240h	FRO Trim (FROTRIM)	32	RW	0000_0000h
244h	FRO Trim (FROTRIM_SET)	32	RW	0000_0000h
248h	FRO Trim (FROTRIM_CLR)	32	RW	0000_0000h
24Ch	FRO Trim (FROTRIM_TOG)	32	RW	0000_0000h
250h	FRO Expected Trim Count (TEXPCNT)	32	RW	0000_0000h
254h	FRO Expected Trim Count (TEXPCNT_SET)	32	RW	0000_0000h
258h	FRO Expected Trim Count (TEXPCNT_CLR)	32	RW	0000_0000h
25Ch	FRO Expected Trim Count (TEXPCNT_TOG)	32	RW	0000_0000h
260h	FRO Auto Tune Trim (AUTOTRIM)	32	R	0000_0000h
264h	FRO Auto Tune Trim (AUTOTRIM_SET)	32	R	0000_0000h
268h	FRO Auto Tune Trim (AUTOTRIM_CLR)	32	R	0000_0000h
26Ch	FRO Auto Tune Trim (AUTOTRIM_TOG)	32	R	0000_0000h
270h	FRO Trim Count (TRIMCNT)	32	R	0000_0000h
274h	FRO Trim Count (TRIMCNT_SET)	32	R	0000_0000h
278h	FRO Trim Count (TRIMCNT_CLR)	32	R	0000_0000h
27Ch	FRO Trim Count (TRIMCNT_TOG)	32	R	0000_0000h

32.6.2 Control Status (CSR)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CSR	200h	Control Status
CSR_SET	204h	Writing 1 to a bit in this register ensures that the corresponding bit in CSR is 1
CSR_CLR	208h	Writing 1 to a bit in this register ensures that the corresponding bit in CSR is 0
CSR_TOG	20Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CSR

Function

Configures FRO in Closed Loop mode or Open Loop mode and also controls individual clock gating.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0	TUNEONC...	TRIM_LO...	0					TRUPREQ	TUNE_ERR	LOL_ERR
W							W1C							W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	CLKGATE					TUNEONCE	COARSEN	TRUPEN	TREN	0	0	0	FROEN
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved
26 —	Reserved
25 TUNEONCE_DONE	FRO Tune Once Done Flag Indicates when the FRO tune once is complete and ready to trigger another tune once sequence. When this flag is cleared, it triggers another tune once sequence. **NOTE** You can only clear this flag by writing 1 to the corresponding bit in CSR_CLR. 0b - Not complete 1b - Complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 TRIM_LOCK	FRO Trim Lock Flag Indicates when autotrimming is complete and locked to the target trimmed frequency that you configure by using FRO Expected Trim Count (TEXPCNT), if FRO is configured in Closed Loop mode (TREN = 1 and TRUPEN = 1). **NOTE** • If the trimmed frequency drifts outside the expected range determined by TEXPCNT[TEXPCNT_RANGE], this flag is cleared automatically. Under this scenario, LOL_ERR sets to indicate an unexpected loss-of-lock condition. • If either TREN or TRUPEN is cleared, this flag also clears automatically, and LOL_ERR is not set. 0b - Not locked 1b - Locked
23-19 —	Reserved
18 TRUPREQ	Trim Update Request Flag Indicates whether a trim update request is detected and FRO is running outside the expected frequency range that you configure by using FRO Expected Trim Count (TEXPCNT). **NOTE** • This flag is set only when you configure FRO with TREN = 1 and TRUPEN = 0. • An interrupt is generated when this field is 1 and TRUPREQ_IE is 1. **NOTE** You can only clear this flag by writing 1 to the corresponding bit in CSR_CLR. 0b - Not detected 1b - Detected
17 TUNE_ERR	Tune Error Flag Indicates whether an autotune error is detected and FRO has saturated high- and low-byte trim bits without reaching the target frequency in Closed Loop mode (TREN = 1 and TRUPEN = 1). **NOTE** You can only clear this flag by writing 1 to the corresponding bit in CSR_CLR. 0b - Not detected 1b - Detected
16 LOL_ERR	Loss-of-Lock Error Flag Indicates whether FRO has detected a loss-of-trimmed-lock condition when running in Closed Loop mode (TREN = 1 and TRUPEN = 1).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE You can only clear this flag by writing 1 to the corresponding bit in CSR_CLR. 0b - Not detected 1b - Detected
15-14 —	Reserved
13 —	Reserved
12-8 CLKGATE	FRO Clock Enable Enables the FRO clocks. This field controls the FRO output clocks and allows the option to gate them off. 0_xxxxb - Disables FRO divider 8 clock 1_xxxxb - Enables FRO divider 8 clock x_0xxxb - Disables FRO divider 6 clock x_1xxxb - Enables FRO divider 6 clock x_x0xxb - Disables FRO divider 3 clock x_x1xxb - Enables FRO divider 3 clock x_xx0xb - Disables FRO divider 2 clock x_xx1xb - Enables FRO divider 2 clock x_xxx0b - Disables FRO divider 1 clock x_xxx1b - Enables FRO divider 1 clock
7 TUNEONCE	Tune Once Control When running in Closed Loop mode (TREN = 1 and TRUPEN = 1) with this field set, the FRO frequency autotuner will pause after tuning once. The autotuner waits until the TUNEONCE_DONE flag is cleared before it tunes again. When using least disturbance (COARSEN = 0), the autotuner can only move one low-byte trim and may not achieve the tuning goal. The tune once feature can also be used with TREN = 1 and TRUPEN = 0. In this configuration, the behavior is almost identical to when running in Closed Loop mode, except the FRO trim will always be controlled by FRO Trim (FROTRIM) (instead of the FRO Auto Tune Trim (AUTOTRIM) calculated by the autotuner). This means the application is responsible for programming FRO Trim (FROTRIM) to the next desired trim value based on the FRO Trim Count (TRIMCNT) result. To avoid disturbing the FRO frequency during tuning, the FRO Trim (FROTRIM) should only be changed after TUNEONCE_DONE is set and before clearing TUNEONCE_DONE to trigger the next tune once sequence. **NOTE** This field must be configured before enabling TREN.

Table continues on the next page...

Table continued from the previous page...

Field	Function
6 COARSEN	Coarse Trim Enable Enables autotrimming of the FRO high-byte trim bits, which have a coarse frequency adjustment when FRO is programmed in Closed Loop mode (TREN = 1 and TRUPEN = 1). Writing 1 to this field allows autotrimming of the FRO high-byte trim bits. When this field is 0, only the low-bytes or fine trim bits are adjusted automatically. **NOTE** When autotrim updating is enabled (TRUPEN = 1), it is recommended to keep this field as 0 to minimize frequency disturbance and risk of overshoot. 0b - Disable 1b - Enable
5 TRUPEN	FRO Autotrim Update Enable Enables FRO to run in Closed Loop mode when TREN = 1, trimming the FRO to an external clock source. If this field is 0, FRO is configured in Open Loop mode. Configure the FRO frequency by using FRO Trim (FROTRIM). If this field is 1, FRO autotrimming is enabled. Configure the target frequency by using FRO Expected Trim Count (TEXPCNT). **NOTE** You must write 1 to TREN before enabling this field. 0b - Disable 1b - Enable
4 TREN	FRO Trim Enable Enables the FRO frequency to be compared with an expected count that you configure in FRO Trim Count (TRIMCNT) by using an external reference clock. If this field is 0, FRO is configured in Open Loop mode. Configure the FRO frequency by using FRO Trim (FROTRIM). If this field is 1, FRO autotrimming is enabled. FRO is trimmed to an external reference clock. Configure the target frequency by using FRO Expected Trim Count (TEXPCNT). TRUPREQ is set when the FRO frequency is outside TEXPCNT[TEXPCNT_RANGE] and TRUPEN is disabled. **NOTE** • Writing 1 to this field and writing 0 to TRUPEN allows the FRO frequency to be compared to the expected frequency that you configure in FRO Trim Count (TRIMCNT), but keeps FRO running in Open Loop mode, where its output frequency is based on FRO Trim (FROTRIM) configurations. • To configure FRO in Closed Loop mode and automatically adjust the FRO output clock based on the target frequency that you configure by using FRO Trim Count (TRIMCNT), you must enable this field and TRUPEN.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 FROEN	FRO Enable Enables FRO. **NOTE** Depending on the chip's configuration, FRO can be on by default. When FRO is off by default, writing 1 to this field enables FRO. FRO will only be powered down when both FRO_PD and this field are configured to disable FRO. See External signals. 0b - Disable 1b - Enable

32.6.3 Trim Configuration 1 (CNFG1)

Offset

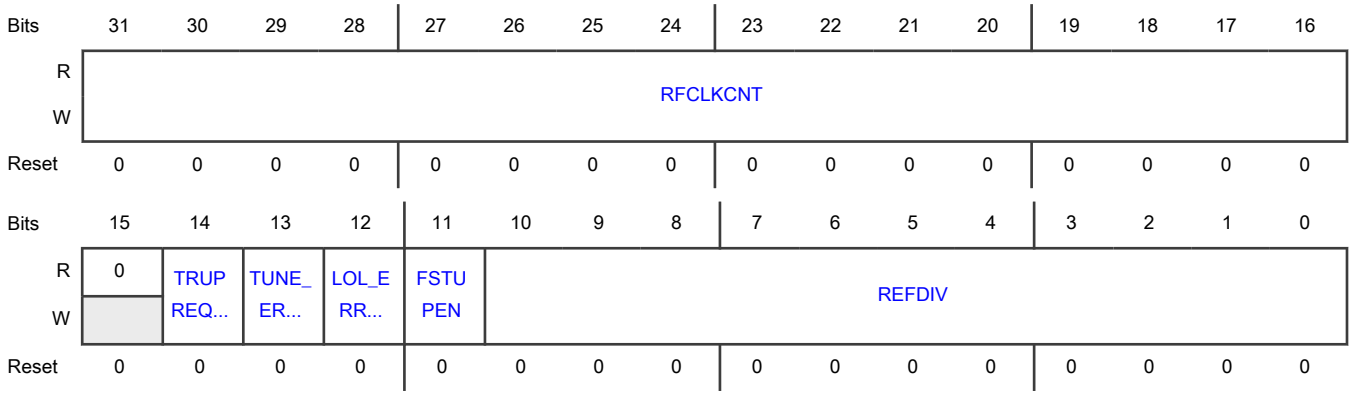
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CNFG1	210h	Trim Configuration 1
CNFG1_SET	214h	Writing 1 to a bit in this register ensures that the corresponding bit in CNFG1 is 1
CNFG1_CLR	218h	Writing 1 to a bit in this register ensures that the corresponding bit in CNFG1 is 0
CNFG1_TOG	21Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CNFG1

Function

Controls the FRO fast startup enable, FRO reference clock divider, FRO sample window when running in Autotuner or Closed Loop mode, and FRO interrupt enable.

Diagram



Fields

Field	Function
31-16 RFCLKCNT	Reference Clock Counter Specifies the number of reference clocks where FRO is sampled when running in Closed Loop mode (CSR[TREN] = 1). This window should be large enough to sample the FRO for a minimum of 4000 FRO clock cycles. Use the following formula to calculate the minimum RFCLKCNT to use: RFCLKCNT ≥ 4000 × divided reference frequency ÷ FRO target frequency **NOTE** Count value of 0 is invalid and not supported.
15 —	Reserved
14 TRUPREQ_IE	Trim Update Request Interrupt Enable Enables the trim update request to generate an interrupt when FRO is running in Closed Loop mode (CSR[TREN] = 1 and CSR[TRUPEN] = 1) and CSR[TUNE_ERR] is set. 0b - Disable 1b - Enable
13 TUNE_ERR_IE	Tune Error Interrupt Enable Enables the tune error to generate an interrupt when FRO is running in Closed Loop mode (CSR[TREN] = 1 and CSR[TRUPEN] = 1) and CSR[TUNE_ERR] is set. 0b - Disable 1b - Enable
12 LOL_ERR_IE	Loss-of-Lock Error Interrupt Enable Enables a loss-of-lock error to generate an interrupt when FRO is running in Closed Loop mode (CSR[TREN] = 1 and CSR[TRUPEN] = 1) and CSR[LOL_ERR] = 1. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
11 FSTUPEN	FRO Fast Startup Enable Enables FRO fast startup. During the fast startup period, the FRO will output a divide-by-2 clock until the frequency is within 5% of the target. Then the FRO will automatically switch to the undivided clock output. The amplitude and duty cycle are stable while the frequency ramps to its final value. 0b - Disable 1b - Enable
10-0 REFDIV	OSC Reference Clock Divider Specifies the divide ratio to divide the OSC reference clock for FRO trimming. **NOTE** This field is an N-divider supporting a divide ratio of 1 (REFDIV = 0h) to a divide ratio of 2048 (REFDIV = 7FFh).

32.6.4 Trim Configuration 2 (CNFG2)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CNFG2	220h	Trim Configuration 2
CNFG2_SET	224h	Writing 1 to a bit in this register ensures that the corresponding bit in CNFG2 is 1
CNFG2_CLR	228h	Writing 1 to a bit in this register ensures that the corresponding bit in CNFG2 is 0
CNFG2_TOG	22Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CNFG2

Function

Controls the delay before FRO is sampled after adjusting the FRO trim values in Autotuner mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				TRIM1_DELAY											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TRIM2_DELAY											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 TRIM1_DELAY	Trim 1 Delay Register Specifies the number of divided reference clocks to wait for the FRO frequency to settle after trim values are updated by 1 decimal unit. This delay is applied only when the FRO is running in Closed Loop mode (CSR[TREN] = 1 and CSR[TRUPEN] = 1) and coarse trimming is disabled (CSR[COARSEN] = 0). **NOTE** • A count value of 0 is invalid and not supported. • A minimum trim delay of 5 µs should be used.
15-12 —	Reserved
11-0 TRIM2_DELAY	Trim 2 Delay Register Specifies the number of divided reference clocks to wait for the FRO frequency to settle after trim values are updated by 16 decimal unit steps (when CSR[COARSEN] = 0) or 128 decimal unit steps (when CSR[COARSEN] = 1). This delay is applied only when the FRO is running in Closed Loop mode (CSR[TREN] = 1 and CSR[TRUPEN] = 1). **NOTE** • A count value of 0 is invalid and not supported. • If CSR[COARSEN] = 0, a minimum trim delay of 30 µs should be used. • If CSR[COARSEN] = 1, a minimum trim delay of 150 µs should be used.

32.6.5 FRO Trim (FROTRIM)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
FROTRIM	240h	FRO Trim
FROTRIM_SET	244h	Writing 1 to a bit in this register ensures that the corresponding bit in FROTRIM is 1
FROTRIM_CLR	248h	Writing 1 to a bit in this register ensures that the corresponding bit in FROTRIM is 0
FROTRIM_TOG	24Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in FROTRIM

Function

Configures the FRO trim values when FRO is configured in Open Loop mode.

NOTE

- This register is reset only on POR.
- A larger [COARSE_TRIM:FINE_TRIM] trim value corresponds to a higher FRO output frequency.
- The desired trim can be found by observing the FRO Auto Tune Trim (AUTOTRIM) value after the FRO is locked in Closed Loop Mode OR by using some other frequency measurement method supported on the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										TRIMTEMP					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0[1]	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				COARSE_TRIM				FINE_TRIM							
W																
Reset	0	0	0	0	0[1]	0	0	0	0	0[1]	0	0	0	0	0	0

1. The metal-option value sets the reset values.

Fields

Field	Function
31-22 —	Reserved
21-16 TRIMTEMP	Trim Temperature Specifies the temperature trim.
15-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-7 COARSE_TRIM	Coarse Trim Adjusts the frequency with a resolution of 2%.
6-0 FINE_TRIM	Fine Trim Adjusts the frequency with a resolution of 0.1%.

32.6.6 FRO Expected Trim Count (TEXPCNT)

Offset

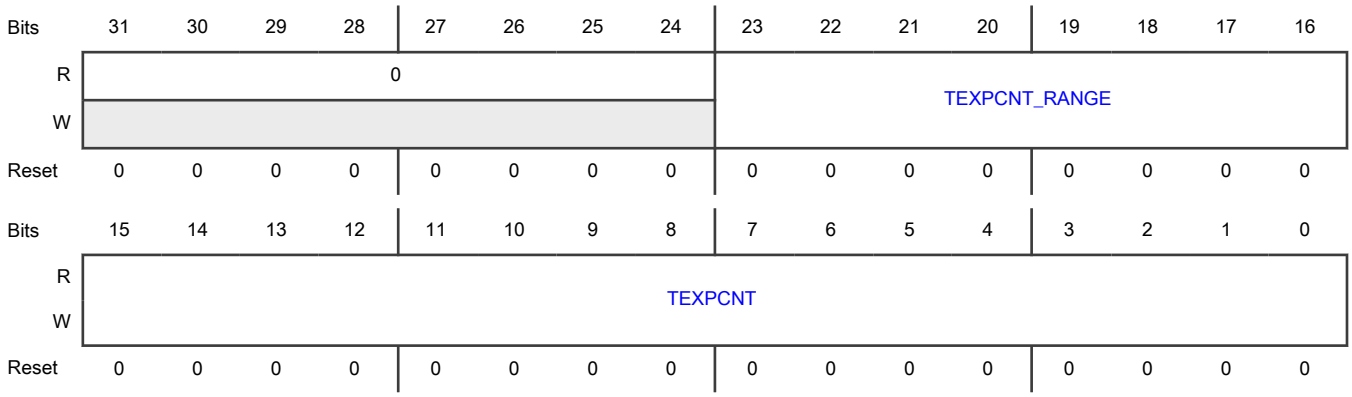
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TEXPCNT	250h	FRO Expected Trim Count
TEXPCNT_SET	254h	Writing 1 to a bit in this register ensures that the corresponding bit in TEXPCNT is 1
TEXPCNT_CLR	258h	Writing 1 to a bit in this register ensures that the corresponding bit in TEXPCNT is 0
TEXPCNT_TOG	25Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TEXPCNT

Function

Configures the target trim FRO frequency count based on the configured CNFG1[RFCLKCNT] value and a known reference clock.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 TEXPCNT_RANGE	Trim Expected Count Range Specifies the + or – counts that the FRO frequency can be off from TEXPCNT to be considered locked. A smaller count corresponds to a lower tolerance for deviation from the target frequency. The percentage of frequency deviation is roughly equal to the ratio of TEXPCNT_RANGE / TEXPCNT.
15-0 TEXPCNT	Trim Expected Count Specifies the expected trim frequency target count when CSR[TREN] = 1. Use the following equation to determine the TEXPCNT for a specific target frequency: TEXPCNT = RFCLKCNT × (FRO target frequency ÷ divided reference frequency)

32.6.7 FRO Auto Tune Trim (AUTOTRIM)

Offset

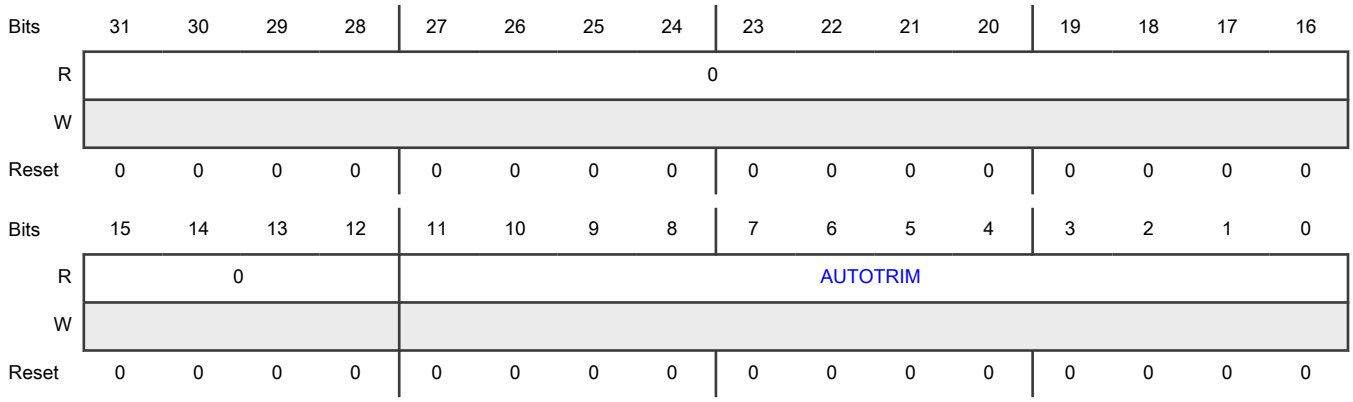
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
AUTOTRIM	260h	FRO Auto Tune Trim
AUTOTRIM_SET	264h	Writing 1 to a bit in this register ensures that the corresponding bit in AUTOTRIM is 1
AUTOTRIM_CLR	268h	Writing 1 to a bit in this register ensures that the corresponding bit in AUTOTRIM is 0
AUTOTRIM_TOG	26Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in AUTOTRIM

Function

Returns the autotune trim value.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-0 AUTOTRIM	Auto Tune Trim Value Stores the autotrim value when FRO is running in Closed Loop mode.

32.6.8 FRO Trim Count (TRIMCNT)

Offset

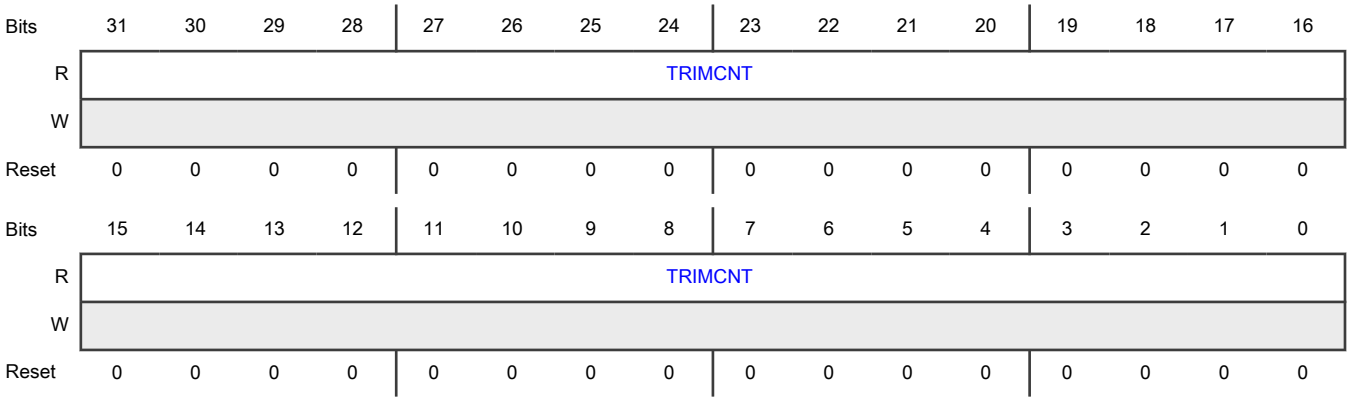
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TRIMCNT	270h	FRO Trim Count
TRIMCNT_SET	274h	Writing 1 to a bit in this register ensures that the corresponding bit in TRIMCNT is 1
TRIMCNT_CLR	278h	Writing 1 to a bit in this register ensures that the corresponding bit in TRIMCNT is 0
TRIMCNT_TOG	27Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TRIMCNT

Function

Stores the count value that FRO last sampled.

Diagram



Fields

Field	Function
31-0 TRIMCNT	Trim Count Stores the last FRO sampled count value. **NOTE** The FRO frequency can be derived from the formula below. Frequency = TRIMCNT × (divided ref clock frequency) ÷ CNFG1[REFCLKCNT]

Chapter 33
Frequency Measurement (FREQME)

33.1 Chip-specific FREQME information

Table 270. Reference links to related information

Topic	Related module	Reference
Full description	FREQME	FREQME
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Input multiplexing	INPUTMUX	INPUTMUX

33.1.1 Module instances

This device has one instance of the FREQME module.

Table 271. FREQME instance

Features	FREQME
Located in	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

33.1.2 Reference clock and target clock

FREQME reference clock and target clock are configured by INPUTMUX0 registers.

Table 272. Reference clock and target clock configuration

Clock	Configured by
Reference clock	INPUTMUX0_FREQME_TRIG0[TRIGIN]
Target clock	INPUTMUX0_FREQME_TRIG1[TRIGIN]

33.2 Overview

FREQME accurately measures the frequency of an on- or off-chip target clock signal using a selectable on-chip reference clock. For example, it can accurately determine the frequency of a low-power oscillator that varies depending on process and temperature.

33.2.1 Block diagram

Figure 107. Block diagram

See the chip-specific FREQME information for the target and reference clock selection options.

33.2.2 Features

- High-accuracy Frequency Measurement mode for on- and off-chip clocks
- Pulse Width Measurement (PWM) mode
- Reference and target clock inputs selectable from among various chip-specific options
- Optional measurement complete interrupt
- Result out-of-range detection with optional interrupt

33.3 Functional description

The following sections describe FREQME functional details.

33.3.1 Frequency Measurement mode

In Frequency Measurement mode (CTL_W[PULSE_MODE] = 0), FREQME counts the number of target clock cycles that occur during a specified number of cycles from a reference clock that has a known frequency. You can then calculate the target clock frequency based on the frequency of the reference clock, the number of reference clock cycles, and the number of target clock cycles (see Equation 1).

The frequency measurement circuit is based on two 31-bit counters—one clocked by the selected reference clock and one by the selected target clock. You can only read the target clock counter. FREQME synchronizes the clocks at the start and end of each count sequence during preparation for the next count sequence. A count sequence consists of incrementing the target clock counter for each target clock pulse that occurs during the time defined by $2^{CTRL_W[REF_SCALE]}$ periods of the reference clock.

After selecting reference and target clocks, you initiate a measurement cycle by writing 1 to CTRL_W[MEASURE_IN_PROGRESS]. You can then poll this same field, which automatically transitions to 0 when the measurement operation completes. Alternatively, the completion of the measurement operation can generate an interrupt.

The measurement cycle terminates when the reference counter equals the value $2^{CTRLSTAT[REFSCALE]}$. If CTRLSTAT[REFSCALE] = 0, the reference counter counts to one and stops. You can use this feature to measure the frequency of a fast target clock using a slow reference clock such as the 32 kHz clock without taking much time for the measurement to complete. The penalty is reduced accuracy in the measurement.

When the counting operation completes, the state of the target counter is loaded into CTRL_R[RESULT], and the CTRL_W[MEASURE_IN_PROGRESS] field is 0. You can then read the value and calculate the target frequency as follows, with the frequencies given in MHz.

$$F_{target} = (CTRL_R[RESULT] - 2) \times F_{reference} \div 2^{CTRLSTAT[REFSCALE]}$$

Equation 1. Calculating the target clock frequency

33.3.1.1 Accuracy

The Frequency Measurement mode can measure the frequency of any on-chip (or off-chip) clock (referred to as the target clock) with a high degree of accuracy by using an on-chip clock of known frequency as a reference clock.

Uncertainty in the reference clock (for example a ±1 % accuracy of the clock) adds to the measurement error of the target clock. In general, though, this additional error is less than the uncertainty of the reference clock.

33.3.2 Pulse Measurement mode

When you write 1 to CTRL_W[PULSE_MODE] and write 1 to CTRL_W[MEASURE_IN_PROGRESS] to start a measurement cycle, FREQME counts target clock pulses while the reference clock is in a specific state (high or low), selected by writing to CTRL_W[PULSE_POL]. See the description of these fields for more details.

33.3.3 Clocking

The clock inputs are as follows:

- Reference clock (REF_CLK_IN). This clock has a known frequency.
- Target clock (TARGET_CLK_IN). This clock frequency can be calculated based on the reference clock.
- Peripheral clock used for register access (PCLK). This clock is used to write and read memory-mapped registers of this module.

NOTE

These clocks can be all synchronous.

See the chip-specific FREQME information for the target and reference clock selection options.

33.3.4 Interrupts

The following interrupts can optionally be generated at the completion of a frequency measurement cycle.

Table 273. Interrupts

Condition	Interrupt enable field	Interrupt status field
Result is ready to read	CTRL_W[RESULT_READY_INT_EN]	CTRLSTAT[RESULT_READY_INT_EN]
Result is greater than MAX[MAX_VALUE]	CTRL_W[GT_MAX_INT_EN]	CTRLSTAT[GT_MAX_INT_EN]
Result is less than MIN[MIN_VALUE]	CTRL_W[LT_MIN_INT_EN]	CTRLSTAT[LT_MIN_INT_EN]

33.4 External signals

Table 274. External signals

Signal	Description	Direction
REF_CLK_IN	Reference clock. On-chip clock with known frequency chosen as reference.	Input
TARGET_CLK_IN	Target clock. On- or off-chip clock to be measured by FREQME.	Input

33.5 Initialization

Initialize FREQME by performing the following steps:

1. Enable the clock that drives FREQME in the chip-level clock control register. This step enables the register interface and the peripheral function clock.
2. Clear the FREQME peripheral reset in the chip-level reset control register.

3. If measuring an external clock, use the Input/Output pin configuration registers to connect the FREQME target clock input (TARGET_CLK_IN) to an external pin.
4. Ensure that the reference and target clocks are enabled.
5. Select the reference and target clocks.

33.6 Memory map and register definition

This section includes the FREQME memory map and detailed descriptions of all registers.

33.6.1 FREQME register descriptions

33.6.1.1 FREQME memory map

FREQME base address: 4002_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (in Read mode) (CTRL_R)	32	R	0000_0000h
0h	Control (in Write mode) (CTRL_W)	32	W	0000_0000h
4h	Control Status (CTRLSTAT)	32	RW	0000_0000h
8h	Minimum (MIN)	32	RW	0000_0000h
Ch	Maximum (MAX)	32	RW	7FFF_FFFFh

33.6.1.2 Control (in Read mode) (CTRL_R)

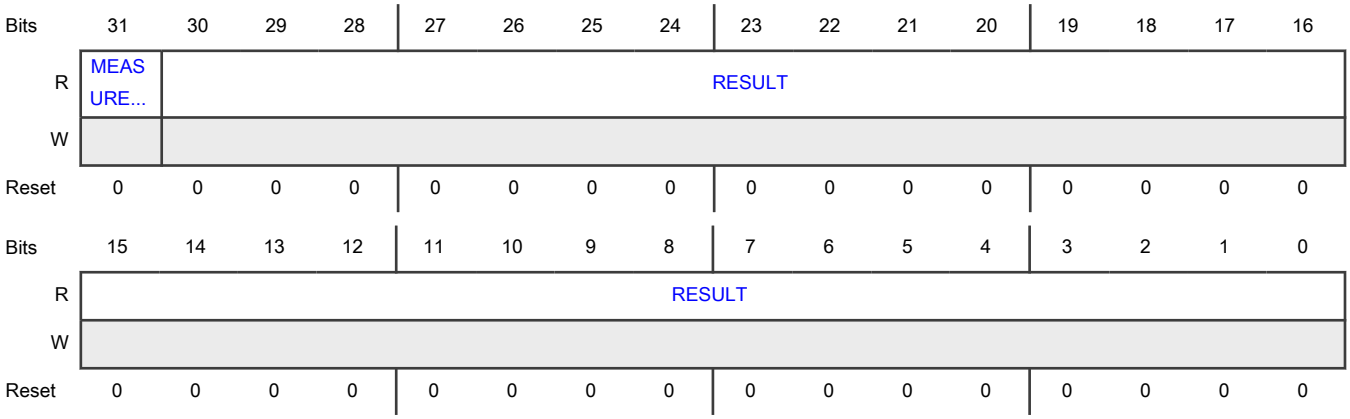
Offset

Register	Offset
CTRL_R	0h

Function

Contains different fields depending on whether you are performing a read or a write. Reading this register provides the measurement results and the status of the measurement cycle.

Diagram



Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement In Progress Indicates whether measurement is in progress or complete. If complete, you can read the result from RESULT. 0b - Complete 1b - In progress
30-0 RESULT	Indicates the measurement result—either the target clock counter value (for Frequency Measurement mode) or pulse width measurement (for Pulse Width Measurement mode). This field is valid only when MEASURE_IN_PROGRESS = 0, In Continuous mode (CTRL_W[CONTINUOUS_MODE_EN] = 1), the value is the result from the most-recently completed measurement.

33.6.1.3 Control (in Write mode) (CTRL_W)

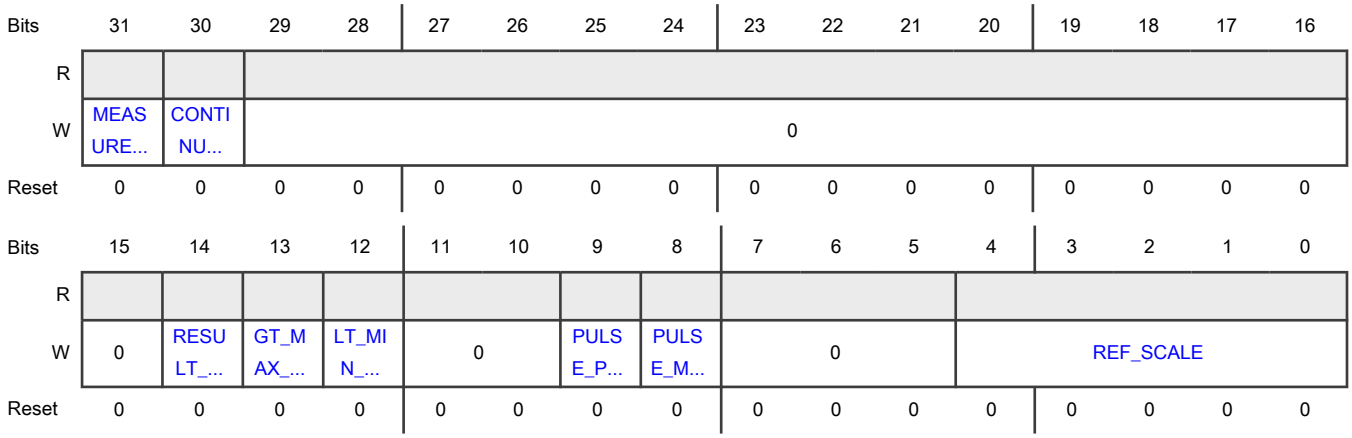
Offset

Register	Offset
CTRL_W	0h

Function

Writes to this register control and configure the measurement cycle.

Diagram



Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement In Progress Initiates a frequency measurement cycle or terminates a measurement cycle that is in progress. Writing 1 to this field initiates a frequency or pulse width measurement process. Hardware automatically writes 0 to the MEASURE_IN_PROGRESS field when the measurement cycle completes. If there is an active measurement in progress, a new measurement starts. Writing 0 to this field forces the termination of any measurement cycle currently in progress and resets CTRL_R[RESULT] or just resets CTRL_R[RESULT] if idle. 0b - Terminates measurement 1b - Initiates measurement
30 CONTINUOUS_MODE_EN	Continuous Mode Enable When you write 1 to both MEASURE_IN_PROGRESS and this field, measurement is performed continuously. The result for the most-recently completed measurement is available in CTRL_R[RESULT]. When the result is out of range (that is, CTRLSTAT[GT_MAX_STAT] = 1 or CTRLSTAT[LT_MIN_STAT] = 1), this field automatically transitions to 0 and the continuous measurement is suspended. After clearing both status bits, you must write 1 to both MEASURE_IN_PROGRESS and this field restart measurement in continuous mode. 0b - Disable 1b - Enable
29-15 —	Reserved
14 RESULT_READY_INT_EN	Result Ready Interrupt Enable Generates interrupt when a measurement completes and the result is ready (CTRLSTAT[RESULT_READY_STAT] = 1).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
13 GT_MAX_INT_EN	Greater Than Maximum Interrupt Enable Generates an interrupt when the result is greater than MAX[MAX_VALUE] (CTRLSTAT[GT_MAX_STAT] = 1). 0b - Disable 1b - Enable
12 LT_MIN_INT_EN	Less Than Minimum Interrupt Enable Generates an interrupt when the result is less than MIN[MIN_VALUE] (CTRLSTAT[LT_MIN_STAT] = 1). 0b - Disable 1b - Enable
11-10 —	Reserved
9 PULSE_POL	Pulse Polarity Specifies whether a pulse width (high period or low period) of the reference clock is measured in the Pulse Width Measurement mode. **NOTE** PULSE_POL is valid only in the Pulse Width Measurement mode. • A high period measurement is triggered by the rising edge on the reference clock input. • A low period measurement is triggered by the falling edge on the reference clock input. 0b - High period 1b - Low period
8 PULSE_MODE	Pulse Width Measurement Mode Select Selects the measurement mode—either Frequency Measurement mode or Pulse Width Measurement mode. In Frequency Measurement mode, measurement begins at the rising edge of the selected reference clock and continues until a count of 2^{REF_SCALE} reference clock pulses is reached. In Pulse Width Measurement mode, the counter starts incrementing when the selected trigger edge (rising edge for high period measurement or falling edge for low period) occurs, and stops at the next edge (falling edge for a high period measurement or rising edge for a low period measurement). Select high or low period measurement by writing the desired value to PULSE_POL. 0b - Frequency Measurement mode 1b - Pulse Width Measurement mode
7-5	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
4-0 REF_SCALE	Reference Clock Scaling Factor Specifies the reference clock scaling factor in Frequency Measurement mode. The reference count cycle is 2^{REF_SCALE}. A higher number provides better accuracy but consumes more processing time. This field is valid only in Frequency Measurement mode.

33.6.1.4 Control Status (CTRLSTAT)

Offset

Register	Offset
CTRLSTAT	4h

Function
Contains the current CTRL_W register configuration fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MEASURE...	CONTINU...	0			RESULT_...	GT_MAX_...	LT_MIN_...	0							
W						W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RESULT_...	GT_MAX_...	LT_MIN_...	0		PULSE_P...	PULSE_M...	0				REF_SCALE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement in Progress Status Indicates the current CTRL_W[MEASURE_IN_PROGRESS] value. 0b - Not in progress 1b - In progress
30	Continuous Mode Enable Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
CONTINUOUS_MODE_EN	Indicates the current CTRL_W[CONTINUOUS_MODE_EN] value. 0b - Disabled 1b - Enabled
29-27 —	Reserved
26 RESULT_READY_STAT	Result Ready Status Indicates that a measurement is complete and CTRL_R[RESULT] is ready to read. Write 1 to this field to clear. 0b - Not complete 1b - Complete
25 GT_MAX_STAT	Greater Than Maximum Result Status Indicates that a measurement is complete and the result is greater than the maximum expected value (CTRL_R[RESULT] > MAX[MAX_VALUE]). Write 1 to this field to clear. 0b - Less than MAX[MAX_VALUE] 1b - Greater than MAX[MAX_VALUE]
24 LT_MIN_STAT	Less Than Minimum Results Status Indicates that a measurement is complete and the result is less than the minimum expected value (CTRL_R[RESULT] < MIN[MIN_VALUE]). Write 1 to this field to clear. 0b - Greater than MIN[MIN_VALUE] 1b - Less than MIN[MIN_VALUE]
23-15 —	Reserved
14 RESULT_READY_INT_EN	Result Ready Interrupt Enable Indicates the current CTRL_W[RESULT_READY_INT_EN] value. 0b - Disabled 1b - Enabled
13 GT_MAX_INT_EN	Greater Than Maximum Interrupt Enable Indicates the current CTRL_W[GT_MAX_INT_EN] value. 0b - Disabled 1b - Enabled
12	Less Than Minimum Interrupt Enable Indicates the current CTRL_W[LT_MIN_INT_EN] value.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LT_MIN_INT_EN	0b - Disabled 1b - Enabled
11-10 —	Reserved
9 PULSE_POL	Pulse Polarity Indicates the current CTRL_W[PULSE_POL] value. 0b - High period 1b - Low period
8 PULSE_MODE	Pulse Mode Indicates the current CTRL_W[PULSE_MODE] value. 0b - Frequency Measurement mode 1b - Pulse Width Measurement mode
7-5 —	Reserved
4-0 REF_SCALE	Reference Scale Indicates the current CTRL_W[REF_SCALE] value.

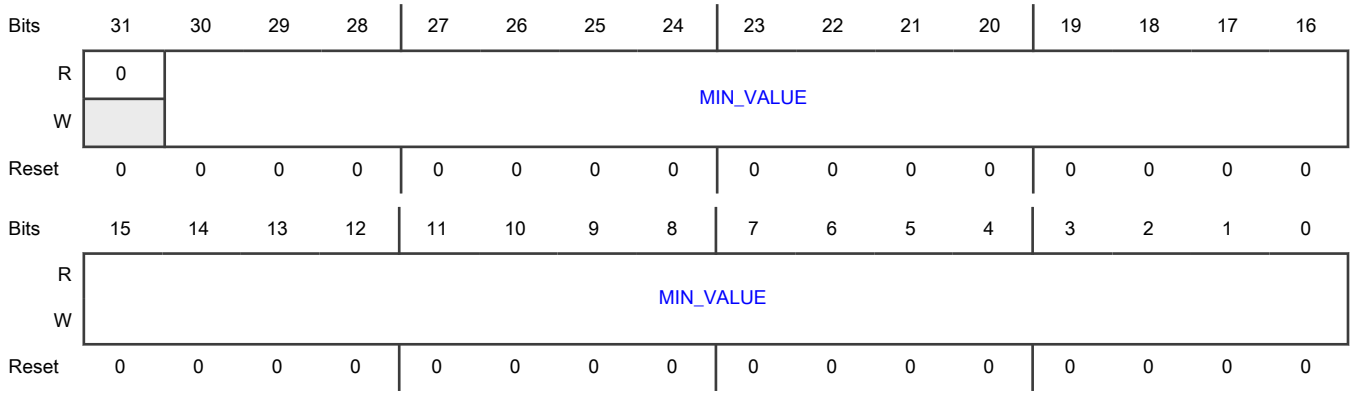
33.6.1.5 Minimum (MIN)

Offset

Register	Offset
MIN	8h

Function
Specifies the minimum expected value for the measurement result.

Diagram



Fields

Field	Function
31 —	Reserved
30-0 MIN_VALUE	Minimum Value Minimum expected value for the measurement result.

33.6.1.6 Maximum (MAX)

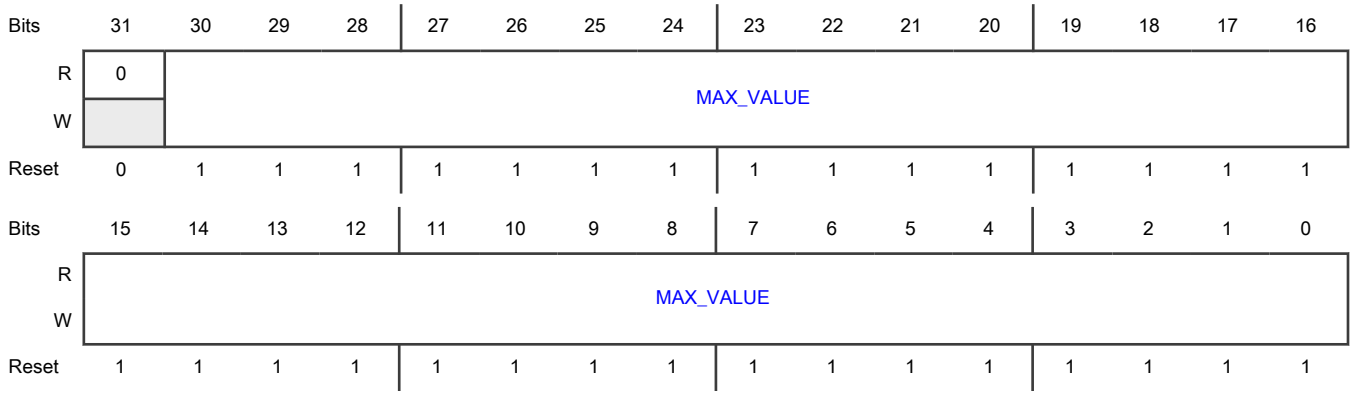
Offset

Register	Offset
MAX	Ch

Function

Specifies the maximum expected value for the measurement result.

Diagram



Fields

Field	Function
31 —	Reserved
30-0 MAX_VALUE	Maximum Value Maximum expected value for the measurement result.

Chapter 34
Nano Power 32 kHz Oscillator (OSC32KNP)

34.1 Chip-specific OSC32KNP information

Table 275. Reference links to related information

Topic	Related module	Reference
Full description	OSC32KNP	OSC32KNP
Clocking		Clock overview Clock controller
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Real time clock subsystem	RTC_SS	RTC_SS
Signal multiplexing	Port control	Signal multiplexing

34.1.1 Module instances

This chip contains one instance of the OSC32KNP module.

Table 276. OSC32KNP instance

Features	OSC32KNP
Located in	VDD1V8_AON
Access	CPU0, CPU1, HiFi4, HiFi1, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

34.1.2 Disable/enable OSC32KNP

You can enable or disable the OSC32KNP module via RTC_SS0_WAKE_TIMER_CTRL[SHUTOFF_OSC32].

34.2 Overview

OSC32KNP contains the following parts:

- An analog module that provides a 32.768 kHz clock to the chip with an external crystal oscillator
- A digital module used to configure the analog part and controls the monitoring of clock stability

34.2.1 Block diagram

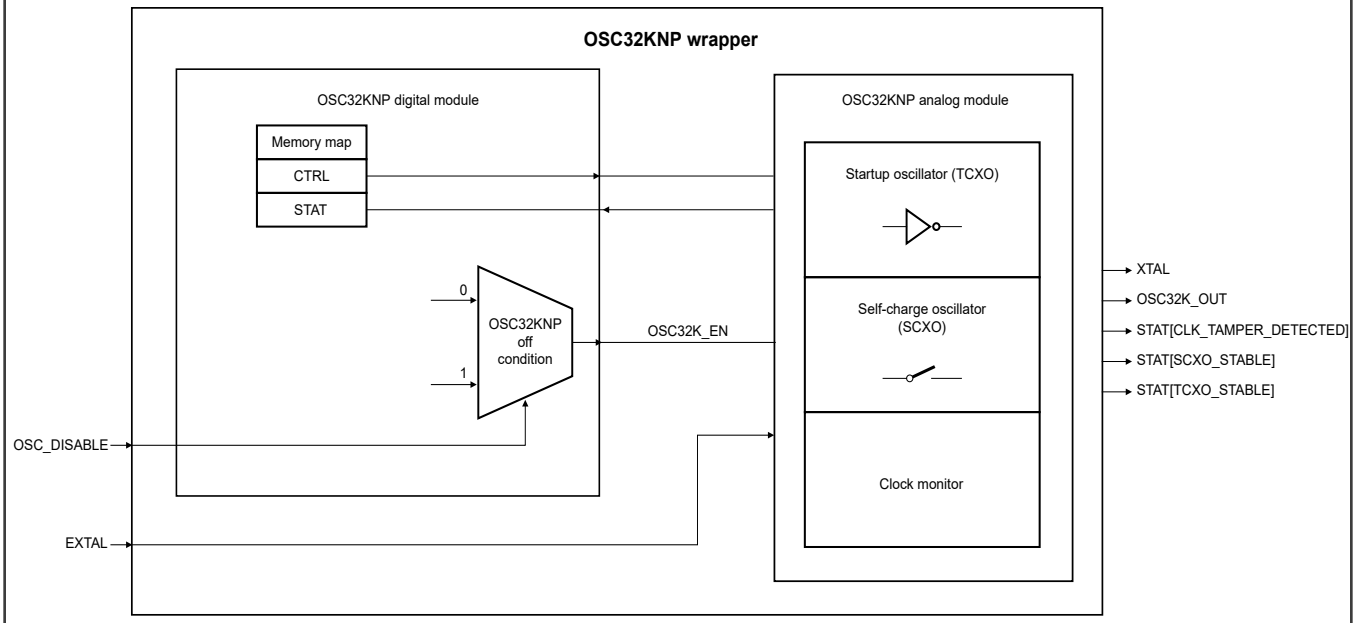


Figure 108. Block diagram

34.2.2 Features

- Provides a 32.768 kHz clock to the chip
- Provides a configurable clock accuracy
- Supports the clock in High-Power and Low-Power (Nano-Power) modes
- Detects clock tamper events across all operational power modes
- Supports current and switch cap resistor clock tracking
- Supports scan channel to turn off analog signals
- Supports disabling of OSC32KNP via OSC_DISABLE input signal

34.3 Functional description

34.3.1 Modes

34.3.1.1 Low-Power mode

The Low-Power mode uses the self-charge oscillator (SCXO) to ensure a ultra-low power consumption in battery application. This mode is also known as Nano-Power mode. OSC32KNP is fully functional in this mode.

34.3.1.2 High-Power mode

The High-Power mode uses the startup oscillator (TCXO) which enables the module to work with a higher ESR (equivalent series resistance) resonator up to 120 kΩ with high immunity performance to the external noise. OSC32KNP is fully functional in High-Power mode.

34.3.1.3 Clock Monitor mode

The clock tamper event test occurs in this mode. This test allows you to monitor the clock in a given system for loss-of-clock errors or out-of-range errors.

34.3.1.4 Bypass mode

In this mode, OSC32KNP recognizes an external 32.768 kHz input clock, and passes the clock directly to the output port.

34.3.2 Disabling OSC32KNP

If this module is not used, especially when there's no external crystal oscillator, OSC32KNP can be disabled.

NOTE

See chip-specific information on how to disable this module for your specific SoC.

After disabling, attempts to access OSC32KNP registers might cause a bus hang.

34.3.3 Clocking

The OSC32KNP registers and logic use the internally generated 32 kHz clock synchronized with the bus clock.

34.3.4 Reset

The bus reset resets the OSC32KNP registers and logic.

34.3.5 Interrupts

This module has no interrupts.

34.4 External signals

Signal	Description	Direction
VDD_1P8	1.8 V power supply	Supply
EXTAL	External crystal oscillator 32.768 kHz input	Input
XTAL	32.768 kHz crystal oscillator output	Output
OSC32K_OUT	OSC32KNP clock output	Output
OSC_DISABLE	Signal for disabling OSC32KNP	Input

34.5 Initialization

This module does not require initialization.

34.6 OSC32KNP register descriptions

34.6.1 OSC32KNP memory map

OSC32KNP base address: 4006_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (CTRL)	32	RW	0F00_080Ah
8h	Clock Status (STAT)	32	RW	0000_0000h

34.6.2 Control (CTRL)

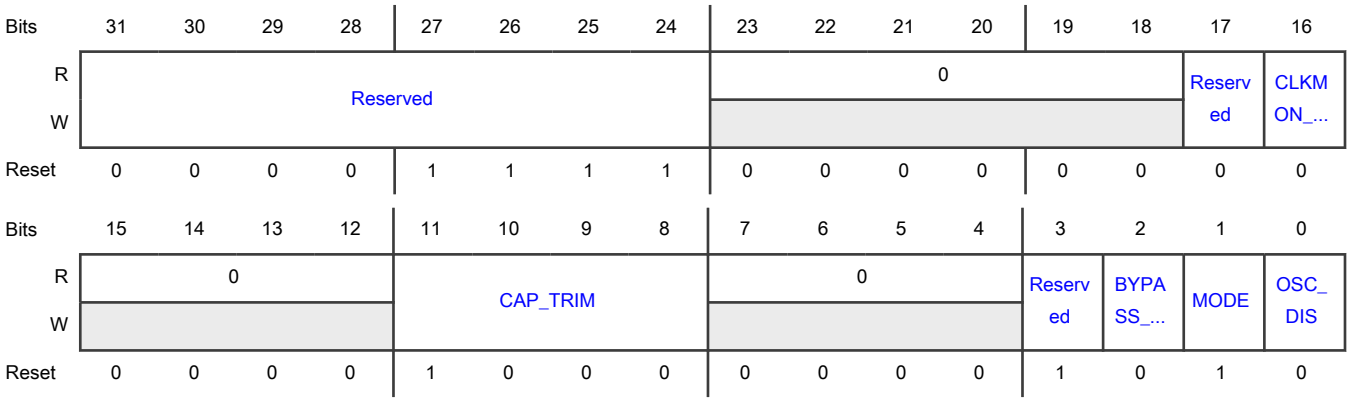
Offset

Register	Offset
CTRL	0h

Function

Provides fields for analog control.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-18 —	Reserved
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 CLKMON_EN	Clock Monitor Enable Enables the clock monitor function. See Clock Monitor mode. 0b - Disable 1b - Enable
15-12 —	Reserved
11-8 CAP_TRIM	Capacitor Trim Value Configures the capacitor load value. 0000b - 0 pF 0001b - 2 pF 0010b - 4 pF 0011b - 6 pF 0100b - 8 pF 0101b - 10 pF 0110b - 12 pF 0111b - 14 pF 1000b - 16 pF 1001b - 18 pF 1010b - 20 pF 1011b - 22 pF 1100b - 24 pF 1101b - 26 pF 1110b - 28 pF 1111b - 30 pF
7-4 —	Reserved
3 —	Do not change the value of this bit.
2 BYPASS_EN	Bypass Mode Enable Enables the Bypass mode of the channel. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 MODE	Mode When CTRL[OSC_DIS] is 1, this field specifies the mode used. If STAT[SCXO_STABLE] is 1, you must set CTRL[OSC_DIS] manually to 0. **NOTE** You can only write 0 to this field from 1, and not vice versa. This field can only be written 1 by reset. 0b - Low-Power (Nano-Power) mode 1b - High-Power mode
0 OSC_DIS	Oscillator Output Disable Disables the output of an oscillator on the channel before changing the OSC32KNP power mode of the channel. 0b - Enable the oscillator output 1b - Disable the oscillator output

34.6.3 Clock Status (STAT)

Offset

Register	Offset
STAT	8h

Function

Provides fields for analog status monitoring.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CLK_TAM...	0						SCXO_ST...	TCXO_ST...
W								W0C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-9 —	Reserved
8 CLK_TAMPER_DETECTED	Clock Tamper Detected Indicates whether a clock tamper event is detected. The clock tamper event reports when the OSC32KNP clock output is out of range. 0b - Not detected 1b - Detected
7-2 —	Reserved
1 SCXO_STABLE	Self-Charge Oscillator (SCXO) Stable Indicates whether the OSC32KNP clock in self-charge is stable. SCXO is used in Low-Power mode. 0b - Unstable 1b - Stable
0 TCXO_STABLE	Startup Oscillator (TCXO) Stable Indicates whether the OSC32KNP clock in transconductance is stable. TCXO is used in High-Power mode. 0b - Unstable 1b - Stable

Chapter 35
Real Time Clock Subsystem (RTC_SS)

35.1 Chip-specific RTC_SS information

Table 277. Reference links to related information

Topic	Related module	Reference
Full description	RTC_SS	RTC_SS
RTC		RTC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

35.1.1 Module instances

This device has two instances of the RTC_SS module, RTC_SS0 and RTC_SS1.

Table 278. RTC_SS instances

Features	RTC_SS0	RTC_SS1
Located in	VDD1_SENSE (HP), VDD1V8_AON (LP)	VDD1_SENSE (HP), VDD1V8_AON (LP)
Access (LP)	CPU0, HiFi4, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

35.2 Overview

RTS_SS provides wake timers and miscellaneous functions for RTC modules.

35.2.1 Block diagram

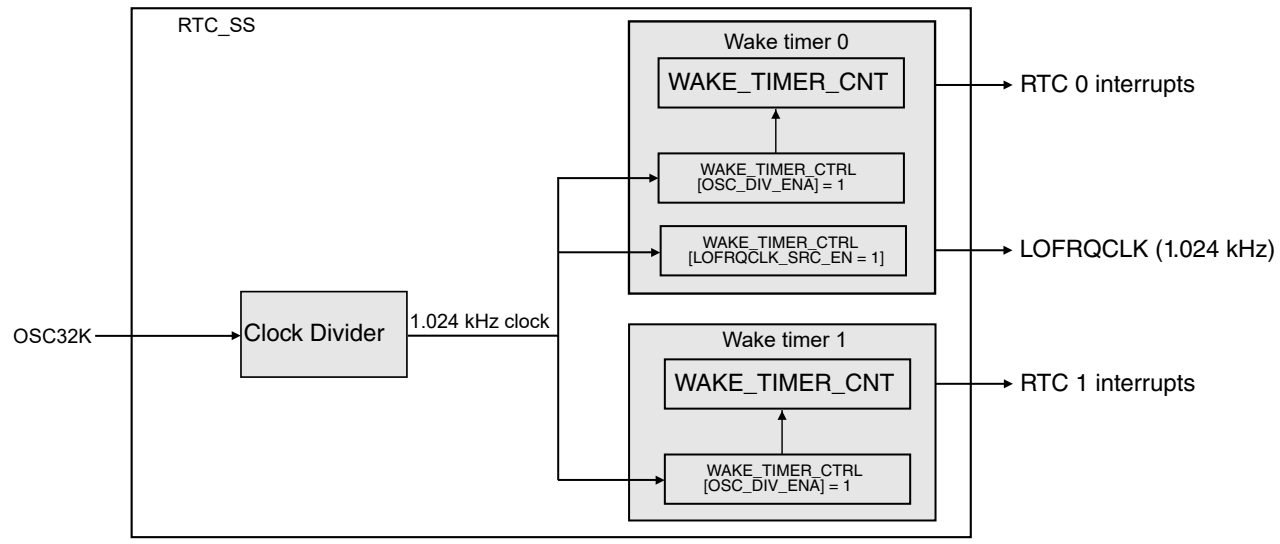


Figure 109. Block diagram

35.2.2 Features

- 32-bit wake-up timer counter in AON domain to wake up the chip.
- 1.024 kHz clock source shut-off control.
- OSC32KNP module shut-off control.

35.3 Functional description

RTC_SS is logically split into 2 parts, RTC_SS0 and RTC_SS1. RTC_SS0 contains a wake timer (Wake Timer 0) and miscellaneous control bits. RTC_SS1 contains a wake timer (Wake Timer 1).

CPU0 and HiFi4 have exclusive access to RTC_SS0 while CPU1 and HiFi1 have exclusive access to RTC_SS1.

35.3.1 Wake timer

Wake timer is a 32-bit down counter clocked at 1.024 kHz when enabled. Writing any non-zero value to the timer's counter register enables the counter and launches a countdown sequence, the counter register decreases by 1 on each 1.024 kHz clock cycle.

Upon timeout (counter value reaches zero), wake timer generates a flag which is also an interrupt source, and then turns itself off until relaunched by writing a non-zero value to the counter register.

When the counter is used as a wake-up interrupt source, the countdown sequence must be launched prior to entering a low-power mode.

35.3.2 Clocking

Signal	Description	Direction
OSC32K	32.768 kHz oscillator clock from OSC32KNP module	Input
LOFRQCLK	1.024 kHz clock source for low-frequency peripherals	Output

35.3.3 Interrupts

RTC_SS has two wake timer interrupt sources. Interrupt 0 (from Wake Timer 0) is routed to CPU0 and interrupt 1 (from Wake Timer 1) is routed to CPU1.

For each wake timer, write 1 to WAKE_TIMER_CTRL[INTR_EN] to enable the interrupt. The interrupt is asserted when the wake timer times out (read value of WAKE_TIMER_CTRL[WAKE_FLAG]=1). The wake timer's status flag bit and interrupt are cleared by writing 1 to WAKE_TIMER_CTRL[WAKE_FLAG]. The status flag is also cleared on module reset.

35.4 External signals

This module has no external signals.

35.5 Initialization

This module has no initialization considerations.

35.6 Memory map and register description

This section includes the RTC_SS memory map and detailed descriptions of all registers.

NOTE

CPU0 and HiFi4 cannot access RTC_SS1 registers. CPU1 and HiFi1 cannot access RTC_SS0 registers.

35.6.1 RTC_SS0 register descriptions

35.6.1.1 RTC_SS0 memory map

RTC_SS0 base address: 4006_8000h

Offset	Register	Width (In bits)	Access	Reset value
C00h	Wake Timer Control (CPU0) (WAKE_TIMER_CTRL)	32	RW	0000_0000h
C0Ch	Wake Timer 0 Counter (CPU0) (WAKE_TIMER_CNT)	32	RW	0000_0000h
C10h	General Purpose (GPR)	32	RW	0000_0000h

35.6.1.2 Wake Timer Control (CPU0) (WAKE_TIMER_CTRL)

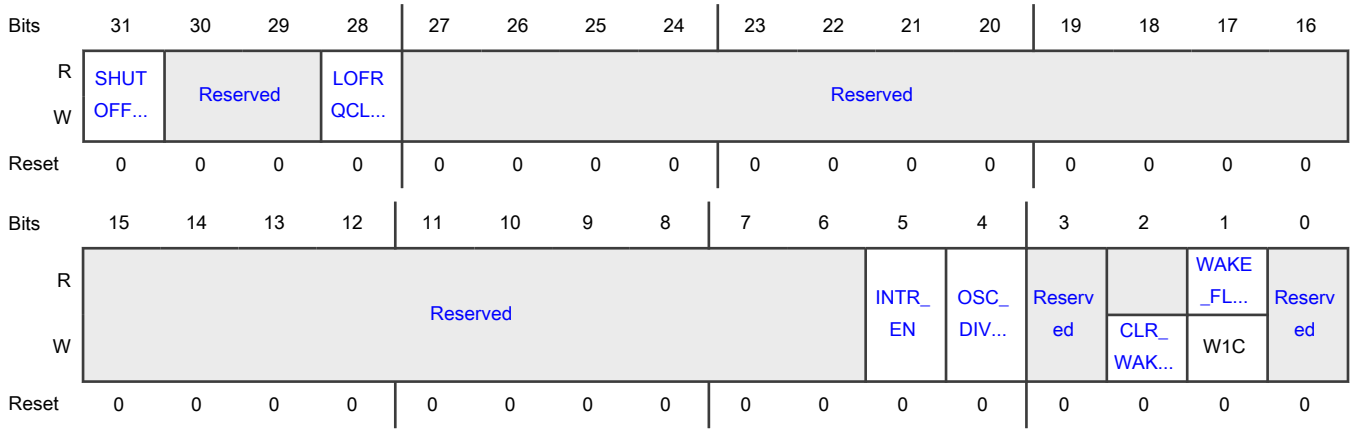
Offset

Register	Offset
WAKE_TIMER_CTRL	C00h

Function

Contains miscellaneous always-on control fields related to wake timer 0 or other modules.

Diagram



Fields

Field	Function
31 SHUTOFF_OSC32	Shut off OSC32KNP Inactivates OSC32KNP module. **NOTE** OSC32KNP module is located outside RTC_SS. You should inactivate the OSC32KNP module when it is not used, especially when there is no external crystal oscillator. After the inactivation, attempts to access OSC32KNP registers can cause bus halt. 0b - Enable 1b - Disable
30-29 —	Reserved
28 LOFRQCLK_SRC_EN	Low Frequency Clock Source Enable Enables 1.024 kHz low frequency clock output. **NOTE** The clock source is reserved for modules outside RTC_SS. 0b - Disable 1b - Enable
27-6 —	Reserved
5 INTR_EN	Interrupt Enable Enables wake timer 0 timeout interrupt. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
4 OSC_DIV_ENA	OSC Divide Enable Enables the divider to generate 1.024 kHz clock for the wake timer 0. OSC_DIV_ENA must be set to 1 to function the wake timer 0. 0b - Disable 1b - Enable
3 —	Reserved
2 CLR_WAKE_TIMER	Clear Wake Timer 0 Clears the wake timer 0 counter and halts operation 0b - No effect. 1b - Clear.
1 WAKE_FLAG	Wake Timer 0 Status Flag Indicates wake timer 0 status. 0b - Not time out 1b - Time out.
0 —	Reserved

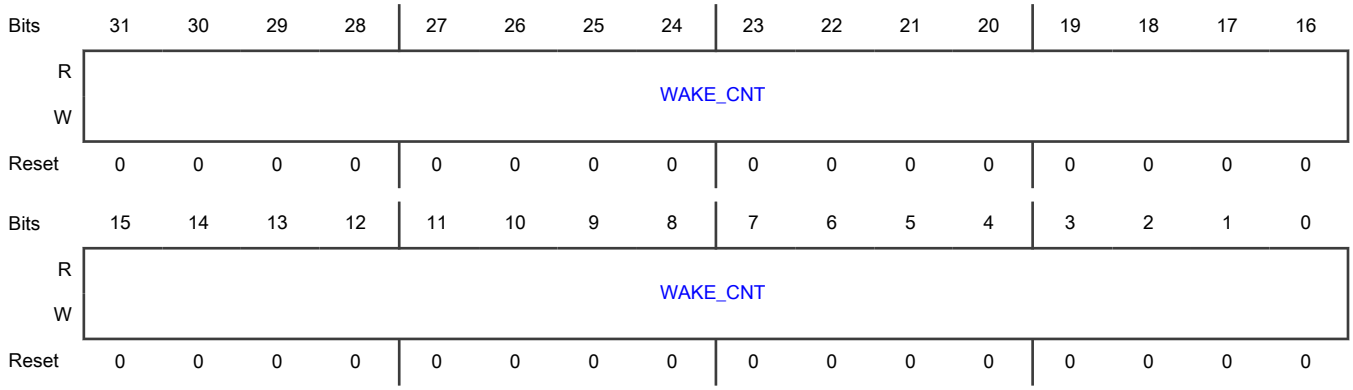
35.6.1.3 Wake Timer 0 Counter (CPU0) (WAKE_TIMER_CNT)

Offset

Register	Offset
WAKE_TIMER_CNT	C0Ch

Function
Specifies the Wake timer counter 0.

Diagram



Fields

Field	Function
31-0 WAKE_CNT	Wake Counter 0 Value A read reflects the current value of the wake timer 0. Due to clock domain crossing, read the register twice in succession and confirm that the two values are the same. If the values are not the same, repeat the procedure until the two values are the same. A non-zero write to this field pre-loads a start-count value into the timer and starts a countdown sequence. During the countdown sequence, wake counter decreases by 1 each 1.024 kHz clock cycle until it reaches 0 (timeout). Do not write to WAKE_TIMER_CNT when down-counting is in progress.

35.6.1.4 General Purpose (GPR)

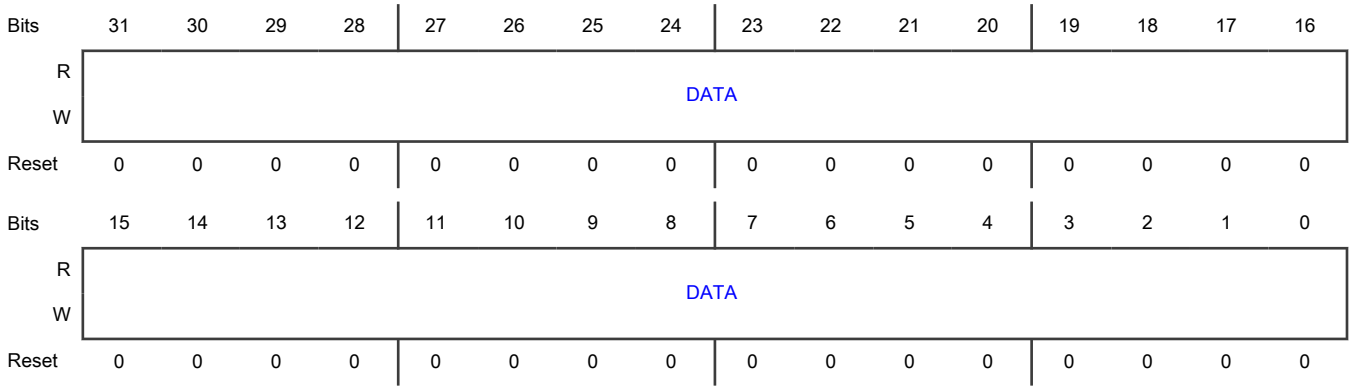
Offset

Register	Offset
GPR	C10h

Function

This register retains contents even during Deep Power Down mode as long as device power is maintained. It is used by ROM code unless using it to configure external reset on XSPI boot-up. This register is programmed by the application for external XSPI boot flash configuration including reset sequence in the case of a non-POR reset sources. Contents of the GPR register can be alternately be configured in the FCB at offset 214h (flashStateCtx). See Reset for more details.

Diagram



Fields

Field	Function
31-0 DATA	Data Data retained during DPD mode or loss of main power.

35.6.2 RTC_SS1 register descriptions

35.6.2.1 RTC_SS1 memory map

RTC_SS1 base address: 4006_9000h

Offset	Register	Width (In bits)	Access	Reset value
C00h	Wake Timer Control (CPU1) (WAKE_TIMER_CTRL)	32	RW	0000_0000h
C0Ch	Wake Timer 1 Counter (CPU1) (WAKE_TIMER_CNT)	32	RW	0000_0000h

35.6.2.2 Wake Timer Control (CPU1) (WAKE_TIMER_CTRL)

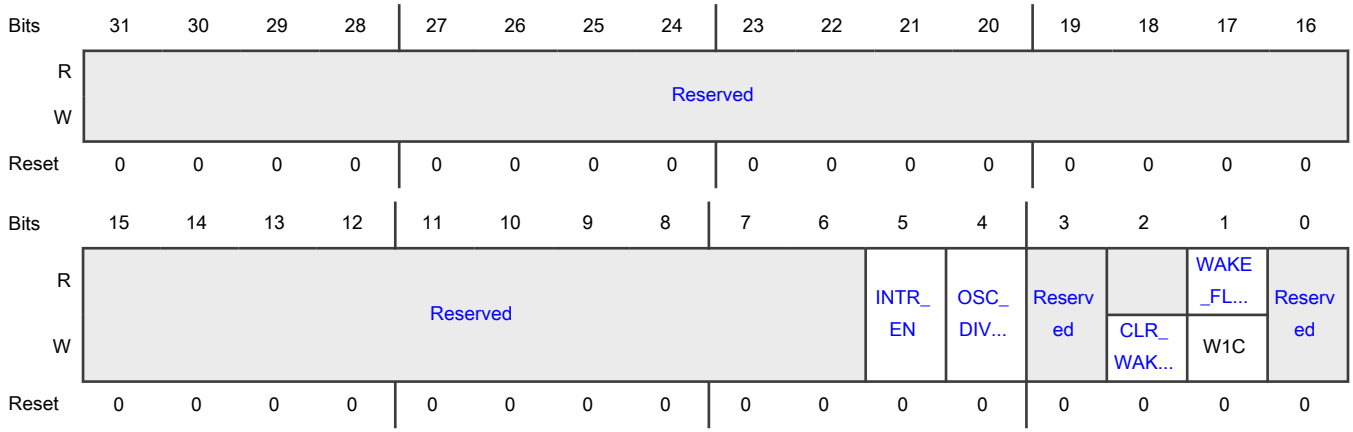
Offset

Register	Offset
WAKE_TIMER_CTRL	C00h

Function

Contains miscellaneous always-on control fields related to wake timer 1.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 INTR_EN	Interrupt Enable Enables wake timer timeout interrupt. 0b - Disable 1b - Enable
4 OSC_DIV_ENA	OSC Divide Enable Enables the divider to generate 1.024 kHz clock for the wake timer 1. OSC_DIV_ENA must be set to 1 to function the wake timer 1. 0b - Disable 1b - Enable
3 —	Reserved
2 CLR_WAKE_TIMER	Clear Wake Timer 1 Clears the wake timer 1 counter and halts operation. 0b - No effect. 1b - Clear.
1 WAKE_FLAG	Wake Timer 1 Status Flag Indicates wake timer 1 status. 0b - Not Timeout. 1b - Timeout.

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 —	Reserved

35.6.2.3 Wake Timer 1 Counter (CPU1) (WAKE_TIMER_CNT)

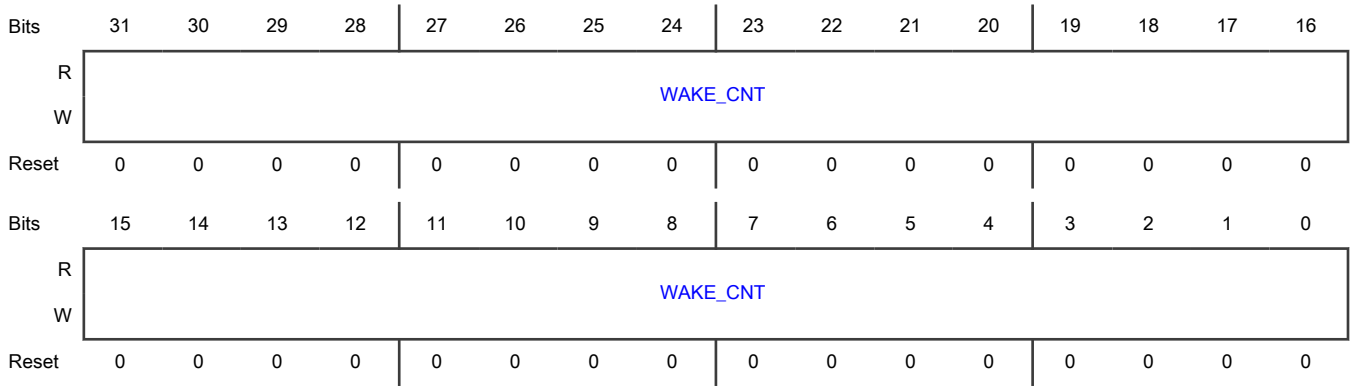
Offset

Register	Offset
WAKE_TIMER_CNT	C0Ch

Function

Specifies wake timer counter 1 value.

Diagram



Fields

Field	Function
31-0 WAKE_CNT	Wake Counter 1 Value A read reflects the current value of the wake timer 1. Due to clock domain crossing, read the register twice in succession and confirm that the two values are the same. If the values are not the same, repeat the procedure until the two values are the same. A non-zero write to this field pre-loads a start-count value into the timer and starts a countdown sequence. During the countdown sequence, wake counter decreases by 1 each 1.024 kHz clock cycle until it reaches 0 (timeout). Do not write to WAKE_TIMER_CNT while down counting is in progress.

Chapter 36
Real Time Clock (RTC)

36.1 Chip-specific RTC information

Table 279. Reference links to related information

Topic	Related module	Reference
Full description	RTC	RTC
RTC_SS		RTC_SS
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

36.1.1 Module instances

This device has two instances of the RTC module, RTC0 and RTC1.

Table 280. RTC instances

Features	RTC0	RTC1
Located in	VDD1_SENSE (HP), VDD1V8_AON (LP)	VDD1_SENSE (HP), VDD1V8_AON (LP)
Access (LP)	CPU0, HiFi4, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Controller or target	Controller	Target

36.1.2 Clocking

The following table shows the clock connections for each RTC instance.

Table 281. Chip level clock source

Module clock signals	Chip level clock source	
	RTC0	RTC1
rtc_clk_out	Not connected	Not connected
rtc_clko_s	Not connected	Not connected
rtc_osc_32k_in_clk_hp	OSC32K	OSC32K
rtc_osc_32k_in_clk_lp	OSC32K	OSC32K

Table continues on the next page...

Table 281. Chip level clock source (continued)

Module clock signals	Chip level clock source	
	RTC0	RTC1
ipg_clk_hp_s	SENSE_MAIN_CLK_1	SENSE_MAIN_CLK_1
ipg_clk_lp_s	1m_lposc	1m_lposc
ipg_clk_hp	1m_lposc	1m_lposc
ipg_clk_lp	Not connected	Not connected
rtc_irc_clk	1m_lposc	Not connected
512 Hz clock	Not connected	Not connected

NOTE

The "Not connected" signals in the above table are not connected to out of the module. Ignore the related information and register fields in this chapter.

36.1.3 VBAT domain

For this chip, VBAT domain refers to the VDD1V8_AON domain. Only the VDD1V8_AO supplies power to the RTC logic in this domain.

In stop mode, VDD1V8_AO keeps supplying power to the RTC logic in VDD1V8_AON domain and the VDD1 to the rest of the RTC logic will be removed.

36.1.4 RTC shared counter

RTC1 shares the data and time counters of RTC0.

36.1.5 Register access after waking up from DSR

When RTC is waking up from DSR, you need to wait for one 32 kHz clock cycle before accessing its registers.

36.2 Overview

Real Time Clock (RTC) is a low-power module that provides time keeping and calendaring functions, protection against spurious memory/register updates. It can also compensate the 1 Hz clock against variations in 32 kHz clock in oscillator due to crystal or temperature.

36.2.1 Block diagram

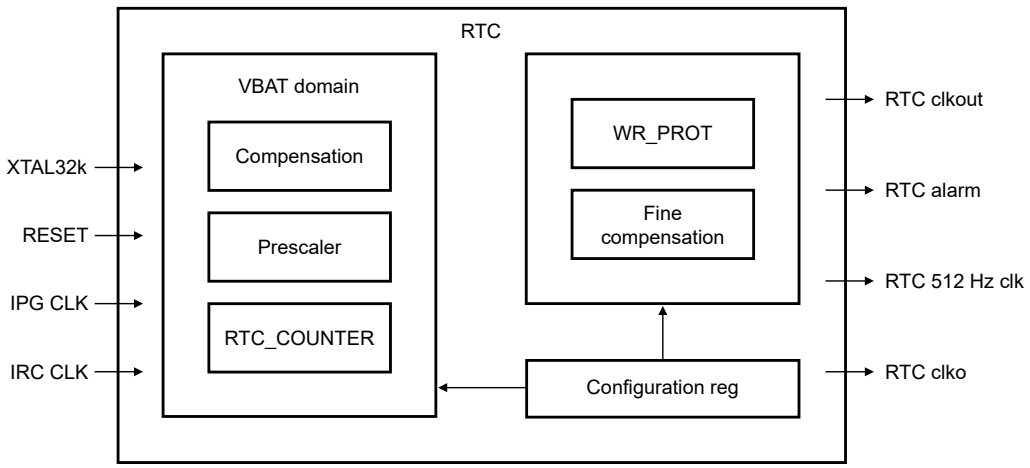


Figure 110. RTC block diagram

36.2.2 Features

RTC includes the following features:

- Designed for low power
  - Time and date counters are interleaved to prevent simultaneous toggling
- Basic clock functions
  - Separate counters for days, hour, minutes, and seconds
  - Calendaring support—Separate counters for year, month, and day of the week
    - Automatic adjustment for daylight saving with user-defined parameters
    - Automatic adjustment for month and leap year
  - External clock support to run the counters if you choose to provide an externally-compensated 1Hz clock
- Time zone offset—RTC uses local time which implicitly contains the time zone offset
- Programmable alarm with interrupt—Alarm is output from RTC in case the MCU uses it as a wake up event
- Periodic interrupts (sampling timer interrupts)
- Hardware compensation—Compensates 1 Hz clock (to the counters) against frequency variations in oscillator clock due to temperature changes or crystal characteristics. Programmable correction factor calculated by firmware.
- 16-bit CPU register programming interface with protection against runaway code
- Option to output the buffered 32.768 kHz clock or the compensated 1 Hz clock.
- The target RTC uses the controller RTC counters.

36.3 Functional description

36.3.1 Modes of Operation

This section describes the RTC modes of operation.

36.3.1.1 Wait mode/Power Down mode

RTC is fully operational in Wait mode (called Wait mode or Power Down mode depending on the device series). RTC runs off a 32.768 kHz clock that is not gated in Wait mode. This means the timekeeping functions (and other functions) that depend on the 32.768 kHz clock continue to operate independent of the CPU. Only the register block's clock is gated and no register contents are lost.

36.3.1.2 Stop mode/Deep Power Down mode

In Stop mode (called Stop mode or Deep Power Down mode depending on the device series), RTC is fully operational and behaves in the same way as it does when in Wait mode. If CPU power (VDD) is removed in Stop mode, RTC switches its power supply source to battery supply (VBAT) and isolates itself from the powered off domains.

NOTE

If the system bus clock is off in Stop mode, no read happens on RTC registers. When the system bus clock is switched on again, synchronization takes time. During the synchronization time, no write is allowed and any read to the RTC register returns FFFEh as the code to indicate the clock domains are synchronizing.

36.3.1.3 VBAT mode

Because it runs off a 32.768 kHz clock, RTC is fully operational in VBAT mode. The 32.768 kHz clock is not gated in VBAT mode and RTC's timekeeping functions (and other functions) that depend on the clock continue to operate independent of the CPU. Only the register block's clock is gated and no register contents are lost. After coming out of VBAT mode, the register content is not accessible for two osc_clk cycles due to synchronization.

36.3.2 Design overview

RTC provides basic timekeeping functions through counters for seconds, minutes, and hours, and calendaring functions via date, day-of-week, month, and year counters, including automatic adjustments for leap years and daylight saving time. Reading these counters indicates the current date and time and writing to these registers sets the date and time as provided by the user.

The alarm is set for a specific hour, minute, and second. If the alarm interrupt is enabled, an interrupt to CPU is generated when the time counters match the configured alarm hour, minute, and second settings. The target RTC uses the controller RTC counters. The alarm can also be configured to match a specific day, month, and year to generate an interrupt. The alarm signal is output to an MCU pin to allow wake up or control of external devices on board.

RTC also provides sampling timer interrupts. Both RTC controller and RTC target have independent alarm functions.

A frequency compensation module is integrated into the RTC to correct any error in the 1 Hz clock due to variations in the 32.768 kHz clock which could be caused by crystal inaccuracy, board variations, or change in temperature. The combined compensation value for both crystal and temperature variation is determined by software and the correction is done in hardware. The target RTC does not need compensation, because in addition to the main counters the target RTC uses the subcounters from the controller RTC.

The registers in RTC are configured via the CPU register programming interface. A protection mechanism is built into RTC to protect its registers against unintended writes by any runaway code. Both the controller RTC and target RTC have the protection mechanism.The protection mechanism requires the CPU to write a specific sequence of codes to the STATUS[7:6] bits to allow write access to the registers. After updating the registers, the CPU can write 10 to STATUS[7:6] bits to enable the write protection. After unlocking the registers, the CPU has a window of two seconds in which to update the register space. Following power-on reset, a window of 15 seconds is allowed for the CPU to configure the RTC, after which the registers are locked. Any updates beyond the unlock window require the CPU to unlock the registers, again.

For detailed description on the complete functionality of RTC, see Time and calendaring functions.

36.3.3 Clocking

The 32.768 kHz clock input from the oscillator is divided to generate a 512 Hz and a 1 Hz clock in RTC.

Table 282. Clocking

Signal name	Description	Direction
rtc_clk_out	1 Hz or 32.768 kHz clock output. Option for selecting compensated 1 Hz clock or buffered 32.768 kHz clock is available in Control (CTRL).	Output
rtc_clko_s	Gated 32.768 kHz clock output to other peripherals. Controlled by CTRL[CLKO_DIS].	Output
rtc_osc_32k_in_clk_hp	32.768 kHz clock input from oscillator, used in HP domain.	Input
rtc_osc_32k_in_clk_lp	32.768 kHz clock input from oscillator, used in LP domain.	Input
ipg_clk_hp_s	Gated system bus clock used for register read or write for high power or the CPU domain.	Input
ipg_clk_lp_s	Gated system bus clock used for register read or write for low power or the battery domain.	Input
ipg_clk_hp	System bus clock used to synchronize the oscillator 32.768 kHz clock to the ipg_clk domain (free running clock), used in HP domain.	Input
ipg_clk_lp	System bus clock used to synchronize the oscillator 32.768 kHz clock to the ipg_clk domain (free running clock), used in LP domain.	Input
rtc_irc_clk	Mhz clock used to fine compensate the oscillator clock.	Input

RTC has a 32.768 kHz clock input from the oscillator. The clock becomes the output to the peripherals and divided to generate a 512 Hz clock and a 1 Hz clock. The 512 Hz clock becomes the output to peripherals and the 1 Hz clock is the input of the counter.

The 1 Hz clock is subject to variations in speed due to temperature changes or crystal irregularities. The input clock rtc_irc_clk runs in cycles of MHz and compensates for these variations by making precise adjustments to the 1 Hz clock.

RTC has one system bus clock used to synchronize the oscillator 32.768 kHz clock to the ipg_clk domain.

Writing to CTRL[CLKOUT] configures either the clock input (32.768 kHz) or the 1 Hz clock as the output from the SOC for use outside RTC. Writing to CTRL[CLKO_DIS] determines if the selected clock outputs to other peripherals.

36.3.4 Interrupts

Write one to IER[ALM_IE] to enable the alarm interrupt. The alarm interrupt is asserted when the programmed alarm value matches the counter values and the ISR[ALM_IS] bit is set. The status bit and interrupt are cleared by writing one. The status register is also cleared on software reset. The counters matched for the alarm interrupt are selected based on the alarm type set in CTRL[ALM_MATCH].

There are eight periodic, sampling timer interrupts. When the clock generation counter counts to the programmed value, the corresponding bit in the status register is set. The status bits are cleared either by writing one or by software reset.

36.3.5 Time and calendaring functions

The controller RTC performs and controls all the chronological functions listed below.

The target RTC shares the controller RTC counters for these functions.

- Implements all counters for date and time and the related control logic
- Calculates leap year and adjusts the day count accordingly
- Increments or decrements counters to adjust for leap seconds
- Tracks the number of days in a month
- Automatically adjusts for daylight saving time

- Generates alarm with selectable matching of different counters. Both controller and target RTC have independent alarm functions.

Dynamic modifications to the date counters are done based on leap year, month, and daylight saving. Additionally, the user software can add or subtract a second to adjust for leap seconds. All changes are controlled by hardware and triggered by software. Used on controller RTC counters only.

A leap year is defined as a year in which the year value is divisible by 4 and 400 and has an extra day in February. Daylight saving is implemented in different regions of the world to shift the local time according to the summer or winter seasons. Time is shifted at a pre-defined time decided by the regional conditions and programmed by the user software. RTC automatically adjusts the time using hardware alarms for daylight saving. The day counter is also adjusted for months which have 28, 29, 30, or 31 days. Used on controller RTC counters only.

Alarm generation is done by the RTC block. Matching the counters is controlled by CTRL[ALM_MATCH] to give various alarm options for daily, monthly, yearly, or one-time alarms.

NOTE

Setting an hour alarm value that coincides with the daylight saving start hour value will not generate an alarm because the hour counter is incremented by two (instead of one) when daylight saving comes into effect. For example, setting ALM_HOURMIN to 0x1500 (hour alarm value = 15 or 3 PM) when DST_HOUR = 0x1411 (DST start hour value = 14 or 2 PM) will cause the hour counter to go from 14 to 16. The hour counter will not equal 15 and therefore the alarm which was set for 15 (or 3 PM) will not trigger. User software must take daylight saving changes into account when setting alarm values.

NOTE

User software must program an alarm time equal to the daylight saving end time (fallback time). The user software must write 0 to CTRL[DST_EN] when the alarm interrupt occurs, otherwise the correct operation might not take place.

36.3.6 RTC compensation logic

The compensation logic supports temperature and frequency compensation and provides an accurate and wide range of compensation suitable for many crystals. It can correct a wide range of crystal offsets from as low as 0.119 PPM.

NOTE

The target RTC does not need compensation logic because it uses compensation logic from the controller RTC.

There are two important components in the compensation logic: coarse compensation and fine compensation. The coarse compensation provides an accurate clock to RTC's internal time and date counters, and fine compensation generates an accurate 1 Hz clock output (via MCU pin) with high resolution clock edge placement (up to 0.88 ppm) and near 50 percent duty cycle.

Enabling the required compensation logic

Coarse compensation is enabled by writing to CTRL[COMP_EN] and fine compensation is enabled by writing to CTRL[FINEEN]. However, when FINEEN = 1, COMP_EN must be equal to 1. Setting both bits to 0 disables all compensation. This is the default state out of reset.

Compensation parameters need to be provided to allow required compensation logic to run. The parameters are provided by writing to the COMPEN register.

NOTE

If CTRL[FINEEN] = 0, then compensation must be disabled using CTRL[COMP_EN] before changing COMPEN for the first time. Successive changes can be done without disabling compensation.

If FINEEN = 1, then there are two options:

- Either clear previously accumulated fractional compensation value by disabling compensation from the CTRL register, program new values, and re-enable compensation and start a fresh
- Or, overwrite the current values leading to the addition of previously accumulated fractional compensation value to programmed fractional compensation value.

The integer part is always overwritten

Compensation parameters

Compensation logic requires the user software to provide compensation parameters in the COMPEN register in order to generate an accurate 1 Hz clock. These parameters are defined depending on the type of compensation enabled.

The compensation parameters (when FINEEN = 0) are defined as:

- **Compensation correction value:** Compensation correction value is a two's complement value by which the 1 Hz Clock is modified (during its generation) by either adding or removing RTC oscillator clock cycles.
- **Compensation interval:** Compensation interval is the duration in seconds over which the correction is applied. This is the time in which the addition or removal of 32.768 kHz clock cycles is done, thereby ensuring that the compensation interval is close to the interval obtained with an ideal 1 Hz clock.

The compensation parameters (when FINEEN = 1) are defined as:

- **Integral compensation value:** This is a two's complement value of the integer part of correction or compensation value that has to be adjusted in every 1 second period. This value is expressed in terms of the number of clock cycles of the RTC oscillator clock.
- **Fraction compensation value:** This is the fractional part of the correction or compensation value that has to be adjusted. This value is expressed as the number of clock cycles of a fixed 4.194304 MHz clock that have to be added. This value is always a positive number.

NOTE

When FINEEN = 1, the compensation interval is by default set to 1, indicating that the compensation will be done at every second.

Coarse compensation logic

Coarse compensation logic provides the accurate 1 Hz clock pulses to the time and date counters and the coarse 1 Hz clock to the fine compensation logic.

When FINEEN = 0: RTC compensates the clock over the provided compensation interval. The addition or removal of clocks is done in a single 1 Hz period, leaving the other 1 Hz periods of the compensation interval the same. This addition or removal of RTC oscillator clock cycles adjusts the 1 Hz clock in a way that keeps the overall compensation interval time close to the same interval being measured with an ideal clock.

When FINEN = 1: The integer part of the compensation (or correction) value is added or removed every 1 Hz period and the fraction part is accumulated and adjusted when the accumulated value equals 1 RTC oscillator clock cycle period.

To perform crystal offset compensation, the user software computes the compensation parameters external to RTC and, using details about the crystal characteristics (aging, drift, and so on), computes the correction factor. The user software then programs it in the two's complement format in the compensation register (COMPEN). Based on the values written in the COMPEN register, the compensation is performed.

To perform temperature compensation, the user software can maintain a lookup table in its memory which lists the change in frequency of 32.768 kHz crystal clock for each degree change in temperature. The CPU wakes up periodically to measure the external temperature via a temperature sensor connected to an A/D converter. The user software uses the lookup table to

determine the compensation factor and writes the value to be compensated (in terms of number of 32.768 kHz clock cycles in two's complement format) in the compensation register (COMPEN). Based on the new values written, the compensation logic adjusts the clock from the next compensation interval.

The user software must compute a common compensation factor if it detects a variation in 32.768 kHz clock due to both temperature and crystal characteristics.

Vital statistics

- Range of compensation interval: 1 second to 255 seconds. If FINEEN = 0, writing a 0 disables compensation. If FINEEN = 1, this interval is always 1 second.
- Range of compensation: -128 to +127 (number of 32.768 kHz clock cycles)
- Selection criteria: Compensation is done only when enabled by user software. User software can disable compensation by writing a 0 compensation interval or setting CTRL[COMPEN] = 0.

Fine compensation logic

The fine compensation logic takes the coarse 1 Hz clock (from coarse compensation block) and generates accurate 1 Hz clock output with accurate clock edge placement and near 50 percent duty cycle.

The fine compensation runs when FINEEN = 1 and uses the MCU's IRC clock to adjust the fractional part of the compensation value every 1 Hz period. This provides an accurate edge placement on the 1 Hz clock.

The fine compensation module automatically adjusts the fractional compensation value for any variation in the IRC clock.

NOTE

Because the IRC clock is generated on MCU, this clock will be disabled when MCU power is OFF or in certain low power modes. In this case, the coarse 1 Hz clock is output from the MCU. The coarse 1 Hz clock is not a 50 percent duty cycle clock.

36.3.7 Write protection mechanism

The write protection mechanism protects the RTC registers from any unexpected updates that can happen from a runaway code. This logic monitors the values written to STATUS[WE]. By default, unconditional write access is allowed to these bits only. For writing the locking and unlocking sequence, 8-bit access should be configured in STATUS[7:0].

To enable write protection, write 10 to these bits. To disable write protection, write the following sequence to those bits: 00, 01, 11, 10.

After a power-on reset, the write protection mechanism is disabled, allowing the user software to configure the RTC block. After configuration is complete, the user code can enable the write protection mode. If this is not done by the user software, the registers are put into write protect mode 15 seconds after power-on. If the write protection mode is unlocked to update registers and not locked again by the user software, the write protection mode is automatically enabled after two seconds.

Any write access made to the registers when write protection is enabled (that is, registers are locked) will cause the transfer error signal to be asserted. Reads are allowed at all times.

NOTE

1. Always check STATUS[WRITE_PROT_EN] after running unlocking and locking sequences and re-run the sequence if STATUS[WRITE_ PROT_EN] is not in the required state.
2. Two consecutive unlocking sequences will lock the registers. After they are unlocked, running an unlock sequence before the timeout will lock RTC registers.

36.4 External Signals

Table 283. External signals

Signal	Description	Direction
RTC_CLKOUT	Output clock to external devices on the board.	Output

36.5 Memory map and registers

36.5.1 RTC register descriptions

The register address is the sum of the base address and the address offset. The base address is defined at the chip level. The address offset is defined at the module level.

NOTE

A 32.768 kHz clock is needed to initialize the RTC. This clock can later be gated when programming the RTC registers.

NOTE

Use valid dates to configure registers. For example, do not configure a value like February 30, 2024, because there is no 30th in February 2024.

36.5.1.1 RTC memory map

RTC0 base address: 4006_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Year and Month Counters (YEARMON)	16	RW	0001h
2h	Days and Day-of-Week Counters (DAYS)	16	RW	0001h
4h	Hours and Minutes Counters (HOURMIN)	16	RW	0000h
6h	Seconds Counters (SECONDS)	16	RW	0000h
8h	Year and Months Alarm (ALM_YEARMON)	16	RW	0000h
Ah	Days Alarm (ALM_DAYS)	16	RW	0000h
Ch	Hours and Minutes Alarm (ALM_HOURMIN)	16	RW	0000h
Eh	Seconds Alarm (ALM_SECONDS)	16	RW	0000h
10h	Control (CTRL)	16	RW	0200h
12h	Status (STATUS)	16	RW	0000h
14h	Interrupt Status (ISR)	16	RW	0000h
16h	Interrupt Enable (IER)	16	RW	0000h
1Ch	Sub Second Counter (RTC_TEST2)	16	R	0000h
22h	Daylight Saving Hour (DST_HOUR)	16	RW	0000h
24h	Daylight Saving Month (DST_MONTH)	16	RW	0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
26h	Daylight Saving Day (DST_DAY)	16	RW	0000h
28h	Compensation (COMPEN)	16	RW	0000h

36.5.1.2 Year and Month Counters (YEARMON)

Offset

Register	Offset
YEARMON	0h

Function

Stores the value of the month and year counters. The year field stores the offset in years from a hardcoded base year (2112), not the actual year value. This is a signed value that ranges from -128 to +127. The actual year value can be calculated by adding the base year (2112) and the offset in the YEARMON[15:8] register, meaning the range of years supported is 1984 (2112 - 128) to 2239 (2112 + 127). For example, if the current year is 2007, then it will be represented in this register as 105 or 0x97. Software should program the offset from the base year into this register.

For year calculation:

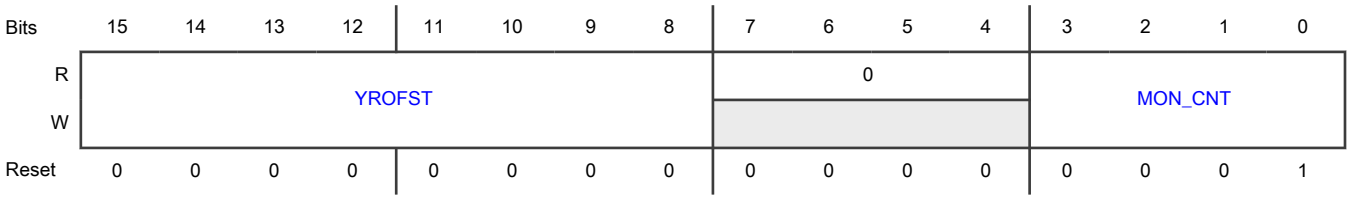
Actual Year = Base Year (for example, 2112) + Offset Year

The month field stores the count value of the months register. Writing to this register loads the months counter with this new value. The valid values are shown in table below.

User software should first determine the state of STATUS[INVAL_BIT] to determine if the counters are stable before a value can be read or changed. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

Both month and year are unaffected by software reset.

Diagram



Fields

Field	Function
15-8 YROFST	Year Offset Count Value

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates the offset in years from the base year (hard coded as 2112) and does not show the actual year value. This is a signed value. Valid values are -128 to 127. The base year is 2112 and if the value of YEAR field is 0x10, the actual year is 2112 + 16 = 2128.
7-4 —	Reserved
3-0 MON_CNT	Month Counter Provides the value of the Months Counter. Valid Values are: 0000b,1101b,1110b,1111b - Illegal Value 0001b - January 0010b - February 0011b - March 0100b - April 0101b - May 0110b - June 0111b - July 1000b - August 1001b - September 1010b - October 1011b - November 1100b - December

36.5.1.3 Days and Day-of-Week Counters (DAYS)

Offset

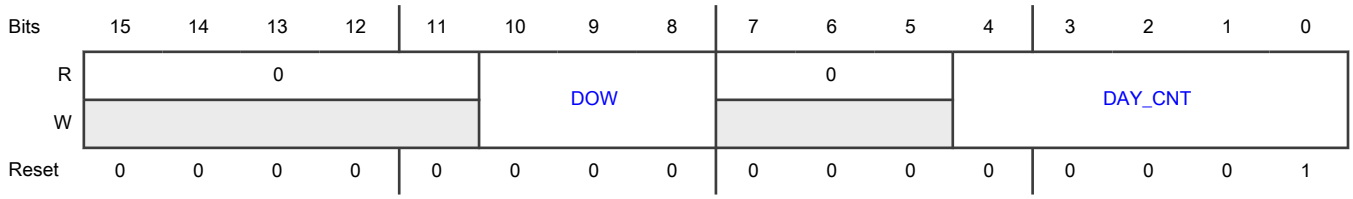
Register	Offset
DAYS	2h

Function

Shows the current value of the day-of-week counter and the days counter. Reading this register returns the latest value of the counters. Writing to this register loads the value to the day-of-week and days counters, and the counters continue to count from this new value. The day-of-week is not calculated automatically and should be written by CPU. This register is unaffected by software reset.

User software should first determine the state of STATUS[INVAL_BIT] to determine if the counters are stable before their value can be read or changed. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

Diagram



Fields

Field	Function
15-11 —	Reserved
10-8 DOW	Day of Week Counter Value 000b - Sunday 001b - Monday 010b - Tuesday 011b - Wednesday 100b - Thursday 101b - Friday 110b - Saturday 111b - Reserved
7-5 —	Reserved
4-0 DAY_CNT	Days Counter Value Valid values are 1 to 31.

36.5.1.4 Hours and Minutes Counters (HOURMIN)

Offset

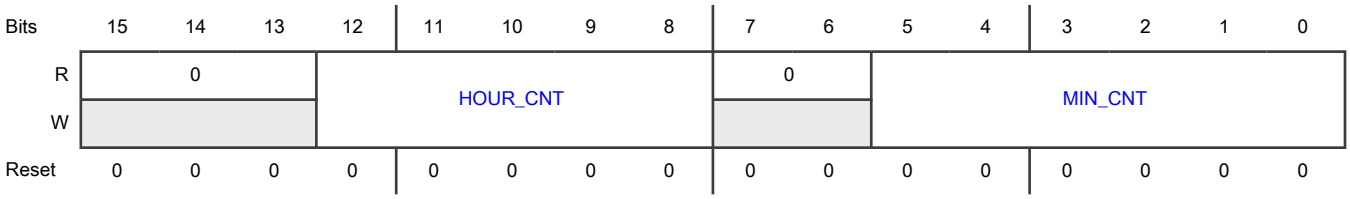
Register	Offset
HOURMIN	4h

Function

Used to program the hours and minutes counter. The register can be read anytime to get the current value of the counters. The hours counter can be set to anything between 0 and 23. The minutes counter can be set to anything between 0 and 59. This register is unaffected by software reset. Only power-on reset can reset this register.

User software should first determine the state of the STATUS[INVAL_BIT] to determine if the counters are stable before their value can be read or changed. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

Diagram



Fields

Field	Function
15-13 —	Reserved
12-8 HOUR_CNT	Hours Counter Value Valid count values are 0 to 23.
7-6 —	Reserved
5-0 MIN_CNT	Minutes Counter Value Valid count values are 0 to 59.

36.5.1.5 Seconds Counters (SECONDS)

Offset

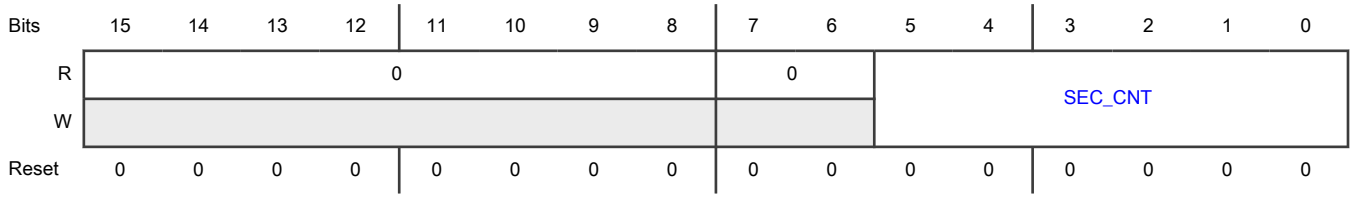
Register	Offset
SECONDS	6h

Function

Used to program the seconds counter. This register can be read anytime to get the current value of the counter. Seconds counter can be set to anything between 0 and 59 both included. This register is unaffected by software reset. Only power-on-reset can reset this register.

User software should first determine the state of the STATUS[INVAL_BIT] to determine if the counters are stable before their value can be read or changed. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

Diagram



Fields

Field	Function
15-8 —	Reserved
7-6 —	Reserved
5-0 SEC_CNT	Seconds Counter Value Valid count values are 0 to 59.

36.5.1.6 Year and Months Alarm (ALM_YEARMON)

Offset

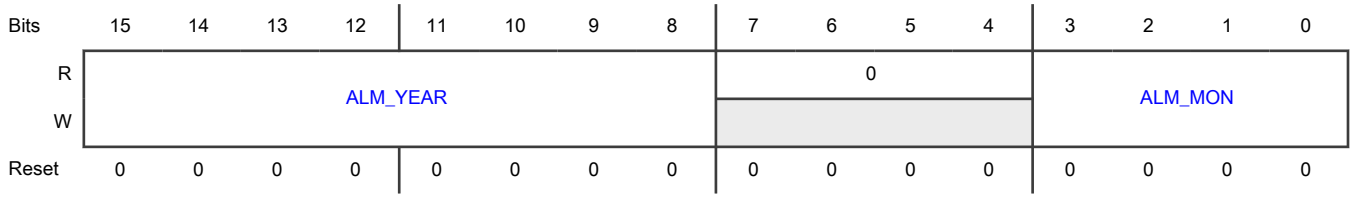
Register	Offset
ALM_YEARMON	8h

Function

Used to configure the months and year setting of the alarm. The alarm setting can be read or written anytime. Alarm interrupt bit is set when all values of alarm seconds, minutes, hours, days, month, and year match their respective counter values. This register is reset to its default state on software reset.

User software can configure the alarm type by using the CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-8 ALM_YEAR	Year Value for Alarm Same as years offset value in Year and Month Counters (YEARMON) .
7-4 —	Reserved
3-0 ALM_MON	Months Value for Alarm Same as months counter value in Year and Month Counters (YEARMON).

36.5.1.7 Days Alarm (ALM_DAYS)

Offset

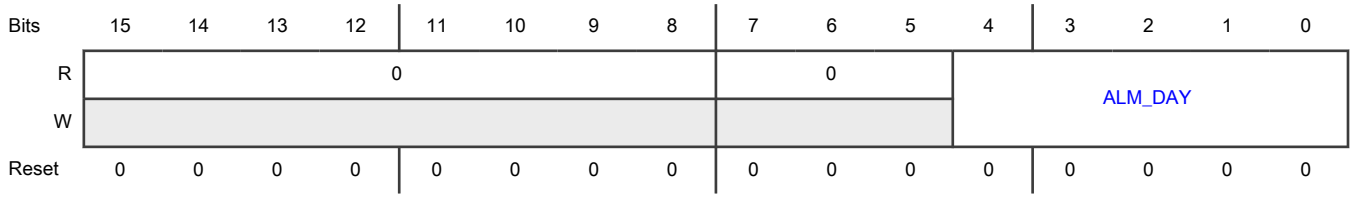
Register	Offset
ALM_DAYS	Ah

Function

Used to configure the day setting of the alarm. The alarm setting can be read or written anytime. Alarm interrupt bit is set when all values of alarm seconds, minutes, hours, days, month, and year match their respective counter values. This register is reset to its default state on software reset.

User software can configure the alarm type using CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-8 —	Reserved
7-5 —	Reserved
4-0 ALM_DAY	Days Value for Alarm Same as days counter value in Days and Day-of-Week Counters (DAYS) .

36.5.1.8 Hours and Minutes Alarm (ALM_HOURMIN)

Offset

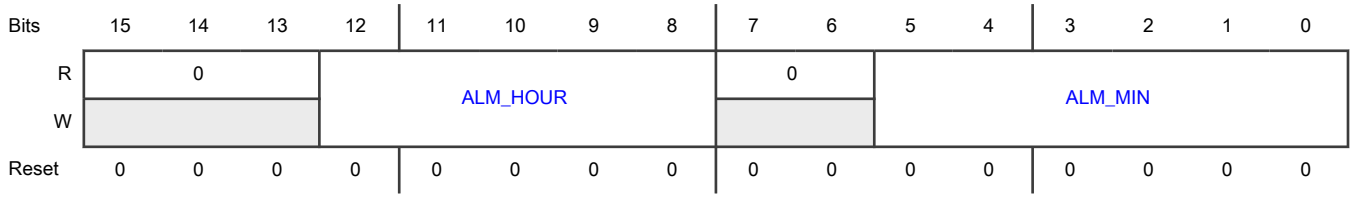
Register	Offset
ALM_HOURMIN	Ch

Function

Used to configure the hour and minute setting of the alarm. The alarm setting can be read or written anytime. This register is reset to default state on software reset.

User software can configure alarm type using CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
12-8 ALM_HOUR	Hours Value for Alarm Same as hours counter value in Hours and Minutes Counters (HOURMIN) .
7-6 —	Reserved
5-0 ALM_MIN	Minutes Value for Alarm Same as minutes counter value in Hours and Minutes Counters (HOURMIN) .

36.5.1.9 Seconds Alarm (ALM_SECONDS)

Offset

Register	Offset
ALM_SECONDS	Eh

Function

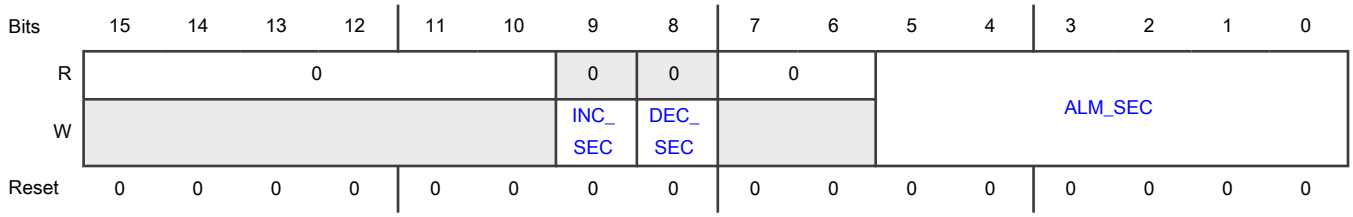
Used to configure the seconds setting of the alarm. The alarm setting can be read or written anytime. This register is reset to default value on software reset.

User software has the option of using bits 9:8 to perform a correction on the seconds counter to compensate for leap seconds. Writing to these bits adds or subtracts 1 from the seconds counter and read returns zeros.

User software should first determine the state of STATUS[INVAL_BIT] to determine if the counters are stable before they can be incremented or decremented. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

User software can configure alarm type using CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
9 INC_SEC	Increment Seconds Counter by 1. Controls the increment of seconds counter in case the software needs to make corrections to compensate for leap seconds or to perform fine trimming of time when needed. Writing to this bit increments the seconds counter and then the bit is cleared on the next posedge.
8 DEC_SEC	Decrement Seconds Counter by 1. Controls the decrement of the seconds counter in case software needs to make corrections for leap seconds or to perform fine trimming of time when needed. Writing to this bit has decrements the seconds counter and then the bit is cleared on the next posedge of the bus clock.
7-6 —	Reserved
5-0 ALM_SEC	Seconds Alarm Value Same as seconds counter value in Seconds Counters (SECONDS) .

36.5.1.10 Control (CTRL)

Offset

Register	Offset
CTRL	10h

Function

Governs all operations occurring inside the RTC. Used to specify the software reset, daylight controls, and the type of alarm function needed.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CLKOUT		0		CLKO_DIS	1	0	0	DST_EN	0	Reserved	ALM_MATCH		COMP_EN	FINEEN
W								SWR								
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
14-13 CLKOUT	RTC Clock Output Selection Selects which clock to output from SoC for use outside RTC. 00b - No output clock 01b - Fine 1 Hz clock with both precise edges 10b - 32.768 kHz clock 11b - Coarse 1 Hz clock with both precise edges
12-11 —	Reserved
10 CLKO_DIS	Clock Output Disable Determines whether the selected clock is output to other peripherals. 0b - The selected clock is output to other peripherals. 1b - The selected clock is not output to other peripherals.
9 —	Reserved
8 SWR	Software Reset Self clearing bit. Asserting this field clears the contents of alarm, interrupt (status and enable) registers, STATUS[CMP_DONE], and STATUS[BUS_ERR], and has no effect on DST, calendaring, time registers. 0b - Software Reset cleared 1b - Software Reset asserted
7 —	Reserved
6 DST_EN	Daylight Saving Enable Enables daylight saving function. The date and time for daylight saving changes are stored in the Daylight Saving Registers. These registers can be changed when this bit is 0. After this bit is set, those registers cannot be changed. When time and date match the values in those registers, daylight adjustment occurs. To disable daylight saving function, write 0 to this bit. 0b - Disabled. Daylight saving changes are not applied. Daylight saving registers can be modified. 1b - Enabled. Daylight saving changes are applied.
5 —	Reserved
4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
3-2 ALM_MATCH	Alarm Match Defines the type of alarm function. Selects which time and calendar counters are used for matching and will generate an alarm. 00b - Only seconds, minutes, and hours matched. 01b - Only seconds, minutes, hours, and days matched. 10b - Only seconds, minutes, hours, days, and months matched. 11b - Only seconds, minutes, hours, days, months, and year (offset) matched.
1 COMP_EN	Compensation Enable **NOTE** Hardware will not allow user software to write 1 to COMP_EN if both the FINEEN and COMP_EN bits are set to 1'b1. 0b - Coarse compensation is disabled. 1b - Coarse compensation is enabled.
0 FINEEN	Fine Compensation Enable 0b - Fine compensation is disabled 1b - Fine compensation is enabled.

36.5.1.11 Status (STATUS)

Offset

Register	Offset
STATUS	12h

Function

Indicates the status of various processes occurring inside the RTC. Also helps the user software to read time or date register when the values are stable and not changing. Compensation Done bit and Bus Error bit are cleared by writing 1. Software reset resets the whole register to its default state.

All memory mapped registers are protected against spurious updates by the write protect mechanism. To unlock the registers, a specific pattern (as mentioned in the table above) has to be written in the write enable bits (WE[1:0]) to enable or disable write protection. The WE[1:0] bits are the only bits that are freely writeable by user software. The write enable bits are self-clearing bits that always return zeros on read.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CMP_DONE	0	0	BUS_ERR	0		CMP_INT	0	0		WRITE_P...	INVAL_B...
W					W1C			W1C	WE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 —	Reserved
11 CMP_DONE	Compensation Done Indicates that the current compensation cycle is complete. This field is cleared by writing 1. This bit is asserted seven 32.768 kHz clock cycles before the actual compensation interval completes so that back-to-back compensation can be enabled. 0b - Compensation busy or not enabled 1b - Compensation completed
10 —	Reserved
9 —	Reserved
8 BUS_ERR	Bus Error Indicates that a read or write cycle was initiated by software when the STATUS[INVAL_BIT] = 1. Write access to time or date registers is nullified (terminate normally) and no register value is changed. Reading when STATUS[INVAL_BIT] is asserted returns 0xFFFF. No transfer error is asserted. This bit is cleared by writing 1. 0b - Read and write accesses are normal. 1b - Read or write accesses occurred when STATUS[INVAL_BIT] was asserted.
7-6 WE	Write Enable Controls entry into and exit from the write protection mode. Both registers are protected by the write protection mechanism. These are self-clearing bits. Reads will return zeros. Disable write protection by writing this sequence: 00b, 01b, 11b, 10b. See Write protection mechanism for more information. **NOTE** When the registers are unlocked, they remain in this unlocked state for a time of 2 seconds, after which they are locked automatically. After power-on reset, the registers are unlocked and then are locked automatically after 15 seconds.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Enable Write Protection - Registers are locked.
5 CMP_INT	Compensation Interval Asserted for a time equal to compensation interval seconds as configured by the user. Used by the MCU to calculate the interrupts serviced during this interval and performs corrections in case of deviations (calibration). This bit toggles on at the start of every new compensation interval and is either 0 or 1 during the entire duration. This bit will not toggle if compensation logic has been disabled by MCU.
4 —	Reserved
3-2 —	Reserved
1 WRITE_PROT_EN	Write Protect Enable Status Indicates that registers are in locked mode and writing to them is disabled. Any write access made to the register space when write protection is enabled (that is, when they are in locked mode) will cause the transfer error signal to be asserted. 0b - Registers are unlocked and can be accessed. 1b - Registers are locked and in read-only mode.
0 INVAL_BIT	Invalidate CPU Read/Write Access Indicates the time and date counters are invalid or changing and therefore should not be read or written to. This bit is asserted for one oscillator clock cycle before and after the 1 Hz (seconds clock) boundary edge. Write access to time and date registers is nullified (terminate normally) and no register value is changed. Reading when STATUS[INVAL_BIT] is asserted returns 0xFFFF. No transfer error is asserted. 0b - Time and date counters can be read or written. Time and date is valid. 1b - Time and date counter values are changing or time and date is invalid and cannot be read or written.

36.5.1.12 Interrupt Status (ISR)

Offset

Register	Offset
ISR	14h

Function

NOTE

For the sampling timer interrupt status bits [14:6], see the chip configuration chapter for the applicable sampling timer frequencies.

Indicates the status of the various real-time clock interrupts. When an event of the types included in this register occurs, the bit will be set in this register regardless of the status of its corresponding interrupt enable bit. The status bits and interrupts are cleared by writing a value of 1. Interrupts may occur when the system clock is idle or in Standby mode. When the system enters the active power mode, an interrupt is indicated to the CPU. The first event of the Sampling Timer interrupts after power-on reset should not be used to qualify any periodic interval. However, the correct periodic interval (that is, 512 Hz or 256 Hz, and so on) should be determined using two sampling timer interrupts. The time between two interrupts should always be the correct time period.

This register is cleared on software reset.

NOTE

Sampling interrupts from 512 Hz to 2 Hz are generated using uncompensated clock. Only 1 Hz is compensated.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IS_512HZ	IS_256HZ	IS_128HZ	IS_64HZ	IS_32HZ	IS_16HZ	IS_8HZ	IS_4HZ	IS_2HZ	IS_1HZ	MIN_IS	HOUR_IS	DAY_IS	ALM_IS	0	0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 IS_512HZ	512 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
14 IS_256HZ	256 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
13 IS_128HZ	128 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
12 IS_64HZ	64 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
11 IS_32HZ	32 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
10 IS_16HZ	16 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 IS_8HZ	8 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
8 IS_4HZ	4 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
7 IS_2HZ	2 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
6 IS_1HZ	1 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
5 MIN_IS	Minutes Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
4 HOUR_IS	Hours Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
3 DAY_IS	Days Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
2 ALM_IS	Alarm Interrupt Status Indicates that the alarm value programmed matches the counter values. 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
1 —	Reserved
0 —	Reserved

36.5.1.13 Interrupt Enable (IER)

Offset

Register	Offset
IER	16h

Function

NOTE

For the sampling timer interrupt enable bits [14:6], see the chip configuration chapter for the applicable sampling timer frequencies.

Enables and disables the various real-time clock interrupts. De-asserting an interrupt enable bit has no effect on the assertion of its corresponding status bit.

Alarm interrupt is asserted on counters matching the alarm setting done in the memory map. The counters matched for the alarm interrupt are selected based on the alarm type set in CTRL[3:2]. The various types of alarms available are shown in the following table. Only one alarm type can be used at a time.

Table 284. Alarm Match Table

ALM_MATCH[1:0] (CTRL[3:2])	Counters Matched	Alarm Type
00	Seconds, Minutes, and Hours	Daily
01	Seconds, Minutes, Hours, and Days	Monthly
10	Seconds, Minutes, Hours, Days, and Months	Yearly
11	Seconds, Minutes, Hours, Days, Months, and Year	One Time

RTC generates a common interrupt. The user software should read the status register in the interrupt service routine to determine which interrupt has occurred.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IE_512HZ	IE_256HZ	IE_128HZ	IE_64HZ	IE_32HZ	IE_16HZ	IE_8HZ	IE_4HZ	IE_2HZ	IE_1HZ	MIN_IE	HOUR_IE	DAY_IE	ALM_IE	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 IE_512HZ	512 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
14 IE_256HZ	256 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
13 IE_128HZ	128 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
12 IE_64HZ	64 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
11 IE_32HZ	32 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
10 IE_16HZ	16 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
9 IE_8HZ	8 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
8 IE_4HZ	4 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
7 IE_2HZ	2 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
6 IE_1HZ	1 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
5 MIN_IE	Minutes Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
4 HOUR_IE	Hours Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Interrupt is disabled. 1b - Interrupt is enabled.
3 DAY_IE	Days Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
2 ALM_IE	Alarm Interrupt Enable Indicates that the alarm value programmed matches the counter values. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
1 —	Reserved
0 —	Reserved

36.5.1.14 Sub Second Counter (RTC_TEST2)

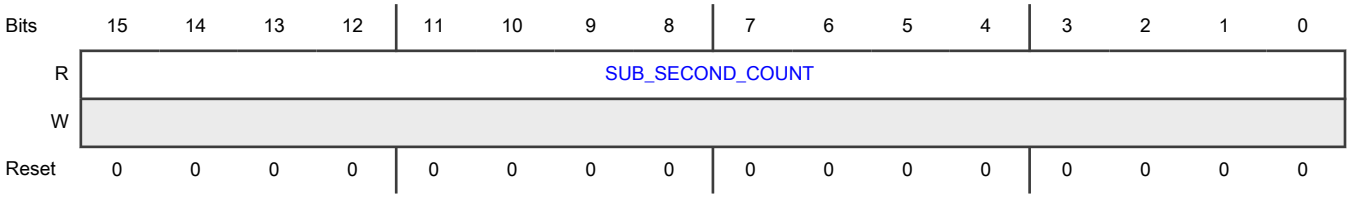
Offset

Register	Offset
RTC_TEST2	1Ch

Function

The RTC_TEST2 counter is used as a subsecond counter.

Diagram



Fields

Field	Function
15-0	Sub Second Counter Value

Table continues on the next page...

Field	Function
SUB_SECOND_COUNT	This counter is used as a sub second counter. Actually it counts the 32 kHZ clk cycles and auto clear every second.

36.5.1.15 Daylight Saving Hour (DST_HOUR)

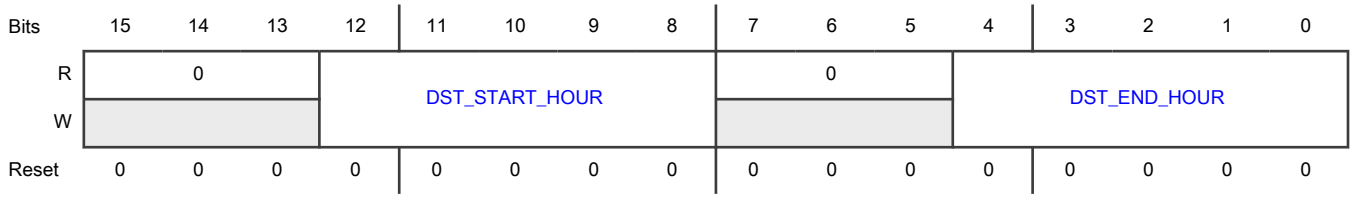
Offset

Register	Offset
DST_HOUR	22h

Function

Stores the time in hours when daylight saving has to be applied or reversed. This register is programmable when CTRL[DST_EN] = 0. When CTRL[DST_EN] = 1, the contents of this register cannot be changed. The user software should program the correct hour value (0-23) per the regional settings. For example, if daylight saving starts at 2:00 AM on March 25 and ends at 2:00 AM on October 28 in 2007 then the time at which the RTC advances or falls back is actually 1:59 AM. Therefore, the user software should program one, not two, for the hour count value (that is, write 0x0101) in this register. A 59-minute count is automatically checked inside RTC and not required to be programmed. This register has no effect on software reset.

Diagram



Fields

Field	Function
15-13 —	Reserved
12-8 DST_START_HOUR	Daylight Saving Time (DST) Hours Start Value The hour value for the time when DST comes into effect. Same as hours counter value in the Hours and Minutes Counters (HOURMIN).
7-5 —	Reserved
4-0 DST_END_HOUR	Daylight Saving Time (DST) Hours End Value The hour value for the time when DST is reversed. Same as hours counter value in the Hours and Minutes Counters (HOURMIN).

36.5.1.16 Daylight Saving Month (DST_MONTH)

Offset

Register	Offset
DST_MONTH	24h

Function

Stores the month when the daylight saving needs to be applied or reversed. This register is programmable when CTRL[DST_EN] = 0. When CTRL[DST_EN] = 1, the contents of this register cannot be changed. The CPU should program the correct month value (1-12) per the regional settings. For example, if daylight saving starts on March 25 and ends on October 28 in 2007, then CPU should write 0x030A in this register. This register has no effect on software reset.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DST_START_MONTH				0				DST_END_MONTH			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 —	Reserved
11-8 DST_START_MONTH	Daylight Saving Time (DST) Month Start Value The month value for the time when DST comes into effect. See Year and Month Counters (YEARMON).
7-4 —	Reserved
3-0 DST_END_MONTH	Daylight Saving Time (DST) Month End Value The month value for the time when DST is reversed. See Year and Month Counters (YEARMON).

36.5.1.17 Daylight Saving Day (DST_DAY)

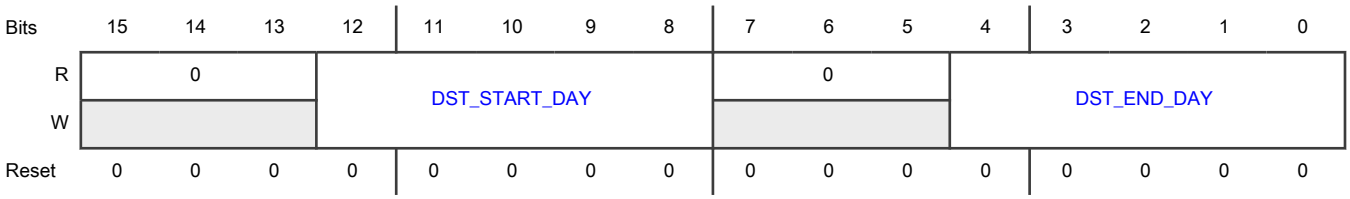
Offset

Register	Offset
DST_DAY	26h

Function

Stores the day when the daylight saving needs to be applied or reversed. This register is programmable when CTRL[DST_EN] = 0. When CTRL[DST_EN] = 1, the contents of this register cannot be changed. The CPU should program the correct day value (1-31) per the regional settings. For example, if the Daylight Saving starts at March 25 and ends at October 28 in 2007, then the CPU should write 0x191C in this register. This register is unaffected by software reset.

Diagram



Fields

Field	Function
15-13 —	Reserved
12-8 DST_START_DAY	Daylight Saving Time (DST) Day Start Value The day value for the time when DST comes into effect.
7-5 —	Reserved
4-0 DST_END_DAY	Daylight Saving Time (DST) Day End Value The day value for the time when DST is reversed.

36.5.1.18 Compensation (COMPEN)

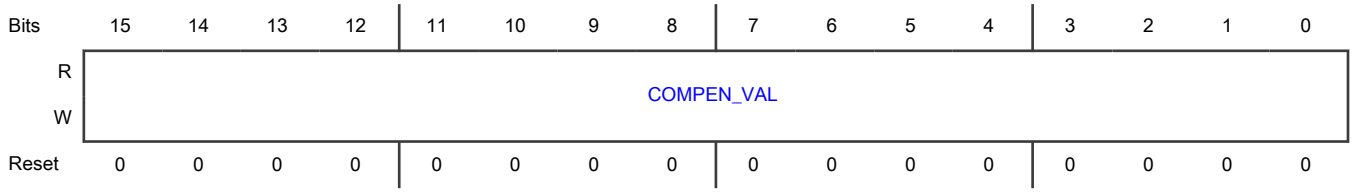
Offset

Register	Offset
COMPEN	28h

Function

Stores the compensation value to be used by the compensation block to correct the 1 Hz clock.

Diagram



Fields

Field	Function
15-0 COMPEN_VAL	Compensation Value Stores the compensation parameters. The definition of this field is dependent on the setting of CTRL[FINEEN]. If CTRL[FINEEN] = 0: • Compensation or Correction Value (COMPEN[7:0]) — Compensation or correction value is a two's complement value that modifies the 1 Hz clock (during its generation) by either adding or removing RTC oscillator clock cycles. • Compensation Interval (COMPEN[15:8]) — Compensation interval is the duration in seconds over which the correction is applied. This is the time in which the addition or removal of 32.768 kHz clock cycles occurs, thereby ensuring that the compensation interval is close to the interval obtained with an ideal 1 Hz clock. If CTRL[FINEEN] = 1 • Integral Compensation Value (COMPEN[15:12]) — This is a two's complement value of the integer part of correction or compensation value that has to be adjusted in every 1 second period. This value is expressed in terms of number of clock cycles of the RTC oscillator clock. • COMPEN[11:7] — Should be zero • Fraction Compensation Value (COMPEN[6:0]) — This is the fractional part of the correction or compensation value that has to be adjusted. This value is expressed as number of clock cycles of a fixed IRC_CLK. This value is always a positive number.

36.5.2 RTC register descriptions

These registers are read only in the target RTC and read the values from the controller RTC registers: YEARMON, DAYS, HOURMIN, SECONDS, DST_HOUR, DST_MONTH, DST_DAY, and COMPEN.

NOTE

Use valid dates to configure registers. For example, do not configure a value like February 30, 2024, because there is no 30th in February 2024.

36.5.2.1 RTC memory map

RTC1 base address: 4006_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Year and Month Counters (YEARMON)	16	R	0001h
2h	Days and Day-of-Week Counters (DAYS)	16	R	0001h
4h	Hours and Minutes Counters (HOURMIN)	16	R	0000h
6h	Seconds Counters (SECONDS)	16	R	0000h
8h	Year and Months Alarm (ALM_YEARMON)	16	RW	0000h
Ah	Days Alarm (ALM_DAYS)	16	RW	0000h
Ch	Hours and Minutes Alarm (ALM_HOURMIN)	16	RW	0000h
Eh	Seconds Alarm (ALM_SECONDS)	16	RW	0000h
10h	Control (CTRL)	16	RW	0200h
12h	Status (STATUS)	16	RW	0000h
14h	Interrupt Status (ISR)	16	RW	0000h
16h	Interrupt Enable (IER)	16	RW	0000h
1Ch	Sub Second Counter (RTC_TEST2)	16	R	0000h
22h	Daylight Saving Hour (DST_HOUR)	16	R	0000h
24h	Daylight Saving Month (DST_MONTH)	16	R	0000h
26h	Daylight Saving Day (DST_DAY)	16	R	0000h
28h	Compensation (COMPEN)	16	R	0000h

36.5.2.2 Year and Month Counters (YEARMON)

Offset

Register	Offset
YEARMON	0h

Function

The function of YEARMON on the target RTC is the same as on the controller RTC YEARMON register, from where the value of this register is directly read.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	YROFST								0				MON_CNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
15-8 YROFST	Year Offset Count Value Indicates the offset in years from the base year (hard coded as 2112) and does not show the actual year value. This is a signed value. Valid values are -128 to 127. The base year is 2112 and if the value of YEAR field is 0x10, the actual year is 2112 + 16 = 2128.
7-4 —	Reserved
3-0 MON_CNT	Month Counter Provides the value of the Months Counter. Valid Values are: 0000b,1101b,1110b,1111b - Illegal Value 0001b - January 0010b - February 0011b - March 0100b - April 0101b - May 0110b - June 0111b - July 1000b - August 1001b - September 1010b - October 1011b - November 1100b - December

36.5.2.3 Days and Day-of-Week Counters (DAYS)

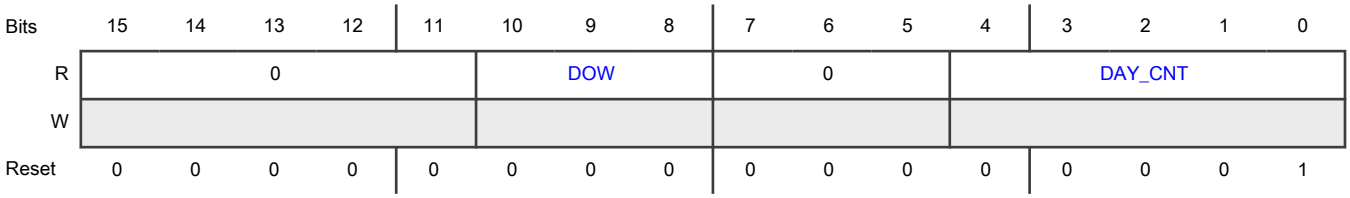
Offset

Register	Offset
DAYS	2h

Function

The function of DAYS on the target RTC is the same as on the controller DAYS register, from where the value of this register is directly read.

Diagram



Fields

Field	Function
15-11 —	Reserved
10-8 DOW	Day of Week Counter Value 000b - Sunday 001b - Monday 010b - Tuesday 011b - Wednesday 100b - Thursday 101b - Friday 110b - Saturday 111b - Reserved
7-5 —	Reserved
4-0 DAY_CNT	Days Counter Value Valid values are 1 to 31.

36.5.2.4 Hours and Minutes Counters (HOURMIN)

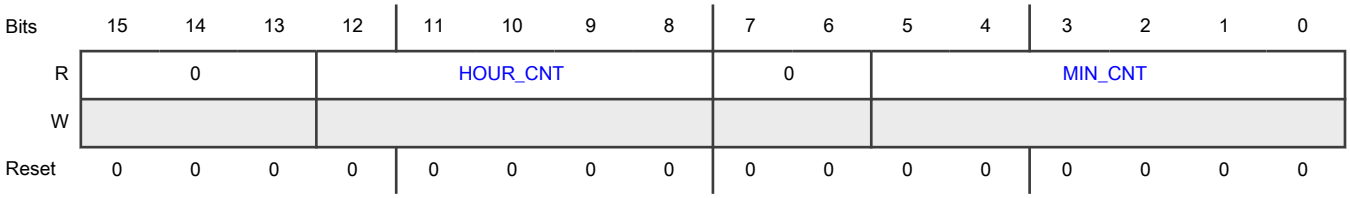
Offset

Register	Offset
HOURMIN	4h

Function

The function of HOURMIN on the target RTC is the same as on the controller RTC HOURMIN register, from where the value of this register is directly read.

Diagram



Fields

Field	Function
15-13 —	Reserved
12-8 HOUR_CNT	Hours Counter Value Valid count values are 0 to 23.
7-6 —	Reserved
5-0 MIN_CNT	Minutes Counter Value Valid count values are 0 to 59.

36.5.2.5 Seconds Counters (SECONDS)

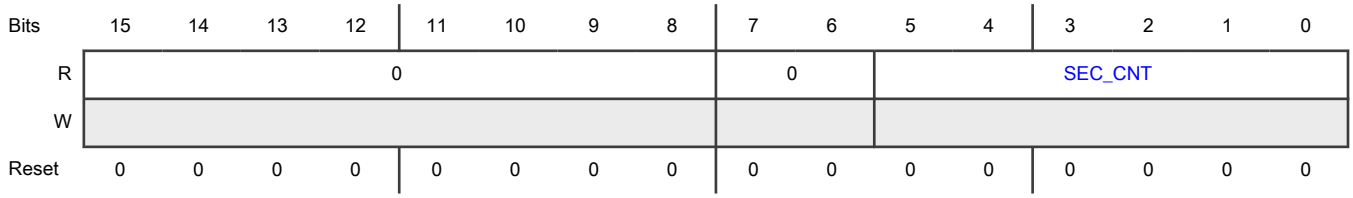
Offset

Register	Offset
SECONDS	6h

Function

The function of SECONDS on the target RTC is the same as on the controller RTC SECONDS register, from where the value of this register is directly read.

Diagram



Fields

Field	Function
15-8 —	Reserved
7-6 —	Reserved
5-0 SEC_CNT	Seconds Counter Value Valid count values are 0 to 59.

36.5.2.6 Year and Months Alarm (ALM_YEARMON)

Offset

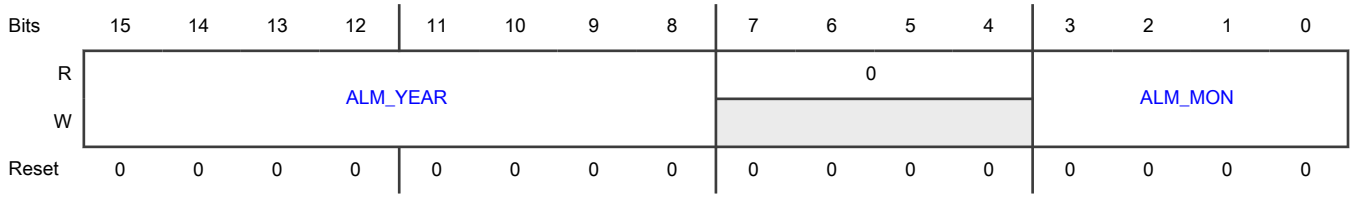
Register	Offset
ALM_YEARMON	8h

Function

Used to configure the months and year setting of the alarm. The alarm setting can be read or written anytime. Alarm interrupt bit is set when all values of alarm seconds, minutes, hours, days, month, and year match their respective counter values. This register is reset to its default state on software reset.

User software can configure the alarm type by using the CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-8 ALM_YEAR	Year Value for Alarm Same as years offset value in Year and Month Counters (YEARMON) .
7-4 —	Reserved
3-0 ALM_MON	Months Value for Alarm Same as months counter value in Year and Month Counters (YEARMON).

36.5.2.7 Days Alarm (ALM_DAYS)

Offset

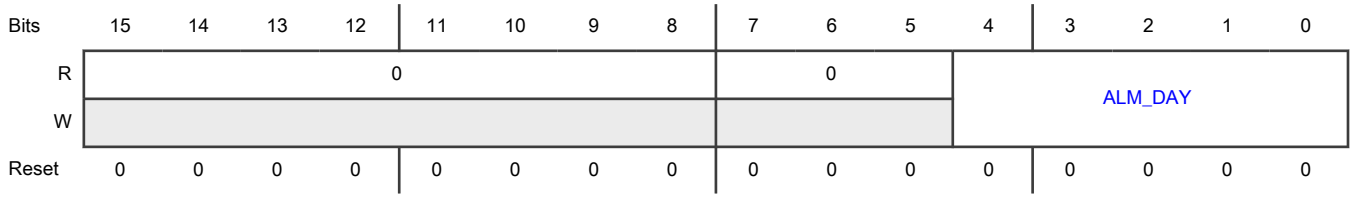
Register	Offset
ALM_DAYS	Ah

Function

Used to configure the day setting of the alarm. The alarm setting can be read or written anytime. Alarm interrupt bit is set when all values of alarm seconds, minutes, hours, days, month, and year match their respective counter values. This register is reset to its default state on software reset.

User software can configure the alarm type using CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-8 —	Reserved
7-5 —	Reserved
4-0 ALM_DAY	Days Value for Alarm Same as days counter value in Days and Day-of-Week Counters (DAYS) .

36.5.2.8 Hours and Minutes Alarm (ALM_HOURMIN)

Offset

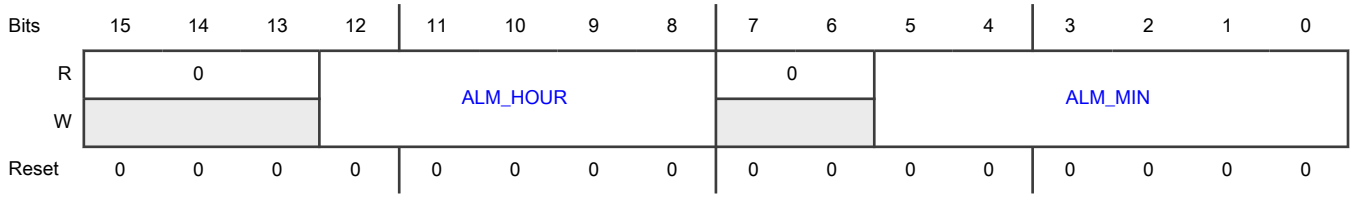
Register	Offset
ALM_HOURMIN	Ch

Function

Used to configure the hour and minute setting of the alarm. The alarm setting can be read or written anytime. This register is reset to default state on software reset.

User software can configure alarm type using CTRL[ALM_MATCH].

Diagram



Fields

Field	Function
15-13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
12-8 ALM_HOUR	Hours Value for Alarm Same as hours counter value in Hours and Minutes Counters (HOURMIN) .
7-6 —	Reserved
5-0 ALM_MIN	Minutes Value for Alarm Same as minutes counter value in Hours and Minutes Counters (HOURMIN) .

36.5.2.9 Seconds Alarm (ALM_SECONDS)

Offset

Register	Offset
ALM_SECONDS	Eh

Function

Used to configure the seconds setting of the alarm. The alarm setting can be read or written anytime. This register is reset to default value on software reset.

User software has the option of using bits 9:8 to perform a correction on the seconds counter to compensate for leap seconds. Writing to these bits adds or subtracts 1 from the seconds counter and read returns zeros.

User software should first determine the state of STATUS[INVAL_BIT] to determine if the counters are stable before they can be incremented or decremented. The assertion of STATUS[INVAL_BIT] ensures that no operation is done at the boundary of a second when counters change value.

User software can configure alarm type using CTRL[ALM_MATCH].

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						0	0	0				ALM_SEC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
9 —	Reserved
8 —	Reserved
7-6 —	Reserved
5-0 ALM_SEC	Seconds Alarm Value Same as seconds counter value in Seconds Counters (SECONDS) .

36.5.2.10 Control (CTRL)

Offset

Register	Offset
CTRL	10h

Function

Governs all operations occurring inside the RTC. Used to specify the software reset, daylight controls, and the type of alarm function needed.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CLKOUT		0		CLKO_DIS	1	0	0		0	Reserved	ALM_MATCH			
W								SWR		0					0	0
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 —	Reserved
14-13	RTC Clock Output Selection Selects which clock to output from SoC for use outside RTC.

Table continues on the next page...

Table continued from the previous page...

Field	Function
CLKOUT	00b - No output clock 01b - Fine 1 Hz clock with both precise edges 10b - 32.768 kHz clock 11b - Coarse 1 Hz clock with both precise edges
12-11 —	Reserved
10 CLKO_DIS	Clock Output Disable Determines whether the selected clock is output to other peripherals. 0b - The selected clock is output to other peripherals. 1b - The selected clock is not output to other peripherals.
9 —	Reserved
8 SWR	Software Reset Self clearing bit. Asserting this field clears the contents of alarm, interrupt (status and enable) registers, STATUS[CMP_DONE], and STATUS[BUS_ERR], and has no effect on DST, calendaring, time registers. 0b - Software Reset cleared 1b - Software Reset asserted
7 —	Reserved
6 —	Value must be zero.
5 —	Reserved
4 —	Reserved
3-2 ALM_MATCH	Alarm Match Defines the type of alarm function. Selects which time and calendar counters are used for matching and will generate an alarm. 00b - Only seconds, minutes, and hours matched. 01b - Only seconds, minutes, hours, and days matched. 10b - Only seconds, minutes, hours, days, and months matched.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Only seconds, minutes, hours, days, months, and year (offset) matched.
1 —	Value must be zero.
0 —	Value must be zero.

36.5.2.11 Status (STATUS)

Offset

Register	Offset
STATUS	12h

Function

Indicates the status of various processes occurring inside the RTC. Also helps the user software to read time or date register when the values are stable and not changing. Compensation Done bit and Bus Error bit are cleared by writing 1. Software reset resets the whole register to its default state.

All memory mapped registers are protected against spurious updates by the write protect mechanism. To unlock the registers, a specific pattern (as mentioned in the table above) has to be written in the write enable bits (WE[1:0]) to enable or disable write protection. The WE[1:0] bits are the only bits that are freely writeable by user software. The write enable bits are self-clearing bits that always return zeros on read.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					0	0	BUS_ERR	0			0	0		WRITE_P...	INVAL_B...
W					0			W1C	WE		0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 —	Reserved
11 —	Value must be zero.

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 —	Reserved
9 —	Reserved
8 BUS_ERR	Bus Error Indicates that a read or write cycle was initiated by software when the STATUS[INVAL_BIT] = 1. Write access to time or date registers is nullified (terminate normally) and no register value is changed. Reading when STATUS[INVAL_BIT] is asserted returns 0xFFFF. No transfer error is asserted. This bit is cleared by writing 1. 0b - Read and write accesses are normal. 1b - Read or write accesses occurred when STATUS[INVAL_BIT] was asserted.
7-6 WE	Write Enable Controls entry into and exit from the write protection mode. Both registers are protected by the write protection mechanism. These are self-clearing bits. Reads will return zeros. Disable write protection by writing this sequence: 00b, 01b, 11b, 10b. See Write protection mechanism for more information. **NOTE** When the registers are unlocked, they remain in this unlocked state for a time of 2 seconds, after which they are locked automatically. After power-on reset, the registers are unlocked and then are locked automatically after 15 seconds. 10b - Enable Write Protection - Registers are locked.
5 —	Value must be zero.
4 —	Reserved
3-2 —	Reserved
1 WRITE_PROT_EN	Write Protect Enable Status Indicates that registers are in locked mode and writing to them is disabled. Any write access made to the register space when write protection is enabled (that is, when they are in locked mode) will cause the transfer error signal to be asserted. 0b - Registers are unlocked and can be accessed. 1b - Registers are locked and in read-only mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 INVAL_BIT	Invalidate CPU Read/Write Access Indicates the time and date counters are invalid or changing and therefore should not be read or written to. This bit is asserted for one oscillator clock cycle before and after the 1 Hz (seconds clock) boundary edge. Write access to time and date registers is nullified (terminate normally) and no register value is changed. Reading when STATUS[INVAL_BIT] is asserted returns 0xFFFF. No transfer error is asserted. 0b - Time and date counters can be read or written. Time and date is valid. 1b - Time and date counter values are changing or time and date is invalid and cannot be read or written.

36.5.2.12 Interrupt Status (ISR)

Offset

Register	Offset
ISR	14h

Function

NOTE

For the sampling timer interrupt status bits [14:6], see the chip configuration chapter for the applicable sampling timer frequencies.

Indicates the status of the various real-time clock interrupts. When an event of the types included in this register occurs, the bit will be set in this register regardless of the status of its corresponding interrupt enable bit. The status bits and interrupts are cleared by writing a value of 1. Interrupts may occur when the system clock is idle or in Standby mode. When the system enters the active power mode, an interrupt is indicated to the CPU. The first event of the Sampling Timer interrupts after power-on reset should not be used to qualify any periodic interval. However, the correct periodic interval (that is, 512 Hz or 256 Hz, and so on) should be determined using two sampling timer interrupts. The time between two interrupts should always be the correct time period.

This register is cleared on software reset.

NOTE

Sampling interrupts from 512 Hz to 2 Hz are generated using uncompensated clock. Only 1 Hz is compensated.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IS_512HZ	IS_256HZ	IS_128HZ	IS_64HZ	IS_32HZ	IS_16HZ	IS_8HZ	IS_4HZ	IS_2HZ	IS_1HZ	MIN_IS	HOUR_IS	DAY_IS	ALM_IS	0	0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 IS_512HZ	512 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
14 IS_256HZ	256 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
13 IS_128HZ	128 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
12 IS_64HZ	64 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
11 IS_32HZ	32 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
10 IS_16HZ	16 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
9 IS_8HZ	8 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
8 IS_4HZ	4 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
7 IS_2HZ	2 Hz Interval Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
6 IS_1HZ	1 Hz Interval Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
5 MIN_IS	Minutes Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
4 HOUR_IS	Hours Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
3 DAY_IS	Days Interrupt Status 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
2 ALM_IS	Alarm Interrupt Status Indicates that the alarm value programmed matches the counter values. 0b - Interrupt is de-asserted. 1b - Interrupt is asserted.
1 —	Reserved
0 —	Reserved

36.5.2.13 Interrupt Enable (IER)

Offset

Register	Offset
IER	16h

Function

NOTE

For the sampling timer interrupt enable bits [14:6], see the chip configuration chapter for the applicable sampling timer frequencies.

Enables and disables the various real-time clock interrupts. De-asserting an interrupt enable bit has no effect on the assertion of its corresponding status bit.

Alarm interrupt is asserted on counters matching the alarm setting done in the memory map. The counters matched for the alarm interrupt are selected based on the alarm type set in CTRL[3:2]. The various types of alarms available are shown in the following table. Only one alarm type can be used at a time.

Table 285. Alarm Match Table

ALM_MATCH[1:0] (CTRL[3:2])	Counters Matched	Alarm Type
00	Seconds, Minutes, and Hours	Daily
01	Seconds, Minutes, Hours, and Days	Monthly
10	Seconds, Minutes, Hours, Days, and Months	Yearly
11	Seconds, Minutes, Hours, Days, Months, and Year	One Time

RTC generates a common interrupt. The user software should read the status register in the interrupt service routine to determine which interrupt has occurred.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IE_512HZ	IE_256HZ	IE_128HZ	IE_64HZ	IE_32HZ	IE_16HZ	IE_8HZ	IE_4HZ	IE_2HZ	IE_1HZ	MIN_IE	HOUR_IE	DAY_IE	ALM_IE	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 IE_512HZ	512 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
14 IE_256HZ	256 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
13 IE_128HZ	128 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
12 IE_64HZ	64 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 IE_32HZ	32 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
10 IE_16HZ	16 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
9 IE_8HZ	8 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
8 IE_4HZ	4 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
7 IE_2HZ	2 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
6 IE_1HZ	1 Hz Interval Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
5 MIN_IE	Minutes Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
4 HOUR_IE	Hours Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
3 DAY_IE	Days Interrupt Enable 0b - Interrupt is disabled. 1b - Interrupt is enabled.
2 ALM_IE	Alarm Interrupt Enable Indicates that the alarm value programmed matches the counter values. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 —	Reserved

36.5.2.14 Sub Second Counter (RTC_TEST2)

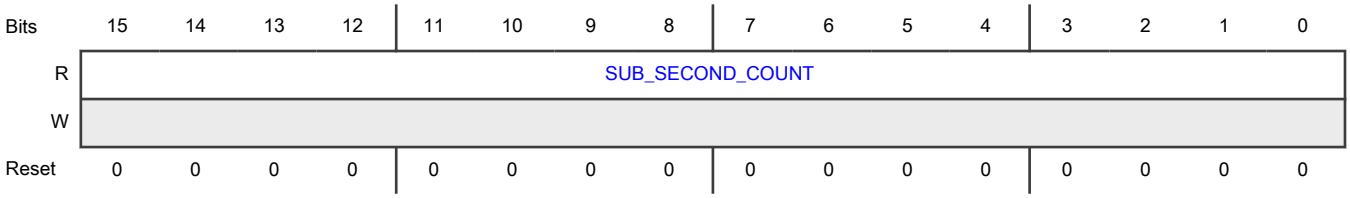
Offset

Register	Offset
RTC_TEST2	1Ch

Function

The value is read from the controller RTC RTC_TEST2 register.

Diagram



Fields

Field	Function
15-0 SUB_SECOND_COUNT	Sub Second Counter Value The value is read from the controller RTC RTC_TEST2[SUB_SECOND_COUNT] register.

36.5.2.15 Daylight Saving Hour (DST_HOUR)

Offset

Register	Offset
DST_HOUR	22h

Function

The function of DST_HOUR on the target RTC is the same as on the controller RTC DST_HOUR register, from where the value of this register is directly read. .

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DST_START_HOUR				0				DST_END_HOUR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-13 —	Reserved
12-8 DST_START_HOUR	Daylight Saving Time (DST) Hours Start Value The hour value for the time when DST comes into effect. Same as hours counter value in the Hours and Minutes Counters (HOURMIN).
7-5 —	Reserved
4-0 DST_END_HOUR	Daylight Saving Time (DST) Hours End Value The hour value for the time when DST is reversed. Same as hours counter value in the Hours and Minutes Counters (HOURMIN).

36.5.2.16 Daylight Saving Month (DST_MONTH)

Offset

Register	Offset
DST_MONTH	24h

Function

The function of DST_MONTH on the target RTC is the same as on the controller RTC DST_MONTH register, from where the value of this register is directly read.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DST_START_MONTH				0				DST_END_MONTH			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 —	Reserved
11-8 DST_START_MONTH	Daylight Saving Time (DST) Month Start Value The month value for the time when DST comes into effect. See Year and Month Counters (YEARMON).
7-4 —	Reserved
3-0 DST_END_MONTH	Daylight Saving Time (DST) Month End Value The month value for the time when DST is reversed. See Year and Month Counters (YEARMON).

36.5.2.17 Daylight Saving Day (DST_DAY)

Offset

Register	Offset
DST_DAY	26h

Function

The function of DST_DAY on the target RTC is the same as on the controller RTC DST_DAY register, from where the value of this register is directly read.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DST_START_DAY				0				DST_END_DAY			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-13 —	Reserved
12-8 DST_START_DAY	Daylight Saving Time (DST) Day Start Value The day value for the time when DST comes into effect.
7-5 —	Reserved
4-0 DST_END_DAY	Daylight Saving Time (DST) Day End Value The day value for the time when DST is reversed.

36.5.2.18 Compensation (COMPEN)

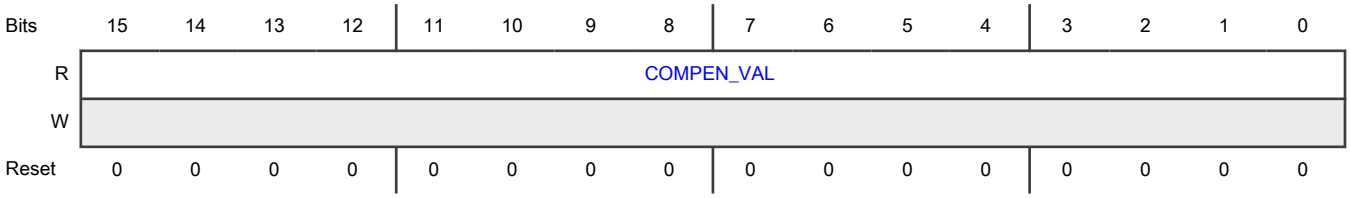
Offset

Register	Offset
COMPEN	28h

Function

Stores the compensation value to be used by the compensation block to correct the 1 Hz clock.

Diagram



Fields

Field	Function
15-0 COMPEN_VAL	Compensation Value Stores the compensation parameters.

Chapter 37
Process, Voltage and Temperature Sensor (PVTS)

37.1 Chip-specific PVTS information

Table 286. Reference links to related information

Topic	Related module	Reference
Full description	PVTS	PVTS
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
System interrupts		System interrupts

37.1.1 Module instances

This device has two instances of the PVTS module, PVTS0 and PVTS1.

Table 287. PVTS instances

	PVTS0	PVTS1
Located in	VDD2_COM	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Sensor 0 monitor	CPU0	CPU1
Sensor 1 monitor	HiFi4	HiFi1
main_clk0	COMPUTE_MAIN_CLK	SENSE_MAIN_CLK
main_clk1	DSP_CLK	SENSE_DSP_CLK
peripheral_clk	COMPUTE_MAIN_CLK	SENSE_MAIN_CLK
PVTS interrupt 0	PVTS0 CPU0 interrupt	PVTS1 CPU1 interrupt
PVTS interrupt 1	PVTS0 HiFi4 DSP core interrupt	PVTS1 HiFi1 DSP core interrupt

37.1.2 PVTS fuse programming

The appropriate PVTS delay values to use for a given device are programmed to fuses during chip manufacturing. The appropriate fuse value corresponding to the selected system configuration should be read from fuses and is then used to write the appropriate DEL_CTRLn register.

Table 288. PVTS fuses for RT700

Fuse word index	Fuse word name	Functions
482	SENSE_PVT_SP3	250 MHz PVTS delay values, HiFi1[15:8], CPU1[7:0]
483	COM_PVT_SP3	325 MHz PVTS delay values, HiFi4[15:8], CPU0[7:0]
484	SENSE_PVT_SP2	160 MHz PVTS delay values, HiFi1[15:8], CPU1[7:0]
485	COM_PVT_SP2	192 MHz PVTS delay values, HiFi4[15:8], CPU0[7:0]
486	SENSE_PVT_SP1	100 MHz PVTS delay values, HiFi1[15:8], CPU1[7:0]
487	COM_PVT_SP1	110 MHz PVTS delay values, HiFi4[15:8], CPU0[7:0]

37.1.3 Optimizing PVTS delay with temperature

The PVTS delay value retrieved from fuses is a safe value that works for the whole temperature range. To achieve greater power savings at warmer temperatures between 30 °C and 85 °C, you can subtract the PVTS temperature deviation shown in the table below from the delay value in the fuses before programming it to the PVTS sensor.

Table 289. PVTS temperature deviation

Domain	Typical voltage (V)	Frequency (MHz)	PVTS temperature deviation	
			-30 °C - 30 °C	30 °C - 85 °C
Compute (CPU0, HiFi4)	1.1	325	0	1
	0.9	192	0	1
	0.8	60	0	1
Sense (CPU1, HiFi1)	1.1	250	0	0
	0.9	160	0	0
	0.8	60	0	1

Updating the PVTS delay with this additional optimization requires that the application monitors the temperature. Some applications may find that monitoring the temperature and changing the PVTS delay is not worth doing. Those applications can use the fixed PVTS delay value with no optimization.

37.2 Overview

The PVTS is a peripheral that monitors the timing margin of one or more main_clk signals within a given power domain at runtime. Refer to the chip-specific section for details on which clocks are monitored. While using the PVTS, the minimum voltage for the domain can potentially be decreased below the normal datasheet specification. PVTS triggers an interrupt when this timing margin is not met, which indicates the power rail voltage for the domain must be increased to meet the required timing.

The PVTS helps reduce voltage and power in active modes, including the Sleep mode. The PVTS is not intended for use in low power modes.

37.2.1 Block diagram

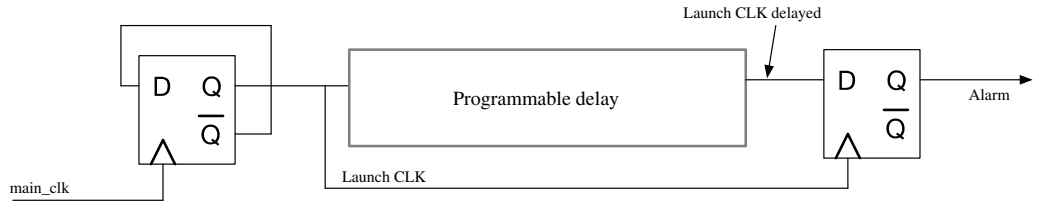


Figure 111. Block diagram

37.2.2 Features

PVTS features:

- Provides 2 sets of sensors to monitor 2 CPUs in the same power domain.
- Each sensor supports up to 62 delay cycles.

37.3 Functional description

37.3.1 PVTS theory of operation

The internal timing of a semiconductor is dependent on the process, voltage, and temperature (PVT) variation. The datasheet specifies the minimum voltage requirements for given operating conditions; however, the datasheet minimum voltage is set to guarantee operation of all devices with a wide range of PVT variations and cover worst case conditions. Most devices and conditions are actually capable of running below the specified minimum voltage.

The PVTS module allows for monitoring unique device frequency behavior taking frequency changes into account to find the actual ideal minimum voltage. This allows for running the device below the specified minimum voltage for given operating conditions and thereby optimizing the power consumption.

37.3.2 Monitor timing margin

The PVTS monitors the timing margin of main_clk using a delay line, see Figure 111. main_clk is divided by 2 and generates the Launch CLK. The Launch CLK is compared to a delayed version of itself. This delay is programmable and must be set to a value that correlates to the speed of that unique device based on process corners for the N-channel and P-channel MOSFETs. There are three primary corners: typical, fast, and slow. With this naming convention each device is described as typical-typical (TT), fast-fast (FF), slow-slow (SS). The timing comparison is done through a flip-flop that uses Launch CLK to clock the delayed version of itself. If the added delay is too long, the flip-flop output is high, indicating a timing issue and triggering an alarm.

For the PVTS to properly monitor the main_clk timing, the delay line must be programmed with a delay that matches the worst-case timing path of the logic clocked by main_clk. See chip-specific section for information on finding the value to program the correct delay value. Once programmed with this delay, the PVTS emulates this worst-case timing path in the domain logic. This delay tracks the actual timing required by this unique device across temperature and voltage variations. As the voltage decreases or as the temperature gets colder, the delayed Launch CLK has a longer delay. When the alarm signal triggers, the application increases the domain voltage, which increases the performance of the logic, which decreases the delay on Launch CLK.

Figure 112 shows the timing of the PVT Delay Line. The status of delayed Launch CLK is sampled on every rising edge of Launch CLK. When operating above the minimum voltage needed for that device and current temperature, the delay is short and the delayed Launch CLK is low at rising edge of Launch CLK, so the alarm output remains low. However, when the domain voltage is too low for current conditions, the delay is long enough that the delayed Launch CLK is still high at the rising edge of Launch CLK. The alarm output is high, triggering an interrupt to indicate that higher domain voltage is needed to reduce the timing delay.

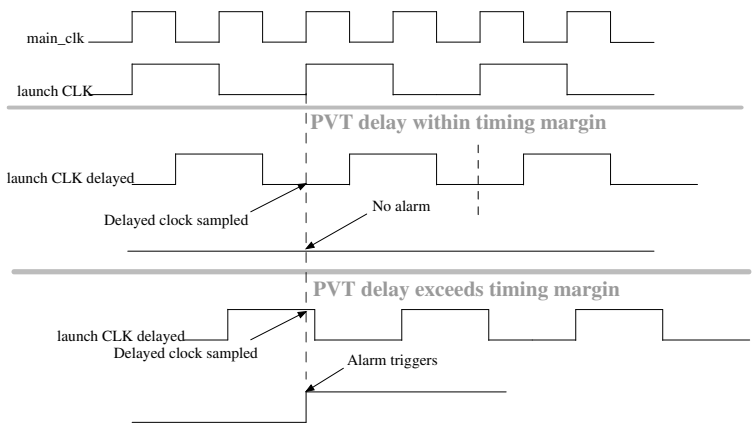


Figure 112. PVT delay line timing with alarm trigger

37.3.3 PVTS considerations with low-power modes

Because PVTS cannot monitor the timing in modes where the main_clk is not running, care must be taken when using PVTS in combination with low-power modes.

Consider a situation where the device is using PVTS and is running at a reduced voltage and then enters a low power mode that disables PVTS. If the device is in the low-power mode for an extended time the temperature could decrease. Now, when the device wakes up, the PVTS reduced voltage might be too low for operation at the new temperature. The PVTS will trigger the alarm after the wakeup, but at the lower voltage the device might not be running properly or capable of requesting the higher voltage.

To avoid conditions like those described above, the recommendation is to ensure that on low-power wakeup the system will always return to the normal minimum voltage specified in data sheet. Depending on the specifics of the power system, this might require raising the voltage before entering the low-power mode. After the device wakes up at the safe voltage, the PVTS can resume monitoring and the voltage can be reduced to the optimal level again.

NOTE

Low-power modes that wake through a reset should return the device to reset/start-up voltage conditions. This should already be accounted for in the system power design/configuration, and therefore special consideration of PVTS before entering these modes is not typically required.

37.3.4 Requirements for using PVTS

Proper usage of the PVTS allows an application to operate the voltage below the data sheet minimum specification. It can lead to issues or faults in the application if the following requirements are not met:

- The minimum voltage required is not always limited by the frequency of the main_clk. The data sheet may also have restrictions on voltage for specific peripherals, features, or use-cases of the MCU. The PVT Sensor is not designed to monitor voltage for other data sheet restrictions. Refer to the device datasheet for other requirements.

  If there are multiple minimum voltage requirements in the datasheet that apply, the highest minimum must be used, regardless of what the PVTS reports. For example, reading OTP fuses might have a minimum voltage specification that should be used. This is important to keep in mind when reading the PVTS delay values from fuses (refer to chip-specific section for details on fuses that store PVTS delay values).

- There are also requirements for interrupts in an application using PVTS. When a higher voltage is required, the PVTS triggers an alarm interrupt, but it is the application that must request the higher voltage from the PMIC. Therefore, best practices must be used to manage interrupts. The PVTS IRQ must be serviced quickly without much latency, and it must not be blocked for long.

- The settling time for the voltage power rail must also be taken into account. Any time the application is using PVTS and reduces the voltage, it must wait long enough for the PMIC to change the voltage, and the supply rail to settle to the new voltage. The exact delay can depend on the board design, like the PMIC/regulator driving the voltage and the capacitance on

this supply rail. If the application drops the voltage too quickly before it has settled, PVTS may not be able to detect a timing issue before the voltage is dropped again, which can lead to a fault occurring.

37.3.5 Clocking

Table 290. PVTS clocks

Clock	Description
peripheral_clk	Used to write and read registers of this module.
main_clk0 main_clk1	CPU clocks the PVTS monitors.

See chip-specific section for more information.

37.3.6 Interrupts

PVTS can trigger the following interrupts, see chip-specific information for more details:

- PVTS interrupt 0
- PVTS interrupt 1

NOTE

These interrupts are always enabled.

37.4 Application information

When the application enables the PVTS, the first goal is to find the lowest voltage for the current conditions. The PVTS alarm indicates when higher voltage is required, but PVTS does not give any indication of when voltage can be further reduced. The application can test lower voltages using this procedure. See Figure 113 for an example of this reduction method.

1. Reduce the voltage by one voltage step
2. Wait for the voltage to settle
3. If no PVTS interrupt is triggered, go back to #1.
4. If PVTS interrupt is triggered, then increase the voltage by one voltage step. This is the optimal/lowest voltage for the current configuration and conditions.

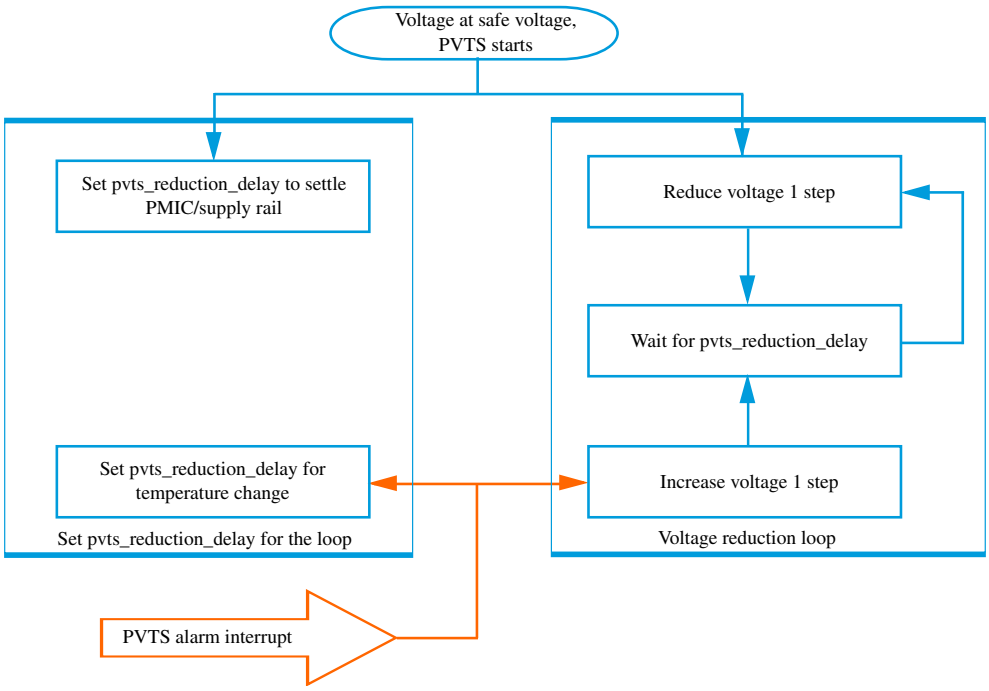


Figure 113. Voltage reduction flowchart with PVTS

NOTE

There may be additional voltage constraints that prevent decreasing the voltage to the point where the PVTS interrupt is triggered. See Requirements for using PVTS.

NOTE

The total wait time for steps one and two should take into account the total time needed to request the voltage decrease, and then for the voltage to settle at the requested value. If an external PMIC is used, then time to control the PMIC should be taken into account.

The next stage for the application is to wait for PVTS alarms indicating the voltage needs to be increased, and also check if the temperature has changed enough that a lower voltage is now acceptable. This can be done by periodically reducing the voltage.

Figure 114 shows an example of voltage dynamically changing as temperature changes. This chart is an example where the minimum voltage specification in the datasheet for the frequency used is 900 mV. This graph starts where the voltage has already been reduced to 850 mV using the procedure for finding the lowest voltage described above and is operating at 25 ℃. In this example, the voltage source has a 25 mV voltage step. While the temperature drops, the PVTS Delay Line gets slower, and at some point the PVTS alarm triggers. The application requests the voltage to increase one step to 875 mV. The temperature continues to drop, so another PVTS alarm triggers, and the voltage is increased to 900 mV. The application tests periodically (for example, every 10 seconds) if a reduced voltage is acceptable, by reducing the voltage one step. Typically, reducing the voltage one step immediately triggers the PVTS interrupt, and the voltage is increased one step back to the previous level. As the temperature warms, at some point the reduced voltage test does not trigger an alarm, and PVTS allows the voltage to reduce one step to 875 mV. As the temperature continues to warm, this example eventually drops down to 800 mV. Then as the temperature then cools back to 25 ℃, PVTS triggers its alarm that voltage must be increased, the voltage is returned to 850 mV.

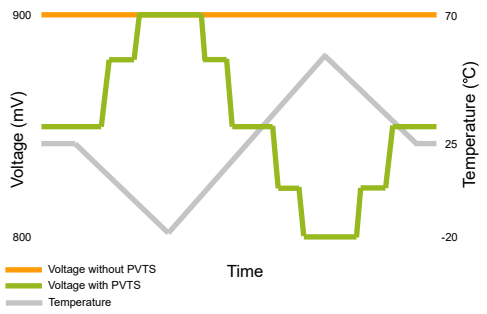


Figure 114. Dynamic voltage at runtime

37.5 Memory map and registers

37.5.1 PVTS register descriptions

37.5.1.1 PVTS memory map

PVTS0 base address: 4000_5000h

PVTS1 base address: 4004_3000h

Offset	Register	Width (In bits)	Access	Reset value
20h	Delay Control (DEL_CTRL0)	32	RW	0000_0000h
2Ch	Alert Counter (ALERT_CNT0)	32	RW	0000_0000h
30h	Alert Counter Control (ALERT_CNT_CTRL0)	32	RW	0000_0000h
A0h	Delay Control (DEL_CTRL1)	32	RW	0000_0000h
ACh	Alert Counter (ALERT_CNT1)	32	RW	0000_0000h
B0h	Alert Counter Control (ALERT_CNT_CTRL1)	32	RW	0000_0000h

37.5.1.2 Delay Control (DEL_CTRL0 - DEL_CTRL1)

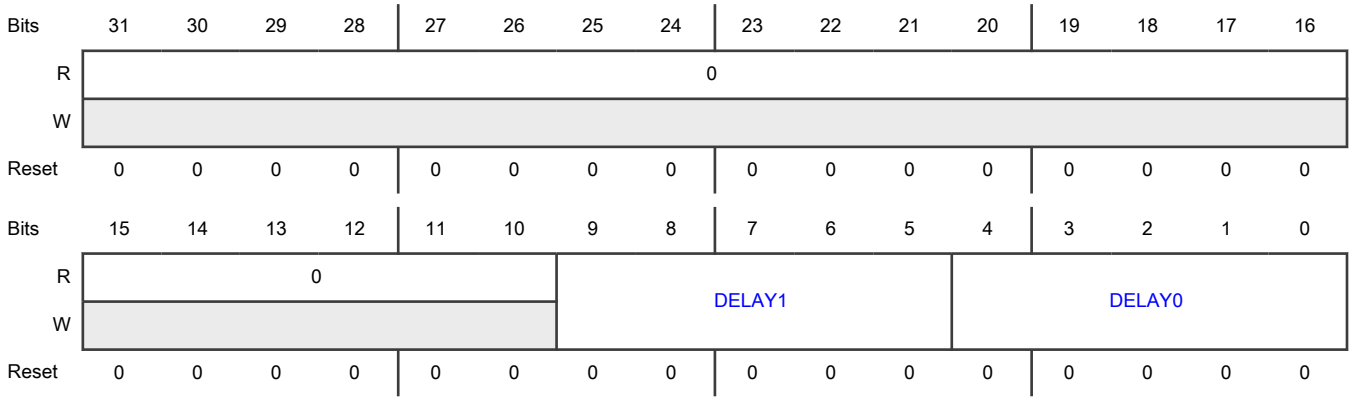
Offset

Register	Offset
DEL_CTRL0	20h
DEL_CTRL1	A0h

Function

Controls the 5 selectable delay elements in each delay section. The total delay is DELAY0 + DELAY1. Setting the total DELAY fields to 0 will stop the PVT Sensor. The sensor will start when the DELAY fields are written with any non-zero value.

Diagram



Fields

Field	Function
31-10 —	Reserved
9-5 DELAY1	Delay Section 1 Configure the delay of section 1. Each bit controls delay element with different delay cycles. Set 1b to enable the delay element for the corresponding delay cycles. The total delay is the accumulation of all the delay cycles whose corresponding bit is set. • [9] - 16 delay cycles • [8] - 8 delay cycles • [7] - 4 delay cycles • [6] - 2 delay cycles • [5] - 1 delay cycle For example, if this field is configured as 10010b, this delay section 1 is configured as 16+2=18 delay cycles.
4-0 DELAY0	Delay Section 0 Configure the delay of section 0. Each bit controls delay element with different delay cycles. Set 1b to enable the delay elements for the corresponding delay cycles. The total delay is the accumulation of all the delay cycles whose corresponding bit is set. • [4] - 16 delay cycles • [3] - 8 delay cycles • [2] - 4 delay cycles • [1] - 2 delay cycles • [0] - 1 delay cycle For example, if this field is configured as 10010b, this delay section 0 is configured as 16+2=18 delay cycles.

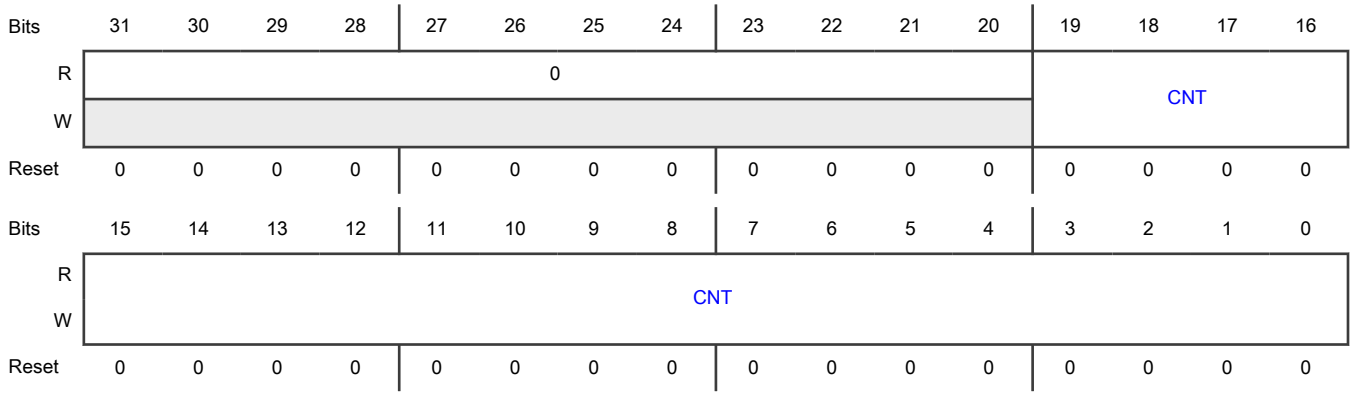
37.5.1.3 Alert Counter (ALERT_CNT0 - ALERT_CNT1)

Offset

Register	Offset
ALERT_CNT0	2Ch
ALERT_CNT1	ACh

Function
Counts the number of alerts.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-0 CNT	Counter The counter accumulates when detecting an alert. Writing any value will clear the counter.

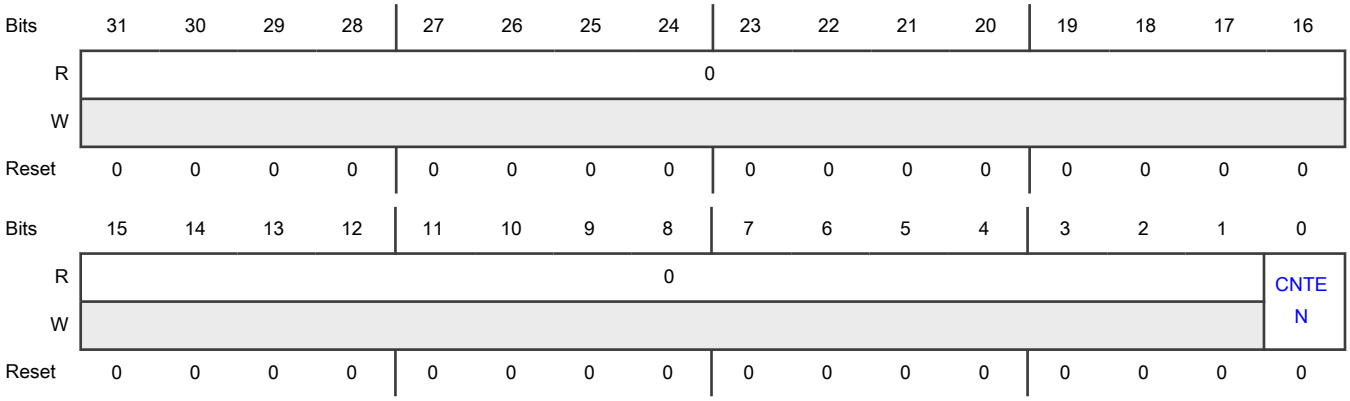
37.5.1.4 Alert Counter Control (ALERT_CNT_CTRL0 - ALERT_CNT_CTRL1)

Offset

Register	Offset
ALERT_CNT_CTRL0	30h
ALERT_CNT_CTRL1	B0h

Function
Control the counter to count the number of alerts.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CNTEN	Counter Enable Enables the counter to count the number of alerts. 0b - Disables 1b - Enables

Chapter 38
Analog-to-Digital Converter (ADC)

38.1 Chip-specific ADC information

Table 291. Reference links to related information

Topic	Related module	Reference
Full description	ADC	ADC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Interrupts and DMA Events	-	Interrupts and DMA events assignments INPUTMUX

38.1.1 Module instances

This device contains one instance of the ADC module, ADC0.

Table 292. ADC instance

Features	ADC0
Located in	VDD1_SENSE
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

38.1.2 ADC0 input connections

See table below for ADC0 input connections.

Table 293. ADC0 channel mapping

ADC0 channel (CMDLn[ADCH])	ADC0 A connection	ADC0 B connection	Input type
0_0000b (CH0)	ADC0_A0 (PIO10_0)	ADC0_B0 (PIO10_1)	External
0_0001b (CH1)	ADC0_A1 (PIO10_2)	ADC0_B1 (PIO10_3)	External
0_0010b (CH2)	ADC0_A2 (PIO10_4)	ADC0_B2 (PIO10_5)	External
0_0100b (CH4)	ADC0_A4 (PIO10_12)	ADC0_B4 (PIO10_13)	External
0_0101b (CH5)	ADC0_A5 (PIO10_14)	ADC0_B4 (PIO10_15)	External
0_0110b (CH6) [1]	PMC temperature sensor (+)	PMC temperature sensor (-)	Internal

1. See PMC temperature sensor for more details. Ignore the temperature sensor information in this chapter.

38.1.3 ADC voltage reference options

The ADC voltage reference options:

- CFG[REFSEL]=00b, VREFH reference pin. If an external reference voltage is not available, the on chip VREF_OUT can be used instead
- CFG[REFSEL]=01b or 10b, VDDA_ADC_1P8 supply pin
- CFG[REFSEL]=11b, reserved

38.1.4 ADC trigger inputs

ADC trigger sources get routed through the Input Multiplexing (INPUTMUX). See ADC0_TRIG0 - ADC0_TRIG1 of INPUTMUX1 in INPUTMUX chapter for available trigger sources.

38.1.5 ADC supply and ground connections

V_{SSA} and V_{REFL} is tied to common ground VSSA_0 in package.

38.2 Overview

The 16-bit ADC is a dual successive approximation converter designed for operation within an integrated microcontroller chip.

NOTE

For the chip-specific modes of operation, see the power management information of the chip.

38.2.1 Block diagram

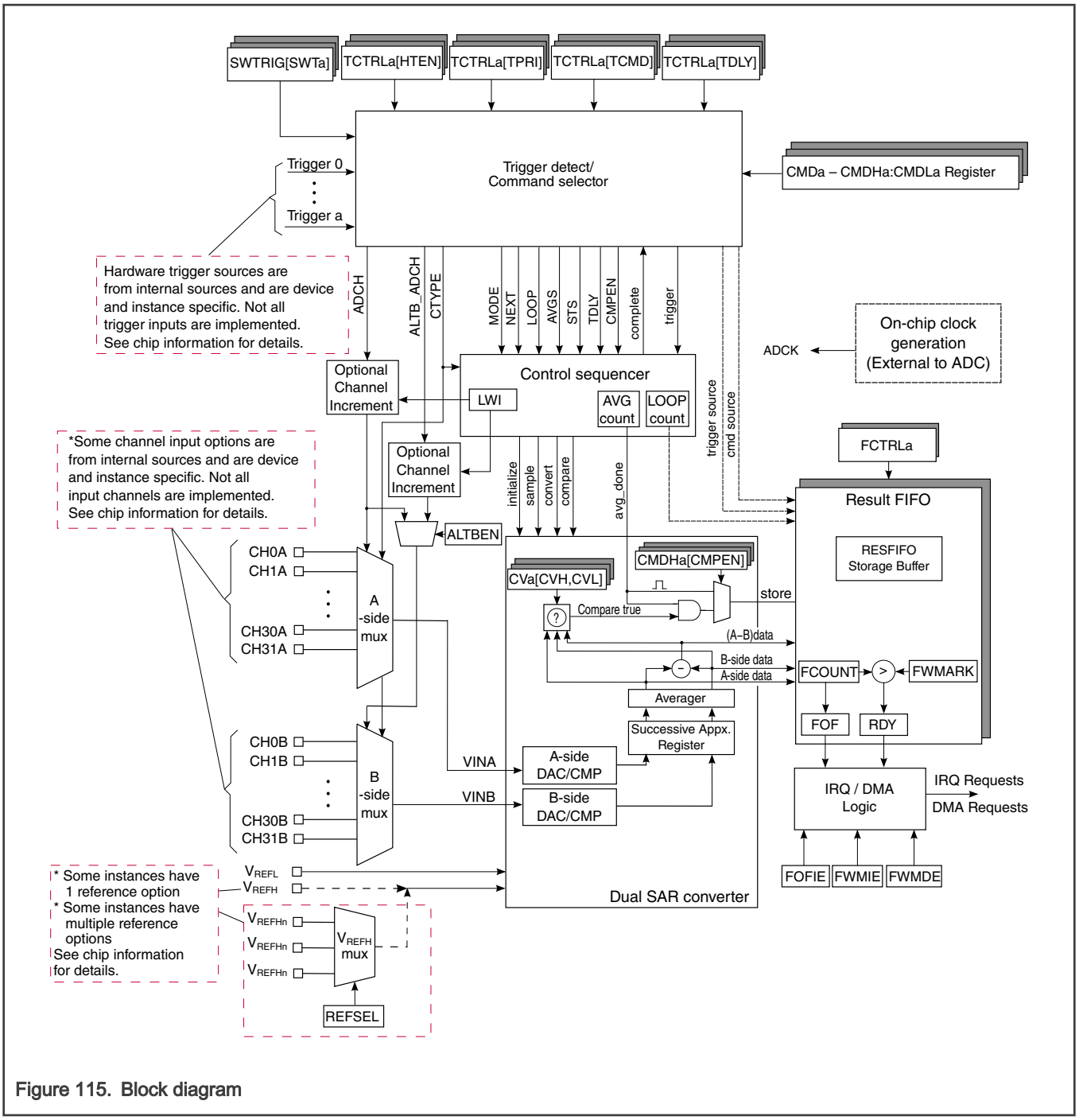


Figure 115. Block diagram

38.2.2 Features

- Linear successive approximation algorithm:
  - Differential operation with 16-bit or 13-bit resolution
  - Single-ended operation with 16-bit or 12-bit resolution
  - One ADC supports two simultaneous single-ended conversions
- Configurable analog input sample time

- Configurable speed options to accommodate operations in low-power modes
- Trigger detection with up to 2 trigger sources with priority-level configuration (with a software or hardware trigger option for each)
- Independent option selection and channel sequence scanning with 15 command buffers
- "Store on true" and "repeat until true" options for automatic compare of less-than, greater-than, within range, or out-of-range detection
- 2 independent result FIFOs, each containing 16 entries (each having a configurable watermark and overflow detection)
- Interrupt, DMA, or polled operations
- Linearity and gain adjustment calibration logic

38.3 Functional description

ADC performs analog-to-digital conversions on any of the software-selectable analog input channels through a successive approximation algorithm. ADC initializes to its lowest power state during reset. The ADC analog circuits can optionally be pre-enabled for faster conversions at the expense of higher idle currents. Selectable trigger events from software or hardware sources initiate the conversion. The trigger detect logic includes a configurable enable and priority scheme for the available trigger sources. ADC has multiple command buffers that provide configurable flexibility for channel scanning and independent channel selections for different trigger sources. Multiple command buffers also allow variable option selection such as differential vs. single-ended, sample time and averaging on a per-channel basis.

ADC optionally averages the result of multiple conversions on a channel before storing the calculated result. The hardware-average function is enabled by writing a nonzero value to CMDH*n*[AVGS] and operates in any of the conversion modes and configurations.

After the conversion and averaging loops are completed, the resulting data is placed in one of the 2 available FIFO data buffers along with other tag information associated with the result. A configurable watermark level supports interrupt or DMA requests when the number of stored data words exceeds the setting. You can also enable interrupts to indicate the occurrence of FIFO overflow.

ADC optionally compares the result of conversion with the contents of two value registers for less-than, greater-than, inside-range, or outside-range detection. The compare function operates in any of the conversion modes and configurations.

ADC also includes a hardware calibration logic to correct offset, gain, and linearity errors. You must execute calibration steps after every reset or power-up operation. Each SAR conversion utilizes calibration data calculated during the calibration routine.

The following flow diagram summarizes the sequencing of an ADC command.

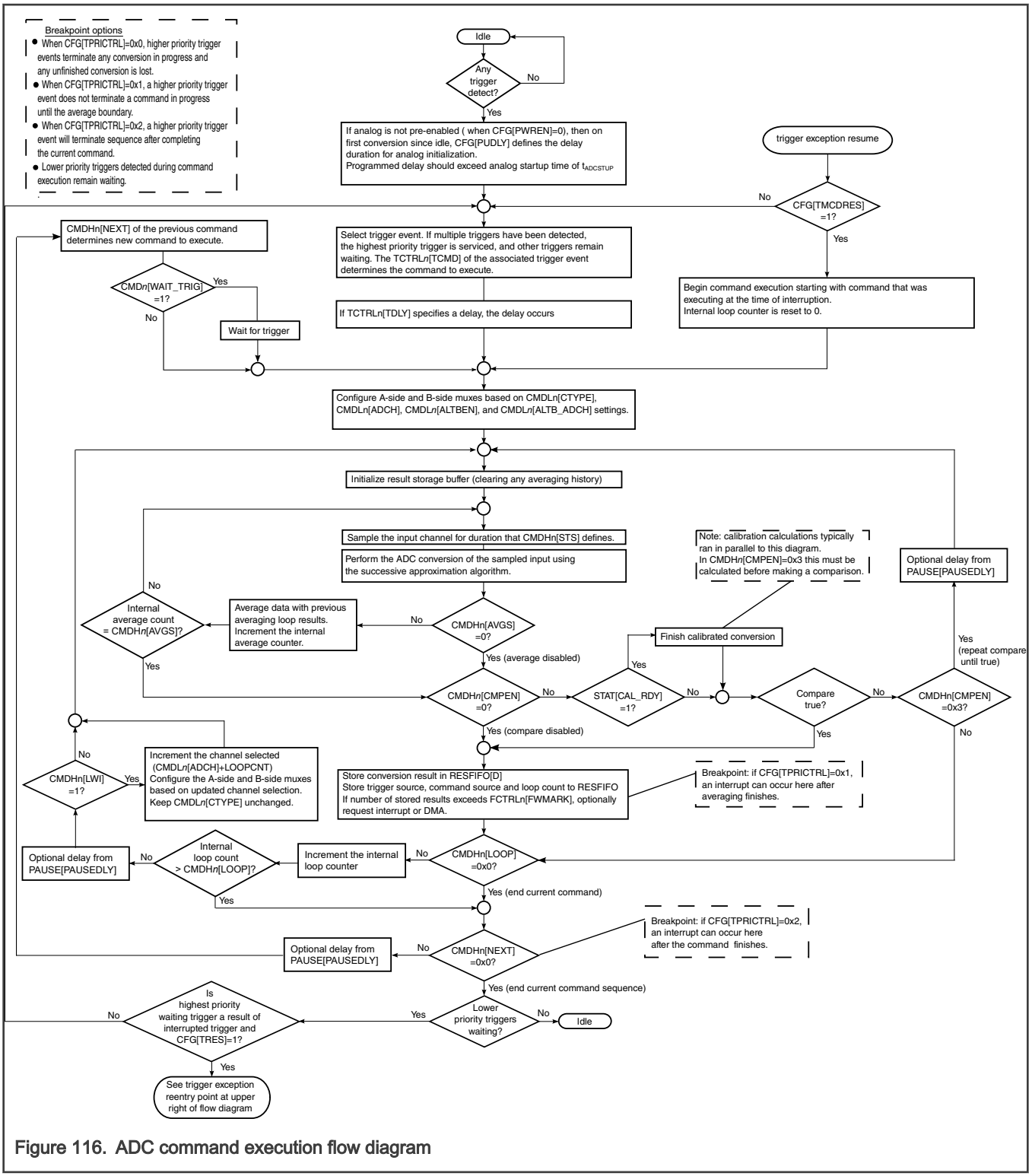


Figure 116. ADC command execution flow diagram

38.3.1 Power control mode

When ADC is in its idle state, the default setting for the ADC analog circuits is disabled. When a trigger is detected and ADC command processing is initiated, the analog circuits are enabled and require a period of initialization before the first conversion cycle. You must program CFG[PUDLY] in a way that a delay duration longer than $t_{ADCSTUP}$ is incurred. The accuracy of the initial conversion(s) after activation is degraded when CFG[PUDLY] is set to too small a value.

You can achieve faster conversion startup times by optionally writing 1 to CFG[PWREN], to pre-enable the analog circuits of ADC at the expense of power consumption even when ADC remains in the idle state. When CFG[PWREN] = 1, the power enable timer is activated and enforces the minimum startup (controlled by CFG[PUDLY]) before detected triggers are allowed to initiate ADC conversions.

38.3.2 Behavior in low-power modes

ADC supports system-level, low-power modes. In these modes, where ADC can remain functional, CTRL[DOZEN] controls the ADC behavior.

When CTRL[DOZEN] = 0, if you configure and enable ADC before Low-Power mode entry, ADC continues to operate while the system transitions to Low-Power mode. ADC does not disrupt any conversion in progress. External hardware trigger detection and active conversion are operational.

When CTRL[DOZEN] = 1, ADC waits for the current averaging iteration or FIFO storage to complete before acknowledging low-power entry requests. After the system enters Low-Power mode, ADC remains in an inactive state until the system exits the mode.

When entering a low-power mode where ADC cannot remain functional, it ignores CTRL[DOZEN]. ADC waits for the current average iteration or FIFO storage to complete before acknowledging a low-power entry request. After the system enters Low-Power mode, ADC remains inactive until the system exits this mode. The configuration of CFG[PWREN] is ignored and the ADC analog is forced to its lowest power state. After wake-up from Low-Power mode, if you preenable the analog before it enters Stop mode (CFG[PWREN] = 1) and the control registers retain the state in Low-Power mode, the analog is automatically reenabled and the power up delay timer begins. Any triggered conversion is stalled until the power-up delay associated with CFG[PUDLY] is complete.

38.3.3 Voltage reference

ADC uses the voltage reference high (V_{REFH}) that is supplied from either an on-chip voltage reference source or from an off-chip source through external pins. V_{REFL} is always from an external pin and must be at the same voltage potential as V_{SSA}. See the chip configuration information on the voltage reference options specific to this packaged device.

ADC supports a programmable selection of the voltage reference used for conversions (through CFG[REFSEL]). See the chip configuration information related to voltage reference options specific to this packaged device.

38.3.4 Trigger detect and command execution

See Figure 116 for a flow diagram of command execution sequencing.

ADC command execution is initiated from up to 2 trigger sources. You can generate each trigger by writing 1 to the corresponding SWTRIG[SWT*n*]. Alternatively, hardware triggers can be generated from asynchronous input sources at the periphery of the module. The number and sources of hardware triggers implemented is chip-specific (see the chip-specific ADC information for a description of available hardware trigger sources on this chip).

Writing 1 to the associated TCTRL*n*[HTEN] enables the hardware trigger source. Each trigger source is assigned a priority through the associated priority control field (TCTRL*n*[TPRI]), and each of the trigger sources is linked to a command buffer through the associated command select field (TCTRL*n*[TCMD]).

When you enable the hardware trigger input, it detects hardware trigger events on the rising-edge of the associated hardware trigger source.

Each trigger source has an associated priority field (TCTRL*n*[TPRI]) that allows arbitration between trigger sources. Arbitration controls the selection of the trigger sequence to execute next and how to handle a trigger exception. Trigger exceptions are defined as allowing a higher priority trigger sequence to interrupt operations of a lower-priority sequence. When a trigger exception occurs, programmable arbitration allows the configurable stop and resume points for low-priority sequences. The fields affecting arbitration are CFG[HPT_EXDI], CFG[TCMDRES], CFG[TRES], and CFG[TPRICTRL]:

- When CFG[HPT_EXDI] = 1b, trigger exceptions are disabled and any higher priority triggers are left pending until the current sequence completes.

NOTE

New triggers are accepted based on priority.

- When CFG[HPT_EXDI] = 0b (default), exceptions are enabled and the higher priority sequence begins executing at a breakpoint that you specify.

CFG[TPRICTRL] determines breakpoint locations, and CFG[TPRICTRL] affects latency for accepting a trigger exception:

- When CFG[TPRICTRL] = 0h, a higher priority trigger causes an immediate command abort and the new command specified by the trigger is immediately started.
- When CFG[TPRICTRL] = 1h, the current conversion is allowed to complete (including averaging) before the higher priority exception is initiated. In this mode, if the command is running through a series of averages, this series completes. However, it is not required to finish the entire command before interruption. For example, if the command consists of four loop iterations, it is not required to complete all four iterations before the interrupt occurs.
- When CFG[TPRICTRL] = 2h, a higher priority trigger begins on completion of the current command. If a command consists of five loop iterations, each containing eight averages, then all 5×8 conversions must complete before accepting the trigger exception.

CFG[TCMDRES] and CFG[TRES] determine what ADC does after accepting a trigger exception. You can program ADC to resume commands after returning from a trigger exception:

- If CFG[TRES] = 0h, commands do not resume automatically after stopped by an exception. However, an interrupt is set to indicate this case has occurred. TSTAT[TEXC_NUM] is used to resolve the trigger that was stopped by an exception.
- If CFG[TRES] = 1h, ADC automatically resumes commands after being stopped by an exception.

By utilizing CFG[TRES] in conjunction with CFG[TCMDRES], you can program ADC to resume commands at one of the two possible locations:

- If CFG[TCMDRES] = 0h, then the trigger that was stopped by an exception is resumed from the beginning of its associated command sequence.

NOTE

Triggers waiting to be resumed take the same priority as TCTRL*n*[TPRI].

- If CFG[TCMDRES] = 1h, then the trigger resumes from the command that it was executing before being interrupted by an exception.

If a lower-priority trigger occurs (that is, a trigger event occurs that is configured for a lower-priority than the trigger source associated with the currently executing command), the trigger detect is left pending until the current command sequence is completed. ADC cannot service lower-priority trigger events until a higher-priority triggered command (or command sequence) completes.

After a conversion is complete (including hardware averaging when CMDH*n*[AVGS] is a nonzero value), ADC places the result in a RESFIFO buffer. When an ADC command selects looping (CMDH*n*[LOOP] is a nonzero value), a command stores multiple conversion results to the FIFO during execution of that command.

At the end of command execution, CMDH*n*[NEXT] selects the next command to be executed. You can execute multiple commands sequentially by configuring each command's CMDH*n*[NEXT]. Writing 0 to CMDH*n*[NEXT] terminates conversions at the completion of the current command. ADC allows unending circular command execution when you write 1 to CMDH*n*[NEXT], in the last command in a sequence to the first command in the sequence.

By default, the command sequence executes automatically in the order you program CMDH*n*[NEXT]. However, you can stall and launch the command execution utilizing CMDH*n*[WAIT_TRIG] based on trigger inputs. For example, if you program TRIGGER2 to start the command sequence CMD1, CMD2, CMD3, then receiving TRIGGER2 one time unconditionally runs this sequence to completion. If you write 1 to CMDH2[WAIT_TRIG], then the sequence pauses after CMD1 until TRIGGER2 is received again. Therefore, sequences can be stalled until receiving a trigger assertion.

Disabling ADC by writing 0 to CTRL[ADCEN] terminates any active ADC command processing. Writing 0 to CTRL[ADCEN] causes the current command (or command sequence) to terminate, clears any pending triggers, and sends ADC to an idle state.

38.3.5 Pause option

If an application does not require maximum conversion rate, you can reduce the effective conversion rate by:

- Implementing periodic trigger events to initiate ADC conversions.
- Selecting a reduced frequency clock as the ADACK source.

Both options are chip-specific and depend on ADC triggering and clocking options external to ADC. The latency associated with ADC analog power-up delay results in a limit on the maximum conversion rate when using periodic triggering.

Another means of reducing conversion rates is by inserting a pause of a programmable duration between LOOP iterations, between commands in a sequence, and between conversions when a command is executing in the "compare until true" configuration. When PAUSE[PAUSEEN] = 1, PAUSE[PAUSEDLY] controls the pause duration during command execution sequencing. The pause delay is a count of (PAUSE[PAUSEDLY] × 4) ADCK cycles.

NOTE

You must not write to Pause (PAUSE) when CTRL[ADCEN] = 1. This can lead to a metastable operation.

See Figure 116 for the places where the pause is optionally inserted during command execution sequencing.

38.3.6 Resynchronization functionality

You can configure any software or hardware trigger source to act as a resynchronization trigger. You can use the trigger-based resynchronization functionality to interrupt a running trigger (resynchronization target) and clear the FIFO it's writing to. This can either be used to abort a running sequence, or restart a running sequence depending on the configuration of CFG[TRES].

If CFG[TRES] = 1, the target sequence is aborted and the FIFO is cleared after the resynchronization occurs. The sequence restarts after resynchronization occurs. If CFG[TRES] = 0, it aborts the target sequence and FIFO is cleared after the resynchronization occurs.

NOTE

The FIFO(s) cleared depend on the resynchronization targets TCTRL*m*[FIFO_SEL_A] and TCTRL*m*[FIFO_SEL_B]. To only clear one FIFO, TCTRL*m*[FIFO_SEL_A] == TCTRL*m*[FIFO_SEL_B]. Results not associated with the resynchronization target are lost if they are stored in either TCTRL*m*[FIFO_SEL_A] or TCTRL*m*[FIFO_SEL_B] at the time of resynchronization.

The resynchronization trigger must have a specific target. Resynchronization occurs only if the resynchronization target is running at the time of the trigger. For the following description, assume that *n* is the resynchronization trigger number and *m* is the resynchronization target number. According to these variables, trigger *n* must resynchronize trigger *m*. To enable a trigger source to act as a resynchronization trigger, the following conditions must be applied:

- TCTRL*n*[RSYNC] must be 1.
- The resynchronization trigger must have a higher priority than the resynchronization target (TCTRL*n*[TPRI] must be less than TCTRL*m*[TPRI]).
- The resynchronization target is specified by using TCTRL*n*[TCMD]. In this case, the resynchronization target, *m*, must be equal to TCTRL*n*[TCMD].
- The resynchronization target, *m*, must be executing commands when the resynchronization trigger, *n*, is asserted.
- Trigger *m* must have at least one conversion left to begin when trigger *n* is received.

If a trigger source *n* has TCTRL*n*[RSYNC] = 1, but some of the aforesaid conditions are not fulfilled, then the trigger source *n* is ignored.

Figure 117 illustrates a resynchronization trigger sequence executing. According to this figure:

- Trigger source 0 executes a sequence of commands when you assert trigger source 1.
- Trigger source 1 is configured to resynchronize trigger source 0.

- Trigger source 0 stops when trigger source 1 is asserted (after some synchronization delay). In addition, the FIFO that trigger source 0 writes to is cleared (FIFO0 in this example).

After the trigger 0 sequence stops, ADC runs the next trigger pending with the highest priority. This is marked as NXT for the next trigger. If you enable the resume functionality, and trigger source 0 has the highest priority pending, then NXT = 0h.

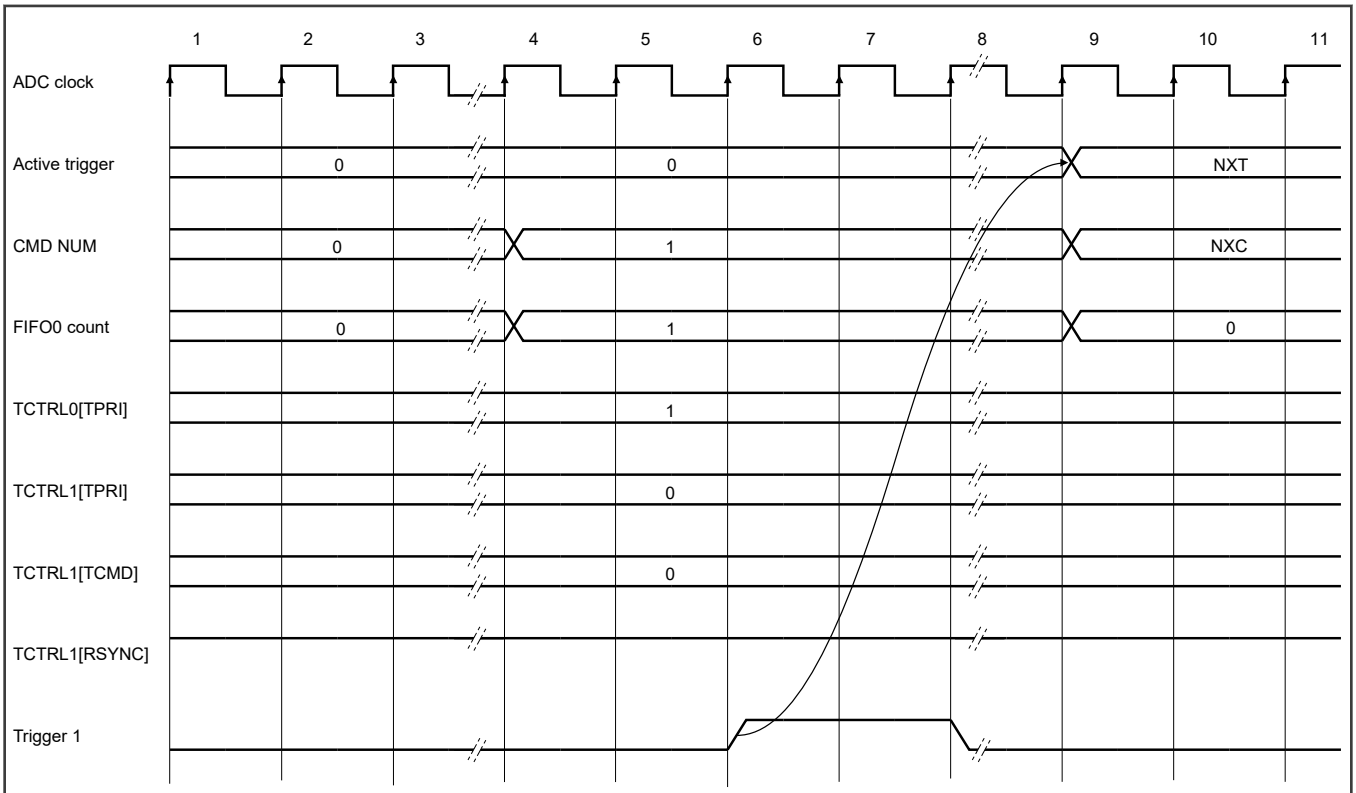


Figure 117. ADC resynchronization example

38.3.7 Temperature sensor

ADC has a dedicated input channel for an on-chip temperature sensor. See the chip-specific channel definition to determine which channel is connected to the on-chip temperature sensor.

To calculate the temperature, you must first execute a conversion of the temperature sensor channel with some configuration requirements. See the chip-specific channel definition to determine which channel corresponds to the temperature sensor. Perform the following procedure to convert the temperature sensor channel:

1. Configure a command buffer register to convert the temperature sensor channel. CMDL*n*[ADCH] = temperature sensor channel.
2. Program the command buffer with the following parameters:
   - 16-bit mode (CMDL*n*[MODE] = 1h)
   - Differential (CMDL*n*[CTYPE] = 2h)
   - Max averaging (CMDH*n*[AVGS] = Ah)
   - Max sample time (CMDH*n*[STS] = 7h)
   - LOOP set to 1 (CMDH*n*[LOOP] = 1h)
   - Loop with increment disabled (CMDH*n*[LWI] = 0h)
   - Compare function disabled (CMDH*n*[CMPEN] = 0h)

3. Configure Trigger Control (TCTRL0 - TCTRL1) with the following parameters:
   - TCTRL*n*[TCMD] = command buffer used in steps 2 and 3
   - TCTRL*n*[FIFO_SEL_A] directs conversion results to FIFO0 or FIFO1
4. Write 1 to the associated field in Software Trigger (SWTRIG) to trigger a conversion of the temperature sensor channel.

After conversion of the temperature sensor channel is complete, two results are stored in the FIFO selected with TCTRL*n*[FIFO_SEL_A]. Each result corresponds to a component of the overall temperature value. Read these values from the FIFO and use the following equation to calculate the ambient temperature.

$$Temp = A \times [\frac{\alpha \times (Vbe8 - Vbe1)}{Vbe8 + (\alpha \times (Vbe8 - Vbe1))}] - B$$

Figure 118. Temperature sensor function

Where:

- Vbe1 is the first value stored in the FIFO as a result of temperature sensor channel conversion.
- Vbe8 is the second value stored in the FIFO as a result of temperature sensor channel conversion.
- A is the slope factor.
- B is the offset factor.
- α is the bandgap coefficient

A, B, and α are specified constant values from the ADC electrical information in the chip data sheet.

38.3.8 Result FIFO operation

ADC includes 2 FIFOs with 16 entries each that store the result of ADC conversions. ADC also stores a valid indicator bit, the trigger source, the source command, and the loop count along with the data. FCTRL*n*[FCOUNT] indicates the number of valid datawords stored in each Data Result FIFO (RESFIFO0 - RESFIFO1).

A programmable watermark threshold supports configurable notification of data availability. When FCTRL*n*[FCOUNT] is greater than FCTRL*n*[FWMARK], it asserts the associated RDY flag. When IE[FWMIE*n*] = 1, a watermark interrupt request is issued. When DE[FWMDE*n*] = 1, a DMA request is issued. Reading RESFIFO provides the oldest unread dataword entry in the FIFO and decrements FCTRL*n*[FCOUNT]. When FCTRL*n*[FCOUNT] falls equal to or below FCTRL*n*[FWMARK], RDY is cleared.

Successive reads of RESFIFO*n* can empty FIFO. When RESFIFO*n*[VALID] = 1, the associated FIFO entry is valid. Reading RESFIFO*n* when the FIFO is empty (when RESFIFO*n*[VALID] = 0 and FCTRL*n*[FCOUNT] = 0h) provides an undefined dataword. All FIFOs are reset by writing 1 to CTRL[RSTFIFO*n*].

If ADC attempts to store a dataword in the FIFO when the FIFO is full, the FIFO overflow flag (FCTRL*n*[FOF]) becomes 1. When IE[FOFIE*n*] = 1, an overflow interrupt request is issued. You can clear STAT[FOF*n*] by writing 1 to it. During an overflow event, no new data is stored, and the data associated with the store that triggered the overflow is lost.

You can steer the conversion results to any FIFO in the design. Utilize TCTRL*n*[FIFO_SEL_A] and TCTRL*n*[FIFO_SEL_B] to determine the FIFO to write the final result. Therefore, depending on the executing trigger, you can steer the result to different locations. Depending on the type of conversion selected, the FIFO destination register fields are interpreted differently. During either differential or single-ended mode (CMDL*n*[CTYPE] ≠ 3h), only one result is produced. The destination during these modes is determined from TCTRL*n*[FIFO_SEL_A]. Dual-single-ended mode uses TCTRL*n*[FIFO_SEL_A] and TCTRL*n*[FIFO_SEL_B] to determine the channel A and channel B destinations, respectively.

38.3.9 Sampling modes

ADC supports three different sampling modes:

- Differential

- Single-Ended
- Dual Single-Ended

You can determine the sampling mode by executing a command using CMDL*n*[CTYPE]. During the execution of a command in Dual Single-Ended mode, two independent conversion results are calculated and stored in selectable FIFO destinations.

NOTE

The command processing is not individually controlled for each independent channel sampled. When operating in Dual Single-Ended mode, both the channels are sampled and processed simultaneously.

The selection of channel B is controlled by CMDL*n*[ALTBEN]. When CMDL*n*[ALTBEN] = 1, the channel for the B-side is selected independently (through the setting of CMDL*n*[ALTB_ADCH]). When CMDL*n*[ALTBEN] = 0, channel B selection is controlled by CMDL*n*[ADCH] and channels A and B are paired (for example CH0A/CH0B, CH1A/CH1B, ...). If comparisons are enabled in Dual Single-Ended mode, then only the A-side channels (CH0A, CH1A, CH2A, and so on) are used for the comparison. The A-side comparison is used to determine whether both the A-side and B-side results are written to the FIFOs:

- If the A-side comparison passes, both the A-side and B-side results are stored.
- If the A-side comparison fails, neither the A-side nor B-side results are written to the FIFOs.

Single-Ended mode is configured to allow either A-side or B-side channels to be sampled. In Single-Ended mode, the results from each conversion can be written to a selectable FIFO using TCTRL*n*[FIFO_SEL_A]. In Differential mode, the final SAR calculation is equivalent to V(CHA) - V(CHB). If the result is negative, then the value is stored in sign-extended two's complement format. TCTRL*n*[FIFO_SEL_A] also determines where differential conversion results are stored. In Dual Single-Ended mode; however, two independent results are produced. Individual control is provided by using TCTRL*n*[FIFO_SEL_A] and TCTRL*n*[FIFO_SEL_B] to select a FIFO destination for both results during this mode.

NOTE

Both single-ended results can be written to the same destination by programming TCTRL*n*[FIFO_SEL_A] equal to TCTRL*n*[FIFO_SEL_B]. In this case, the CH_A result is always stored before the CH_B result.

38.3.10 Compare function

After you sample and convert the input and perform any averaging iterations, CMDH*n*[CMPEN] guides the operation of the automatic compare function to optionally only store the compare result when the compare operation is true. There are multiple options on command sequencing related to the compare function as summarized in Table 294.

NOTE

Latency is added to the end of a compare function until true conversion to resolve the next command or loop in a sequence is achieved. This latency is necessary to calibrate the SAR data before resolving the result of comparison. Delay for this feature is only added when resolving the result of a conversion. This means that intermediate samples during averaging do not include extra latency. Only loop and command boundaries experience this delay. The latency is always less than or equal to five ADC clock cycles.

NOTE

Command buffers do not have an associated Compare Value register. The compare function is only available on Command buffers that have a corresponding Compare Value register.

Table 294. Compare modes

CMDH*n*[CMPEN]	Compare function	Description
0b00	Compare disabled	Does not perform the compare operation and stores the conversion result to the FIFO
0b01	Reserved	—

Table continues on the next page...

Table 294. Compare modes (continued)

CMDH*n*[CMPEN]	Compare function	Description
0b10	Store on true	Performs the compare operation and stores the conversion result in FIFO at the end of averaging when compare is true. If compare is false, it does not store the result to the FIFO. In either true or false condition, LOOP setting is considered and increments LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required.
0b11	Repeat compare until true	Performs the compare operation and stores the conversion results in FIFO at the end of averaging when compare is true. When the true condition is found, the setting of LOOP is considered and increments LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required. If the compare is false, it does not store the result in the FIFO. The conversion is repeated without consideration of LOOP setting and does not increment the LOOP counter.

Depending on the values that you program for CV*n*[CVH] and CV*n*[CVL], the compare operation checks whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. Table 295 describes the usage of the compare values.

Table 295. Compare operations

CV*n*[CVL] vs. CV*n*[CVH]	Operation	Description
Set CV*n*[CVL] < CV*n*[CVH]	Outside range (General form)	Compare true if the result is less than CV*n*[CVL] OR greater than CV*n*[CVH].
Set CV*n*[CVH] to maximum value Set CV*n*[CVL] to compare point	Less than	Compare true if the result is less than CV*n*[CVL].
Set CV*n*[CVL] to minimum value Set CV*n*[CVH] to compare point	Greater than	Compare true if the result is greater than CV*n*[CVH].
Set CV*n*[CVL] > CV*n*[CVH]	Inside range	Compare true if the result is less than CV*n*[CVL] AND greater than CV*n*[CVH].

NOTE

In low-power modes where ADC continues to operate, the compare function can monitor the voltage and only wake the chip when the compare condition is met.

38.3.11 Clocking

ADC operates from the ADCK clock input provided from an on-chip clock select block and is used by the SAR conversion control sequencing logic and the FIFO storage buffer. The ADCK frequency must be within the specified frequency range for ADCK. See chip data sheet for the supported frequency range.

ADC continues operating in low-power modes if CTRL[DOZEN] = 0, and the on-chip clock select block continues to supply an ADCK clock source.

NOTE

In Low-Power mode with CTRL[DOZEN] = 0b0, you can shut off the bus clock, and can configure asynchronous interrupts and DMA requests. In addition, ADC continues processing commands and writing data to the internal FIFO.

ADC has four sources for asynchronous interrupts during Low-Power mode:

- Watermark
- FIFO overflow
- TCOMP
- TEXC

To enable them, properly configure IE[FWMIE*n*], IE[FOFIE*n*], IE[TCOMP_IE], and IE[TEXC_IE] before entering Low-Power mode. When CTRL[DOZEN] = 1, in Low-Power mode, ADC waits for the current averaging iteration and FIFO storage to complete before acknowledging the Low-Power mode entry. When a Low-Power mode request is made with CTRL[DOZEN] = 1, it drops all pending triggers. After acknowledging Low-Power mode request, ADC is forced into its lowest power setting.

38.3.12 Resets

Table 296. ADC resets

Signal	Description
Chip reset	Resets the ADC logic and registers to their default states.
Software reset	Resets, by using CTRL[RST], logic and registers (except Control (CTRL), Gain Calculation Result (GCR0 - GCR1), Calibration General A-Side (CAL_GAR0 - CAL_GAR32), and Calibration General B-Side (CAL_GBR0 - CAL_GBR32)) to their default states.
FIFO reset	Resets CTRL[RSTFIFO0] and CTRL[RSTFIFO1] by implementing write-only control. After a FIFO is reset, that FIFO becomes empty.

38.3.13 Interrupts and DMA requests

ADC includes several sources for interrupts and DMA requests. The following table summarizes these sources.

A programmable watermark threshold supports configurable notification of data availability and can generate either an interrupt exception or a DMA request. When FCTRL*n*[FCOUNT] is greater than FCTRL*n*[FWMARK], it asserts the associated RDY*n* flag. IE[FWMIE*n*] and DE[FWMDE*n*] control the masking for this exception. When RDY*n* is asserted and IE[FWMIE*n*] = 1, a watermark interrupt request is issued. When RDY*n* is asserted and DE[FWMDE*n*] = 1, a DMA request is issued.

The other exception sources can only generate interrupts and do not have a DMA request option. Each of these sources has a mask control field in Interrupt Enable (IE) and no corresponding field in DMA Enable (DE).

Table 297. ADC interrupts and DMA requests

Status (STAT)		Description	Can generate		
Flag	Name		Interrupt?	DMA request?	Low-power wake-up?
RDY*n*	Result FIFO*n* ready flag	Indicates that conversion result data is written to the result FIFO and has a watermark configurable trigger level to	Y	Y	Y

Table continues on the next page...

Table 297. ADC interrupts and DMA requests (continued)

Status (STAT)		Description	Can generate		
Flag	Name		Interrupt?	DMA request?	Low-power wake-up?
		generate an exception request as controlled by FCTRL*n*[FWMARK]			
FOF*n*	Result FIFO*n* overflow flag	Attempts to store data to the FIFO when the FIFO is full and indicates an error condition	Y	N	Y
TCOMP_INT	Trigger completion flag	Indicates that a trigger sequence is complete and all its associated commands have been run	Y	N	Y
TEXC_INT	High priority trigger flag	Indicates that a high-priority trigger exception has occurred	Y	N	Y

38.4 External signals

ADC supports analog channel inputs with differential and single-ended conversion options for all channels. ADC also requires supply and ground connections.

Table 298. ADC signal descriptions

Signal	Description	I/O
V_{DDA}	Analog power supply	I
V_{SSA}	Analog ground	I
CH*n*A–CH0A[1]	A-side analog channel inputs	I
CH*n*B–CH0B[1]	B-side analog channel inputs	I

1. *n* is the maximum channel number supported in the chip (see the chip-specific information to know the number of channels supported on your chip).

38.4.1 Analog channel inputs (CH*n*A and CH*n*B)

CMDL*n*[ADCH] and CMDL*n*[CTYPE] control the selection of paired or individual input channels. Each ADC command independently makes a channel and conversion type selection. Each CMDL*n*[ADCH] channel selection has an associated A side and an associated B side input. You can optionally convert each CMDL*n*[ADCH] pair in a differential mode, but only limited pairs are intended to be converted as differential channels (adjacent pins designed with matched impedance). For the pin pairings available for differential conversions on your chip, see the chip configuration details.

NOTE

Some input channels are from on-chip sources such as temperature sensors and reference voltage sources and can only connect to individual instances of ADC. Some of the input channel options in the bitfield-setting descriptions may not be available for your chip. See the chip-specific information for the channels supported on this chip.

38.5 Initialization

38.5.1 Calibration

ADC consists of multiple calibration functions that you must execute as part of the ADC setup to achieve the specified accuracy. You must run calibration after every reset and before initiating a conversion. Prior to offset calibration or calibration, you must configure ADC's clock source and frequency according to the application's clock source availability and needs. You must enable

ADC (write 1 to CTRL[ADCEN]) before a calibration function runs. If calibration is requested when ADC is actively converting, that sequence completes before initiating the calibration function.

You can achieve improved accuracy during the calibration routines by averaging multiple conversions. You must write 1 to CTRL[CAL_AVGS] for a minimum of 256 averaging during calibration steps. If the application uses ADC in a wide variety of configurations, the configuration for which the highest accuracy is required must be selected, or multiple calibrations can be done for different configurations.

38.5.1.1 16-bit offset calibration function

Use OFSTRIM16 to trim ADC comparator offset voltage for 16-bit ADC conversions. ADC supports an offset calibration function in which OFSTRIM16 is automatically updated. Perform the following procedure for executing the function:

1. Configure the desired averaging though CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 8h or greater).
2. Initiate 16-bit offset calibration by writing 1 to CTRL[CALOFS] when CTRL[CALOFSMODE] is 1.
3. Poll STAT[CAL_RDY]. When STAT[CAL_RDY] is set, the offset calibration function is completed and OFSTRIM16 is updated.

38.5.1.2 12-bit offset calibration function

Use OFSTRIM12 to trim ADC comparator offset voltage for 12-bit ADC conversions. ADC supports an offset calibration function that automatically updates OFSTRIM12. Perform the following procedure for executing the function:

1. Configure the desired averaging through CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 8h or greater).
2. Initiate 12-bit offset calibration by writing 1 to CTRL[CALOFS] when CTRL[CALOFSMODE] is 0.
3. Poll STAT[CAL_RDY]. When STAT[CAL_RDY] is set, the offset calibration function is completed and OFSTRIM12 is updated.

38.5.1.3 Calibration function

ADC includes the following hardware calibration logic in which the A-side and B-side converters are calibrated for gain error and linearity error correction of the raw conversion result:

- GCR0[GCALR] and Calibration General A-Side (CAL_GAR0 - CAL_GAR32) control calibration for the A-side converter.
- GCR1[GCALR] and Calibration General B-Side (CAL_GBR0 - CAL_GBR32) control calibration for the B-side converter.

ADC supports a calibration function that automatically updates CAL_GAR*n* and CAL_GBR*n*. The calibration function also updates GCC0[GAIN_CAL] (associated with A-side calibration) and GCC1[GAIN_CAL] (associated with B-side calibration). For A-side calibration, a software calculation is required to derive GCR0[GCALR] from GCC0[GAIN_CAL]. For B-side calibration, a corresponding calculation is needed to derive GCR1[GCALR] from GCC1[GAIN_CAL]. Perform the following procedure for completing the calibration setup:

1. Execute the 16-bit offset calibration function described in 16-bit offset calibration function. OFSTRIM16 is used during calibration to trim for comparator offset voltage.
2. Configure the desired averaging through CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 8h or greater).
3. Initiate the calibration routine by writing 1 to CTRL[CAL_REQ], which remains 1 until the CAL routine is accepted by ADC. After acceptance, CTRL[CAL_REQ] becomes 0 automatically.
4. Update the A-side calibration data first. Poll GCC0[RDY]. When it is set, the hardware controlled A-side calibration operation is complete, and CAL_GAR*n* and GCC0[GAIN_CAL] are updated. The updated value in GCC0[GAIN_CAL] is further required for software processing as described in the following steps.
5. Read GCC0[GAIN_CAL] and store for use in the gain_adjustment calculation.

6. Calculate the A-side gain_adjustment as (131072) ÷ (131072-GCC0[GAIN_CAL]). GCC0[GAIN_CAL] is two's-complement value. This results in a floating point value between 0 and 2.
7. Convert the floating point value to its integer component (0 or 1) and its fractional component rounded to 16 bits. The integer value is stored in GCR0[GCALR[16]] and the fractional component is stored in GCR0[GCALR[15:0]]. Write this value to GCR0[GCALR].
8. Execute the same calculation for the B-side converter. Poll GCC1[RDY]. When it is set, the hardware controlled B-side calibration operation is complete, and CAL_GBR*n* and GCC1[GAIN_CAL] are updated. The updated value in GCC1[GAIN_CAL] requires further software processing.
9. Read GCC1[GAIN_CAL] and store for use in the gain_adjustment calculation.
10. Calculate the B-side gain_adjustment: (131072) ÷ (131072-GCC1[GAIN_CAL]). Formatting for the value is the same as that for A-side.
11. Write this value to GCR1[GCALR].
12. Set GCR0[RDY] and GCR1[RDY] flags to indicate they are valid, when GCR0[GCALR] and GCR1[GCALR] contain the results from the gain_adjustment calculations. It is acceptable for GCR*n*[GCALR] and corresponding GCR*n*[RDY] to be updated on the same write cycle.

The calibration sequence completes after the aforementioned steps are fulfilled and STAT[CAL_RDY] is set. STAT[CAL_RDY] remains set until you reset the system or request a new calibration sequence.

When STAT[CAL_RDY] is set, you must configure ADC to run in Calibration mode. Each conversion uses a combination of linearity and gain calibration results to correct SAR data. Calibration conversion latency is required to process each sample. However, because of the pipelined nature of data and control sequences, each conversion is initiated without experiencing this calibration delay.

38.5.2 Calibration general A-side and B-side register valid ranges

Calibration General A-Side (CAL_GAR0 - CAL_GAR32) and Calibration General B-Side (CAL_GBR0 - CAL_GBR32) have 33 registers in each array. The data in each register is a signed 16-bit value but has a variable range of possible values. The following table defines the valid range that each register supports.

Table 299. Calibration general registers valid values

Element CAL_G*n*R[N]	Valid range
N = 00h, 20h	–1024 to +1023
N = 01h	–2048 to +2047
N = 02h–03h	–4096 to +4095
N = 04h–07h	–8192 to +8191
N = 08h–0Fh	–16384 to +16383
N = 10h–1Fh	–32768 to +32767

These registers are typically updated automatically during the self-calibration sequence. To reduce the latency associated with the ADC setup, CAL_G*n*R values from a calibration sequence are stored in nonvolatile memory after an initial calibration, and you can write to CAL_G*n*R prior to the first ADC conversion. If you set these registers to values not generated by the calibration function, the linearity error specifications may not be met.

NOTE

You can only write these values in a single access. Byte accesses are not supported.

38.6 ADC register descriptions

38.6.1 ADC memory map

ADC0 base address: 4020_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0200_2C0Bh
4h	Parameter (PARAM)	32	R	0F04_1002h
10h	Control (CTRL)	32	RW	0000_0020h
14h	Status (STAT)	32	RW	0000_0000h
18h	Interrupt Enable (IE)	32	RW	0000_0000h
1Ch	DMA Enable (DE)	32	RW	0000_0000h
20h	Configuration (CFG)	32	RW	0080_0000h
24h	Pause (PAUSE)	32	RW	0000_0000h
34h	Software Trigger (SWTRIG)	32	RW	0000_0000h
38h	Trigger Status (TSTAT)	32	RW	0000_0000h
40h	Offset Trim 16 Bits (OFSTRIM16)	32	RW	0000_0000h
44h	Offset Trim 12 Bits (OFSTRIM12)	32	RW	0000_0000h
A0h - A4h	Trigger Control (TCTRL0 - TCTRL1)	32	RW	0000_0000h
E0h - E4h	FIFO Control (FCTRL0 - FCTRL1)	32	RW	0000_0000h
F0h - F4h	Gain Calibration Control (GCC0 - GCC1)	32	R	0000_0000h
F8h - FCh	Gain Calculation Result (GCR0 - GCR1)	32	RW	0001_0000h
100h	Command Low Buffer (CMDL1)	32	RW	0080_0000h
104h	Command High Buffer (CMDH1)	32	RW	0000_0000h
108h	Command Low Buffer (CMDL2)	32	RW	0080_0000h
10Ch	Command High Buffer (CMDH2)	32	RW	0000_0000h
110h	Command Low Buffer (CMDL3)	32	RW	0080_0000h
114h	Command High Buffer (CMDH3)	32	RW	0000_0000h
118h	Command Low Buffer (CMDL4)	32	RW	0080_0000h
11Ch	Command High Buffer (CMDH4)	32	RW	0000_0000h
120h	Command Low Buffer (CMDL5)	32	RW	0080_0000h
124h	Command High Buffer (CMDH5)	32	RW	0000_0000h
128h	Command Low Buffer (CMDL6)	32	RW	0080_0000h
12Ch	Command High Buffer (CMDH6)	32	RW	0000_0000h
130h	Command Low Buffer (CMDL7)	32	RW	0080_0000h
134h	Command High Buffer (CMDH7)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
138h	Command Low Buffer (CMDL8)	32	RW	0080_0000h
13Ch	Command High Buffer (CMDH8)	32	RW	0000_0000h
140h	Command Low Buffer (CMDL9)	32	RW	0080_0000h
144h	Command High Buffer (CMDH9)	32	RW	0000_0000h
148h	Command Low Buffer (CMDL10)	32	RW	0080_0000h
14Ch	Command High Buffer (CMDH10)	32	RW	0000_0000h
150h	Command Low Buffer (CMDL11)	32	RW	0080_0000h
154h	Command High Buffer (CMDH11)	32	RW	0000_0000h
158h	Command Low Buffer (CMDL12)	32	RW	0080_0000h
15Ch	Command High Buffer (CMDH12)	32	RW	0000_0000h
160h	Command Low Buffer (CMDL13)	32	RW	0080_0000h
164h	Command High Buffer (CMDH13)	32	RW	0000_0000h
168h	Command Low Buffer (CMDL14)	32	RW	0080_0000h
16Ch	Command High Buffer (CMDH14)	32	RW	0000_0000h
170h	Command Low Buffer (CMDL15)	32	RW	0080_0000h
174h	Command High Buffer (CMDH15)	32	RW	0000_0000h
200h - 20Ch	Compare Value (CV1 - CV4)	32	RW	0000_0000h
300h - 304h	Data Result FIFO (RESFIFO0 - RESFIFO1)	32	R	0000_0000h
400h - 480h	Calibration General A-Side (CAL_GAR0 - CAL_GAR32)	32	RW	0000_0000h
500h - 580h	Calibration General B-Side (CAL_GBR0 - CAL_GBR32)	32	RW	0000_0000h
FF8h	Configuration 2 (CFG2)	32	RW	0000_0000h

38.6.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the chip and inclusion and exclusion of several optional features.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	NUM_FIFO			NUM_SEC	CALOFSI	IADCKI	VR1RNGI	0	CSW			MVI	0	DIFFEN	RES
W																
Reset	0	0	1	0	1	1	0	0	0	0	0	0	1	0	1	1

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module specification.
15 —	Reserved This bit always has the value 0.
14-12 NUM_FIFO	Number of FIFOs Indicates the number of result FIFOs implemented in the design. 000b - N/A 001b - One 010b - Two 011b - Three 100b - Four
11 NUM_SEC	Number of Single-Ended Outputs Supported Indicates the number of single-ended channels that can be processed simultaneously. 0b - One conversion 1b - Two conversions
10 CALOFSI	Calibration Function Implemented Indicates whether ADC contains hardware calibration functions to implement calibration. When supported, CTRL[CALOFS] and CTRL[CAL_REQ] can be used to request the calibration routines to run.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not implemented 1b - Implemented
9 IADCKI	Internal ADC Clock Implemented Indicates whether this implementation of ADC includes or implements an internal clock source. When supported, CFG[ADCKEN] is available, implemented, and documented for enabling the clock source. When available, this clock source is used for clock selection logic external to the module for use within the system. 0b - Not implemented 1b - Implemented
8 VR1RNGI	Voltage Reference 1 Range Control Bit Implemented Indicates whether a control bit is implemented for selecting the input voltage range on voltage reference option 1. When range control is required, CFG[VREF1RNG] is available, implemented, and documented for controlling the voltage reference option 1. 0b - Range control not required 1b - Range control required
7 —	Reserved This bit always has the value 0.
6-4 CSW	Channel Scale Width Indicates whether channel scaling is supported by this implementation. When supported, each command buffer implements a control field, CMDL*n*[CSCALE], to configure input scaling. 000b - Not supported (the CSCALE field not implemented) 001b - 1-bit CSCALE supported 110b - 6-bit CSCALE supported
3 MVI	Multi Vref Implemented Indicates whether multiple voltage reference high (VREFH) inputs are supported for this implementation. When supported, CFG[REFSEL] is available and documented for selecting the voltage reference high options. 0b - Single VREFH input supported 1b - Multiple VREFH inputs supported
2 —	Reserved This bit always has the value 0.
1 DIFFEN	Differential Supported

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates whether the differential operation is supported by this implementation. When supported, each command buffer implements control fields (CMDL*n*[CTYPE]) for configuring differential operations and expanding the number of supported channels from 32 to 64. 0b - Not supported 1b - Supported
0 RES	Resolution Indicates the maximum accuracy that is supported by this implementation. When this field is 1, each command buffer has a control field (CMDL*n*[MODE]) for selecting the resolution of conversions for the associated command. 0b - Up to 13-bit differential or 12-bit single-ended resolution 1b - Up to 16-bit differential or 16-bit single-ended resolution supported

38.6.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the size of several variable integration options for this instance on the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMD_NUM								CV_NUM							
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOSIZE								TRIG_NUM							
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-24	Command Buffer Number

Table continues on the next page...

Table continued from the previous page...

Field	Function
CMD_NUM	Indicates the number of implemented command buffers.
23-16 CV_NUM	Compare Value Number Indicates the number of implemented compare value registers.
15-8 FIFOSIZE	Result FIFO Depth Indicates the maximum number of conversion data words, indicating the result FIFO depth, that can be stored in the result FIFO before an overflow occurs. 0000_0001b - 2 0000_0100b - 4 0000_1000b - 8 0001_0000b - 16 0010_0000b - 32 0100_0000b - 64
7-0 TRIG_NUM	Trigger Number Indicates the number of triggers implemented.

38.6.4 Control (CTRL)

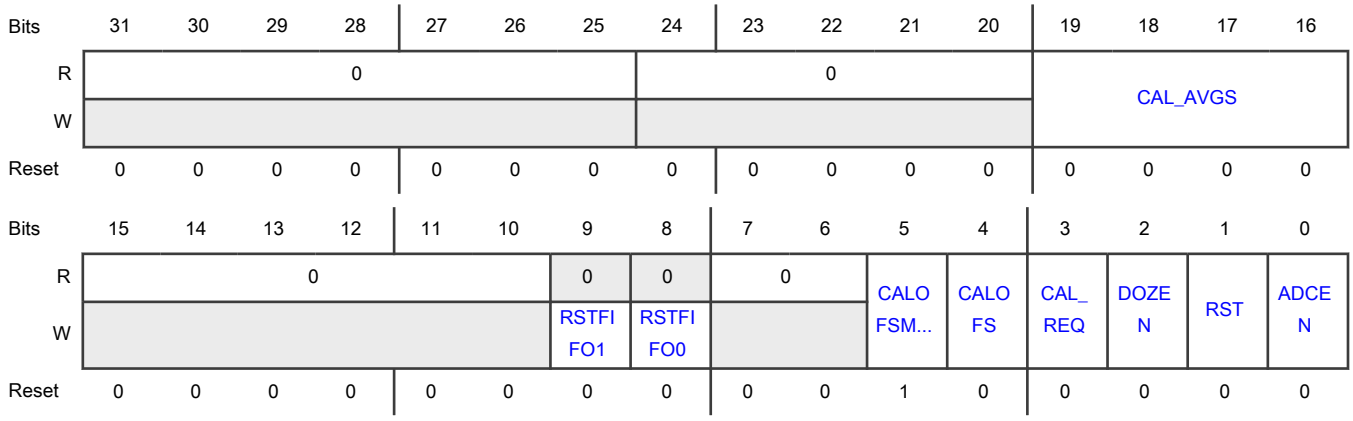
Offset

Register	Offset
CTRL	10h

Function

Includes primary control fields.

Diagram



Fields

Field	Function
31-25 —	Reserved The bit always has the value 0.
24-20 —	Reserved The bit always has the value 0.
19-16 CAL_AVGS	Auto-Calibration Averages Selects the number of ADC conversions averaged to calculate each calibration value. Selecting a higher number of averages leads to more accurate conversions after completing calibration. The recommended minimum setting is for 256 averaging (CAL_AVGS = 8h or greater). CAL_AVGS applies to both CALOFS and CAL_REQ calibration types. You must fix the value when requesting and running calibration with CAL_REQ or CALOFS. 0000b - Single conversion 0001b - 2 0010b - 4 0011b - 8 0100b - 16 0101b - 32 0110b - 64 0111b - 128 1000b - 256 1001b - 512 1010b - 1024
15-10 —	Reserved
9 RSTFIFO1	Reset FIFO 1 0b - No effect 1b - FIFO 1 is reset
8 RSTFIFO0	Reset FIFO 0 0b - No effect 1b - FIFO 0 is reset
7-6 —	Reserved The bit always has the value 0.
5 CALOFSMODE	Configure Mode for Offset Calibration Function

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Configures offset calibration for 16-bit and 12-bit modes. When CALOFSMODE = 1, executing the offset calibration function configures the analog block for 16-bit mode and the results are updated in Offset Trim 16 Bits (OFSTRIM16). When CALOFSMODE = 0, executing the offset calibration function configures the analog block for 12-bit mode and the results are updated in Offset Trim 12 Bits (OFSTRIM12). 0b - 12-bit mode 1b - 16-bit mode
4 CALOFS	Offset Calibration Request Initiates a hardware-offset calibration calculation. The setting of CALOFSMODE controls whether a 16-bit offset calibration or a 12-bit offset calibration is executed. Any conversions in progress are completed before launching the offset calibration function. After being accepted, ADC calculates the offset value, and depending on the setting of CALOFSMODE, updates either Offset Trim 16 Bits (OFSTRIM16) or Offset Trim 12 Bits (OFSTRIM12) automatically. Hardware clears this field after completion of the offset calibration calculation. STAT[CAL_RDY] indicates the completion of the offset calibration routine. 0b - Disable calibration function 1b - Request for offset calibration function
3 CAL_REQ	Auto-Calibration Request Requests the hardware calibration routine to run. This field automatically becomes 0 when the system accepts the hardware calibration request. STAT[CAL_RDY] indicates the calibration routine completion. 0b - Request for hardware calibration not made 1b - Request for hardware calibration made
2 DOZEN	Doze Enable Controls system transition to low-power modes when ADC is converting. When this field is 0, immediate entries to low-power modes are allowed and ADC conversion functions remain enabled. Any conversion in progress is not disrupted. Note that the selected clock sources provided from the on-chip clock source must continue operating. If this field is 1 and ADC is actively converting, and the system is entering Low-Power mode, ADC waits for the current averaging iteration or FIFO storage to complete before acknowledging low-power entry request. After the system enters the low-power state, ADC remains in an inactive state until the system exits the low-power state. When the system enters Low-Power mode in which the ADC operation is not supported, this field is ignored and ADC waits for the current transfer to complete any pending operation. 0b - ADC conversions enabled in Low-Power mode 1b - ADC conversions disabled in Low-Power mode
1 RST	Software Reset Resets all internal logic and registers, except for Control (CTRL), Gain Calculation Result (GCR0 - GCR1), Calibration General A-Side (CAL_GAR0 - CAL_GAR32), and Calibration General B-Side (CAL_GBR0 - CAL_GBR32). This field remains 1 until you write 0 to it. 0b - Do not reset ADC logic

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Reset ADC logic
0 ADCEN	ADC Enable Enables ADC. 0b - Disable 1b - Enable

38.6.5 Status (STAT)

Offset

Register	Offset
STAT	14h

Function

Indicates the current status of ADC.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				CMDACT				0							TRGACT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ADC_ACT...	CAL_RDY	TCOMP_I...	TEXC_INT	0				FOF1	RDY1	FOF0	RDY0
W							W1C	W1C					W1C		W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved This bit always has the value 0.
27-24 CMDACT	Command Active

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Works with ADC_ACTIVE to determine which conversion is running. This field indicates the command that is being actively processed. Commands are only shown here when they are being actively processed by the SAR routine. 0000b - No command 0001b - Command 1 0010b - Command 2 0011b-1111b - Associated command number
23-17 —	Reserved
16 TRGACT	Trigger Active Indicates the trigger that is being actively processed, running through a trigger delay, or is being converted by the SAR routine. This field indicates the command associated with TRGACT that is running through a conversion when CMDACT is a nonzero value. 0b - Trigger 0 1b - Trigger 1
15-12 —	Reserved
11 ADC_ACTIVE	ADC Active Indicates whether ADC is currently processing a conversion or has pending triggers to service. When this field is 0, ADC is in the Idle state with no pending triggers to service and no active commands to process. If this field is 1, it indicates that this field is processing a conversion (running through the power-up delay or servicing a trigger). 0b - Idle 1b - Processing a conversion
10 CAL_RDY	Calibration Ready Indicates whether the calibration request received from CTRL[CAL_REQ] or CTRL[CALOFS] is complete and the results are ready. This flag is automatically cleared when a new hardware calibration or offset calibration request is made. 0b - Calibration is incomplete or not ready 1b - Calibration complete
9 TCOMP_INT	Interrupt Flag for Trigger Completion Indicates whether a trigger sequence is complete (all associated commands are run). If complete, TCOMP_INT is asserted. **NOTE** IE[TCOMP_IE] must be 1 for the specific trigger source to flag an interrupt after completion.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - Either IE[TCOMP_IE] is 0 or no trigger sequences are complete 1b - Trigger sequence completed; all data stored in the associated FIFO When writing 0b - No effect 1b - Clear the flag
8 TEXC_INT	Interrupt Flag for High Priority Trigger Exception Indicates whether a high-priority trigger exception occurred. In case it occurred and CFG[TRES] = 0, this flag is asserted, which happens only if trigger exception interrupts are enabled (IE[TEXC_IE] = 1h). TEXC_INT and TSTAT[TEXC_NUM] can be used in conjunction to resolve the trigger source that was interrupted. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No trigger exceptions occurred 1b - Trigger exceptions occurred; acknowledgment is pending When writing 0b - No effect 1b - Clear the flag
7-4 —	Reserved
3 FOF1	Result FIFO1 Overflow Flag Indicates whether more data is written to this field than it can hold since the last time the flag was cleared. The new data is not stored and the FIFO contains the original contents. This flag asserts regardless of the value of IE[FOFIE1]. However, an interrupt request is issued only when IE[FOFIE1] is 1. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No result FIFO1 overflow occurred 1b - At least one result FIFO1 overflow occurred When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
2 RDY1	Result FIFO1 Ready Flag Indicates whether the number of valid datawords in the result FIFO1 is greater than the watermark level defined in FCTRL*n*[FWMARK]. When it is above the watermark level, the flag asserts regardless of the value of IE[FWMIE1]. However, an interrupt request or DMA request occurs only when the associated fields, IE[FWMIE1] and DE[FWMDE1], are 1. This flag is cleared when FCTRL*n*[FCOUNT] (which decrements on each read of Data Result FIFO (RESFIFO0 - RESFIFO1)) is less than or equal to the watermark level set in FCTRL*n*[FWMARK]. 0b - Below watermark level 1b - Above watermark level
1 FOF0	Result FIFO 0 Overflow Flag Indicates whether more data is written to the result FIFO 0 than it can hold since the last time the flag was cleared. Newer data is not stored and the FIFO keeps holding the original content. This flag asserts regardless of the value of IE[FOFIE0]. However, an interrupt request is issued only if IE[FOFIE0] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No result FIFO 0 overflow occurred 1b - At least one result FIFO 0 overflow occurred When writing 0b - No effect 1b - Clear the flag
0 RDY0	Result FIFO 0 Ready Flag Indicates whether the number of valid datawords in the result FIFO 0 is greater than the watermark level set in FCTRL*n*[FWMARK]. This flag asserts regardless of the value of IE[FWMIE0]. However, an interrupt request or DMA request occurs only when the associated control fields, IE[FWMIE0] and DE[FWMDE0], are 1. This flag is cleared when FCTRL*n*[FCOUNT] (which decrements on each read of Data Result FIFO (RESFIFO0 - RESFIFO1)) is less than or equal to the watermark level set in FCTRL*n*[FWMARK]. 0b - Below watermark level 1b - Above watermark level

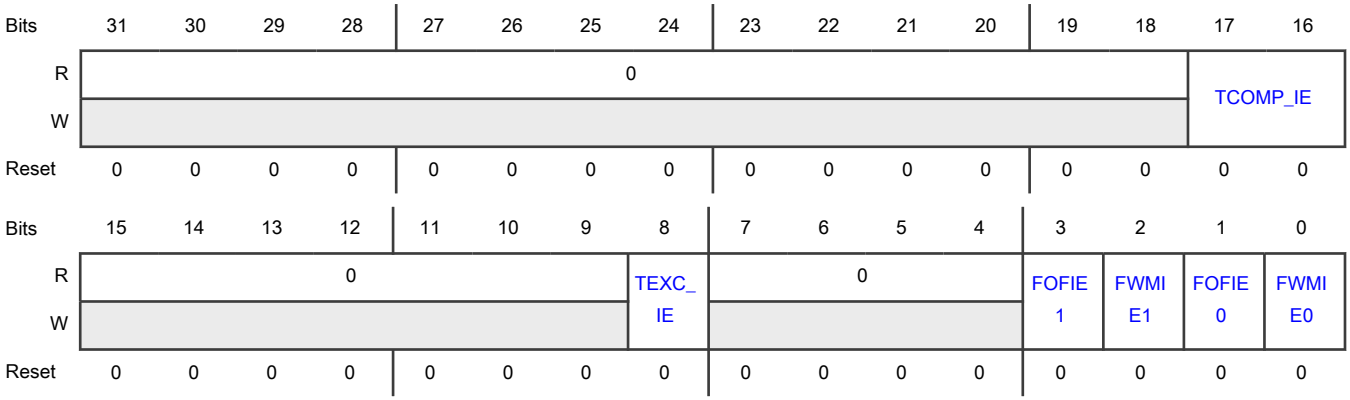
38.6.6 Interrupt Enable (IE)

Offset

Register	Offset
IE	18h

Function
Includes system interrupt masking control fields.

Diagram



Fields

Field	Function
31-18 —	Reserved This field always has the value 0.
17-16 TCOMP_IE	Trigger Completion Interrupt Enable Enables trigger completion interrupts. Each bit in this field corresponds to a trigger source (TCOMP_IE[0] corresponds to trigger 0, TCOMP_IE[1] corresponds to trigger 1, and so on). A trigger completion interrupt is used to indicate when a complete trigger command sequence has been executed. All results are stored in the FIFO when a sequence is considered complete. 00b - Disable 01b - Enable for trigger source 0 only 10b - Enable for trigger source 1 only 11b - Enable trigger 0 and trigger 1 completion interrupts for both trigger sources
15-9 —	Reserved This field always has the value 0.
8 TEXC_IE	Trigger Exception Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Enables ADC to assert an interrupt request when a high-priority trigger exception occurs. TSTAT[TEXC_NUM] contains the value of the corresponding trigger affected by the exception. 0b - Disable 1b - Enable
7-4 —	Reserved
3 FOFIE1	Result FIFO1 Overflow Interrupt Enable Enables ADC to generate overflow interrupt requests when STAT[FOF1] is asserted. This field specifies whether the FIFO 1 overflow interrupts are enabled. 0b - Disable 1b - Enable
2 FWMIE1	FIFO1 Watermark Interrupt Enable Enables ADC to generate watermark interrupt requests when STAT[RDY1] is asserted. This field specifies whether the FIFO1 watermark interrupts are enabled. 0b - Disable 1b - Enable
1 FOFIE0	Result FIFO 0 Overflow Interrupt Enable Enables ADC to generate overflow interrupt requests when the FOF*n* flag is asserted. This field specifies whether the FIFO 0 overflow interrupts are enabled. 0b - Disable 1b - Enable
0 FWMIE0	FIFO 0 Watermark Interrupt Enable Enables ADC to generate watermark interrupt requests when STAT[RDY0] is asserted. This field specifies whether the FIFO 0 watermark interrupts are enabled. 0b - Disable 1b - Enable

38.6.7 DMA Enable (DE)

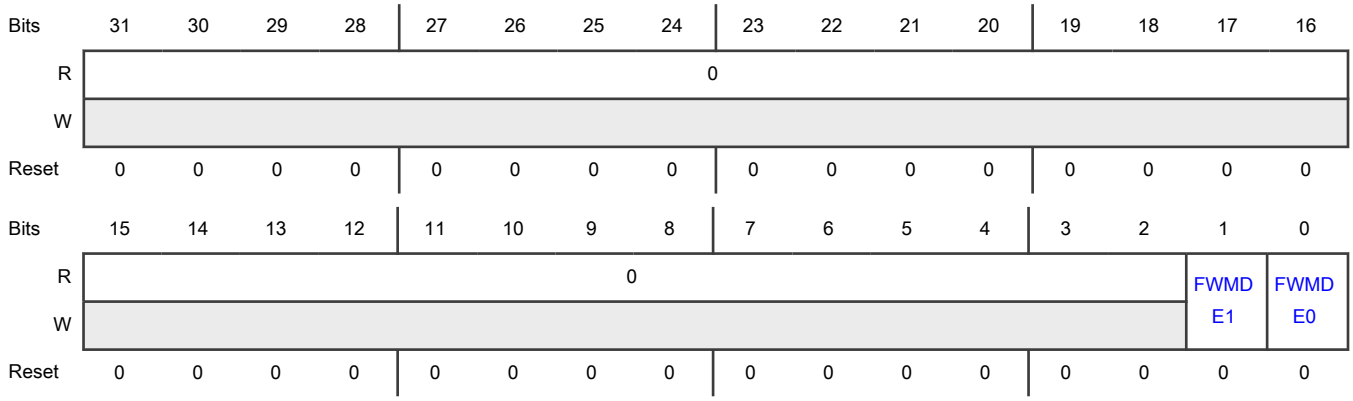
Offset

Register	Offset
DE	1Ch

Function

Includes DMA request masking control fields.

Diagram



Fields

Field	Function
31-2 —	Reserved This bit always has the value 0.
1 FWMDE1	FIFO1 Watermark DMA Enable Enables ADC to generate DMA requests when STAT[RDY1] is asserted. 0b - Disable 1b - Enable
0 FWMDE0	FIFO 0 Watermark DMA Enable Enables ADC to generate DMA requests when STAT[RDY0] is asserted. 0b - Disable 1b - Enable

38.6.8 Configuration (CFG)

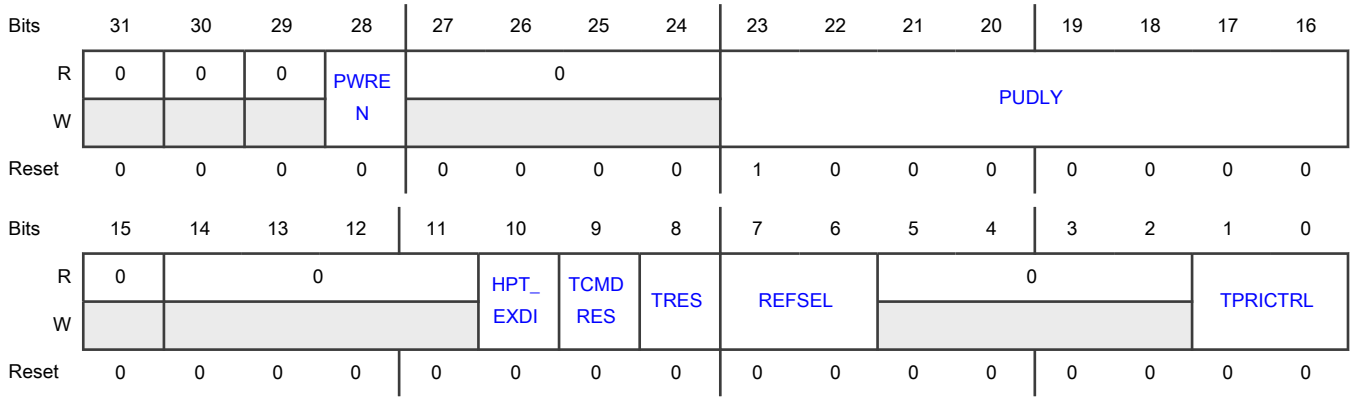
Offset

Register	Offset
CFG	20h

Function

Controls ADC functions that are common to all commands. You cannot make changes to this register when CTRL[ADCEN] is 1. Any writes to this register when CTRL[ADCEN] is 1 are ignored.

Diagram



Fields

Field	Function
31 —	Reserved This bit always has the value 0.
30 —	Reserved This bit always has the value 0.
29 —	Reserved This bit always has the value 0.
28 PWREN	ADC Analog Pre-Enable Enables the ADC analog circuits. When you write 1 to this field, the user code must be delayed for a period exceeding the analog startup time of $t_{ADCSTUP}$ before enabling ADC for operation. ADC is still operational even when CFG[PWREN] = 0 but command execution start is delayed for a period defined by PUDLY. See the chip data sheet for ADC idle and ADC active power consumption parameters. If this field is 0, ADC analog circuits are only enabled when conversions are active. Performance is affected because of analog startup delays. If this field is 1, ADC analog circuits are pre-enabled and ready to execute conversions without startup delays (at the cost of higher DC consumption). **NOTE** A single power up delay (PUDLY) is executed immediately after you write 1 to this field. The detected trigger does not begin ADC operations until the power up delay time has passed. After this initial delay expires, the analog remains pre-enabled and no additional delays are executed. 0b - Not pre-enabled 1b - Pre-enabled and ready to execute conversions
27-24 —	Reserved This bit always has the value 0.
23-16	Power Up Delay

Table continues on the next page...

Table continued from the previous page...

Field	Function
PUDLY	Specifies power up delay. You must program PUDLY to a nonzero value to enable ADC. **NOTE** PUDLY is executed in one of the two modes depending on the value configured in PWREN. If CFG[PWREN] = 0, ADC analog circuits are powered on only when ADC is active. The delay defined by PUDLY is executed after an initial trigger transitions ADC from its Idle state. This delay allows time for the analog circuits to stabilize after being powered on. The startup delay count of (PUDLY×4) ADCK cycles must result in a longer delay than the analog startup time of $t_{ADCSTUP}$. Accuracy of the initial conversion(s) after activation is degraded if PUDLY is set to too small a value. At the end of active conversions, if there are no subsequent pending conversions, the ADC analog is automatically reverted to its low-power idle state and PUDLY runs again after ADC is awakened with a later trigger. If CFG[PWREN] = 1 prior to ADC activation, the analog circuits are pre-enabled and the activation delay defined by PUDLY is executed immediately. After the delay has been executed, the analog remains enabled regardless of ADC activity. This configuration has the benefit of beginning conversions immediately after trigger event detection at the cost of increased DC power consumption of the analog circuits. **NOTE** ADC does not begin an initial conversion until you execute PUDLY regardless of the PWREN setting.
15 —	Reserved This bit always has the value 0.
14-11 —	Reserved This bit always has the value 0.
10 HPT_EXDI	High Priority Trigger Exception Disable Disables high-priority exception. This field controls how higher priority trigger events are handled. See Trigger detect and command execution for a detailed description on trigger exception handling. 0b - Enable 1b - Disable
9 TCMDRES	Trigger Command Resume Determines where to resume from a high-priority trigger exception. CFG[TRES] must be 1 for TCMDRES to be used. If TCMDRES is 1, the trigger sequences interrupted by a high-priority trigger exception are automatically resumed from the command executing before the exception. If TCMDRES is 0, the trigger sequences interrupted by a high-priority trigger exception are automatically restarted from the beginning. **NOTE** The trigger that was interrupted when TCMDRES was 1 resumes at the command it was interrupted and if in LOOP mode, the CMDH*n*[LOOP] count always restarts at the first loop count iteration. 0b - Restart automatically 1b - Resume command execution

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 TRES	Trigger Resume Enable Determines what happens after completing a high-priority trigger exception. If this field is 1, the trigger sequences interrupted by a high-priority trigger exception are automatically resumed or restarted. The sequence can be resumed along command boundaries or restarted from the beginning of a sequence. This is controlled through TCMDRES. If this field is 0, the trigger sequences interrupted by a high-priority trigger exception do not resume or restart automatically. In this case, you can resume interrupted triggers by monitoring STAT[TEXC_INT] (or ISR handling for a trigger exception interrupt when IE[TEXC_IE] is 1). By reading TSTAT[TEXC_NUM], you can determine the trigger that was interrupted and restart the trigger by writing 1 to the corresponding SWTRIG[SWT*n*] field. 0b - Do not resume or restart automatically 1b - Resume or restart automatically
7-6 REFSEL	Voltage Reference Selection Selects the voltage reference high used for conversions. **NOTE** See the chip-specific ADC information on the voltage reference options specific to this packaged device. 00b - Option 1 (default) 01b - Option 2 10b - Option 3 11b - Reserved
5-2 —	Reserved This bit always has the value 0.
1-0 TPRICTRL	ADC Trigger Priority Control Controls the way higher-priority trigger exceptions are handled. See Trigger detect and command execution for more information on trigger event handling. If this field is 0b and a higher-priority trigger is detected during command processing, the current conversion is aborted and the new command specified by the trigger is started. If this field is 1b and a higher-priority trigger is received during command processing, the current command is stopped after completing the current conversion. If averaging is enabled, the averaging loop is completed. However, CMDH*n*[LOOP] is ignored and the higher-priority trigger is serviced. If this field is 10b and a higher-priority trigger is received during command processing, the current command is completed (averaging, looping, comparing) before servicing the higher-priority trigger. 00b - Abort the current conversion and start the new command specified by the trigger 01b - Stop the current command after completing the current conversion 10b - Complete the current command before servicing the higher-priority trigger

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Reserved

38.6.9 Pause (PAUSE)

Offset

Register	Offset
PAUSE	24h

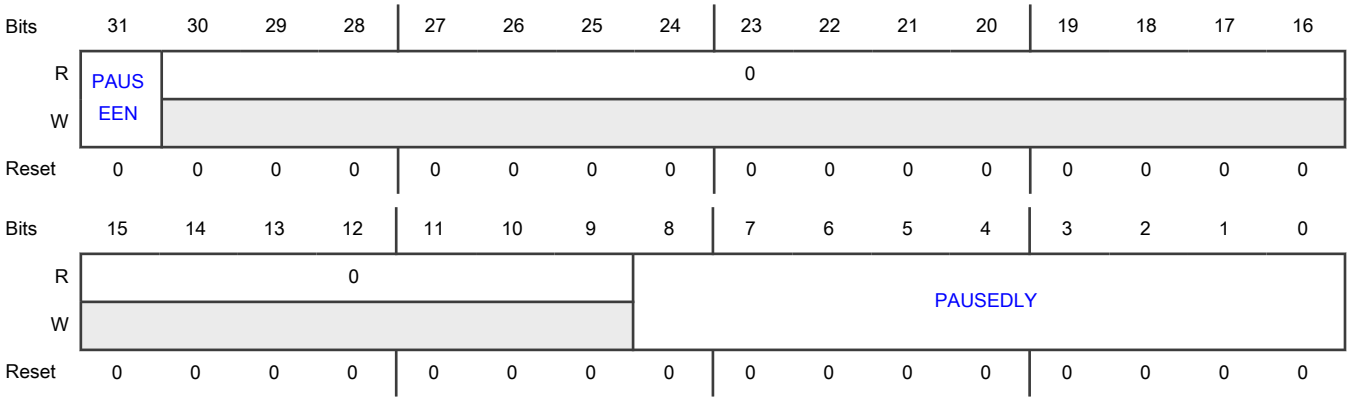
Function

Controls an optionally inserted delay between conversions.

NOTE

Do not make modifications to this register when CTRL[ADCEN] = 1.

Diagram



Fields

Field	Function
31 PAUSEEN	Pause Option Enable Enables the ADC pause function. When enabled, a programmable delay is inserted during command execution sequencing between LOOP iterations, between commands in a sequence, and between conversions when a command is executing for the "Compare Until True" configuration. **NOTE** CMDH*n*[WAIT_TRIG] and Pause (PAUSE) are mutually exclusive. CMDH*n*[WAIT_TRIG] takes priority over Pause (PAUSE) between commands when both are enabled. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
30-9 —	Reserved This bit always has the value 0.
8-0 PAUSEDLY	Pause Delay Controls the duration of pause during command execution sequencing when PAUSEEN = 1. The pause delay is a count of (PAUSEDLY × 4) ADCK cycles.

38.6.10 Software Trigger (SWTRIG)

Offset

Register	Offset
SWTRIG	34h

Function

Initiates software-triggered conversations. Writes to this register are ignored when CTRL[ADCEN] = 0.

NOTE

There is an approximate synchronization delay of three ADC clock cycles in asserting CTRL[ADCEN] until SWTRIG can be accepted.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														0	0
W															SWT1	SWT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved This bit always has the value 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 SWT1	Software Trigger 1 Event Indicates whether a trigger 1 event is generated. Writing 1 to SWT1 is ignored when the trigger 1 event is being serviced or is pending. 0b - Not generated 1b - Generated
0 SWT0	Software Trigger 0 Event Indicates whether a trigger 0 event is generated. Writing 1 to SWT0 is ignored when the trigger 0 event is being serviced or is pending. 0b - Not generated 1b - Generated

38.6.11 Trigger Status (TSTAT)

Offset

Register	Offset
TSTAT	38h

Function

Contains status flags to indicate when trigger sequences are completed or interrupted by a high-priority trigger exception. Each flag in this register is set by hardware and cleared by software.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														TCOMP_FLAG	
W															W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														TEXC_NUM	
W															W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-18 —	Reserved This bit always has the value 0.
17-16 TCOMP_FLAG	Trigger Completion Flag Indicates trigger completion. Each bit corresponds to a trigger. When the corresponding trigger sequence is completed, a bit in TCOMP_FLAG is set. For example, if trigger 1 is completed, then bit 17 is set in TCOMP_FLAG. This register is active only if the corresponding bit in IE[TCOMP_IE] = 1. **NOTE** A synchronization delay may be added to the end of the final conversion in a sequence prior to this flag is set. This delay can range from 2-4 ADC clock cycles. 00b - No trigger is complete (trigger completion interrupts are disabled) 01b - Trigger 0 is complete (trigger 0 has enabled completion interrupts) 10b - Trigger 1 is complete (trigger 1 has enabled completion interrupts) 11b - Trigger 0 and trigger 1 sequences are complete (both triggers have enabled completion interrupts)
15-2 —	Reserved This bit always has the value 0.
1-0 TEXC_NUM	Trigger Exception Number Indicates whether a trigger is interrupted by a high-priority exception. Each bit corresponds to a trigger. When the corresponding trigger sequence is interrupted by a high-priority trigger exception, 1 is set in its bit position. For example, if trigger 1 is interrupted, then bit 1 is set in TEXC_NUM. This register is active regardless of the value in IE[TEXC_IE]. 00b - No trigger is interrupted (or CFG[TRES] = 1) 01b - Trigger 0 is interrupted 10b - Trigger 1 is interrupted 11b - Trigger 0 and trigger 1 sequences are interrupted

38.6.12 Offset Trim 16 Bits (OFSTRIM16)

Offset

Register	Offset
OFSTRIM16	40h

Function

Trims comparator offset for 16-bit ADC conversions (CMDL*n*[MODE] == 0b1). ADC supports an offset calibration function in which this register is updated automatically.

To trigger an automatic update of this register, write 1 to CTRL[CALOFS] with CTRL[CALOFSMODE] = 1. After completing the offset calibration sequence, STAT[CAL_RDY] becomes 1 and OFSTRIM16[OFSTRIM_A] and OFSTRIM16[OFSTRIM_B] are updated with 10-bit signed values.

OFSTRIM16[OFSTRIM_A] and OFSTRIM16[OFSTRIM_B] are used to minimize the offset for normal 16-bit conversions (when CMDL*n*[MODE] = 1). This register set supports a range of -512 to 511. Each increment is 1/2 LSB, resulting in a programmable offset range of -256 to +255.5 LSB for 16-bit conversions.

The OFSTRIM16 trim values are also used for the calibration function.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						OFSTRIM_B									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						OFSTRIM_A									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved This bit always has the value 0.
25-16 OFSTRIM_B	Trim for Offset in B-side Converter for 16-bit Conversions Specifies the trim for offset in B-side converter for 16-bit conversions. OFSTRIM_B is a 10-bit signed value between -512 and 511. This value is subtracted from the raw ADC B-side conversion result.
15-10 —	Reserved
9-0 OFSTRIM_A	Trim for Offset in A-side Converter for 16-bit Conversions Specifies the trim for offset in A-side converter for 16-bit conversions. OFSTRIM_A is a 10-bit signed value between -512 and 511. This value is subtracted from the raw ADC A-side conversion result.

38.6.13 Offset Trim 12 Bits (OFSTRIM12)

Offset

Register	Offset
OFSTRIM12	44h

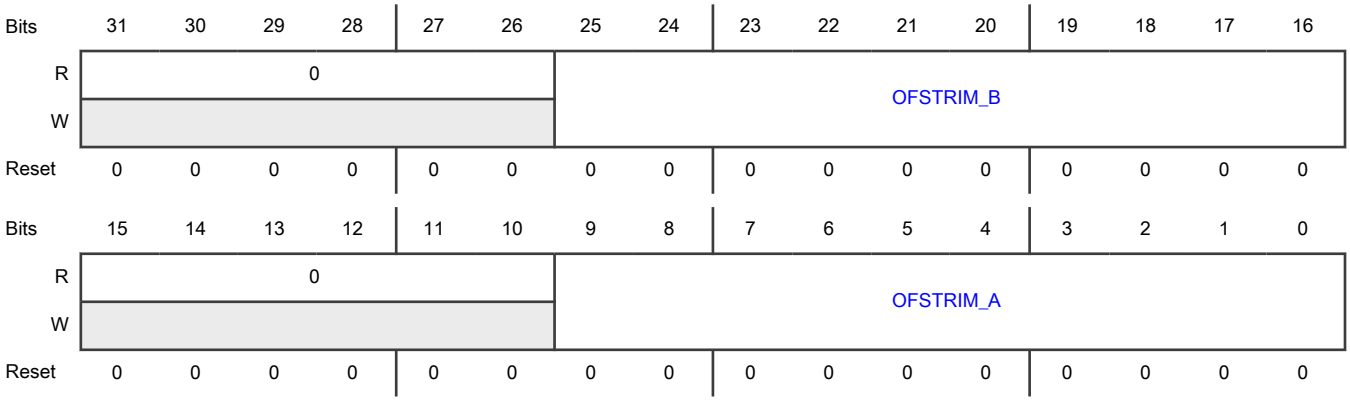
Function

Trims comparator offset for 12-bit ADC conversions (CMDL*n*[MODE] == 0b0). ADC supports an offset calibration function in which this register is updated automatically.

To trigger an automatic update of this register, write 1 to CTRL[CALOFS] with CTRL[CALOFSMODE] = 0b0. On completion of the offset calibration sequence, STAT[CAL_RDY] becomes 1, and OFSTRIM12[OFSTRIM_A] and OFSTRIM12[OFSTRIM_B] are updated with 10-bit signed values.

OFSTRIM12[OFSTRIM_A] and OFSTRIM12[OFSTRIM_B] are used to minimize the offset for normal 12-bit conversions (when CMDL*n*[MODE] = 0). The OFSTRIM registers support a range of -512 to 511. Each increment is 1/32 LSB, resulting in a programmable offset range of -16 to +15.96875 LSB for 12-bit conversions.

Diagram



Fields

Field	Function
31-26 —	Reserved This bit always has the value 0.
25-16 OFSTRIM_B	Trim for Offset in B-side Converter for 12-bit Conversions Specifies the trim for offset in B-side converter for 12-bit conversions. OFSTRIM_B is a 10-bit signed value between -512 and 511. This value is subtracted from the raw ADC B-side conversion result.
15-10 —	Reserved This bit always has the value 0.
9-0 OFSTRIM_A	Trim for Offset in A-side Converter for 12-bit Conversions Specifies the trim for offset in A-side converter for 12-bit conversions. OFSTRIM_A is a 10-bit signed value between -512 and 511. This value is subtracted from the raw ADC A-side conversion result.

38.6.14 Trigger Control (TCTRL0 - TCTRL1)

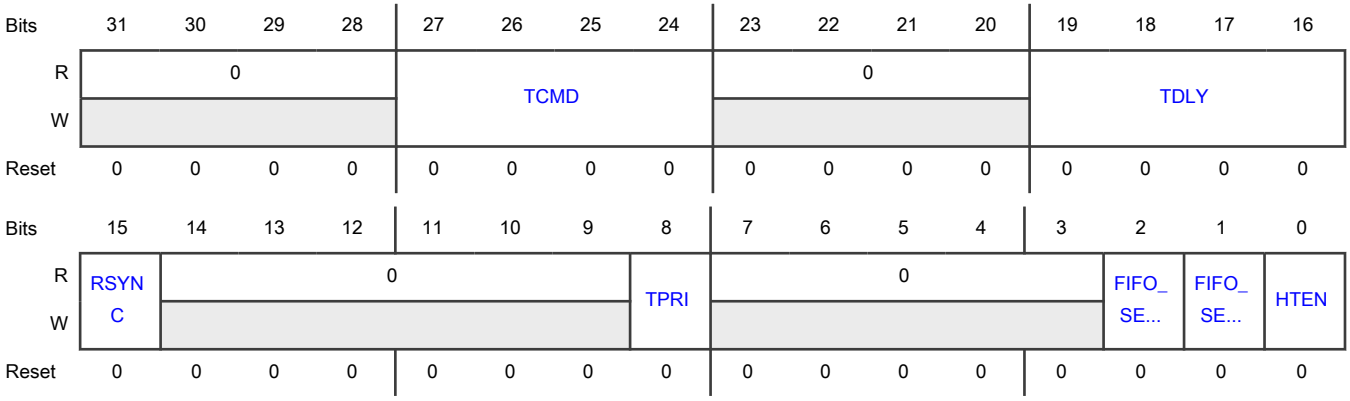
Offset

Register	Offset
TCTRL0	A0h
TCTRL1	A4h

Function

Implements control fields associated with each implemented trigger source. When ADC is actively executing commands, only one of the TCTRL*n* registers is actively controlling ADC conversions. You must not update that register when ADC is active. Any write to TCTRL*n* registers when that trigger control register is controlling the ADC operation can cause unpredictable behavior.

Diagram



Fields

Field	Function
31-28 —	Reserved This bit always has the value 0.
27-24 TCMD	Trigger Command Select Selects the command from command buffer to execute on detection of the associated trigger event. If a trigger is received when ADC is busy converting and the new trigger has a higher priority than the trigger associated with the current command being executed, the command in this register, associated with the new trigger, is executed under the control of CFG[TPRICTRL]. See section Trigger detect and command execution for more information on the relationship between CFG[TPRICTRL] and TCMD. 0000b - Not a valid selection from the command buffer (trigger event is ignored) 0001b - CMD1 is executed 0010b-1110b - Corresponding CMD is executed 1111b - CMD15 is executed
23-20	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	This bit always has the value 0.
19-16 TDLY	Trigger Delay Select Selects the trigger delay duration to wait at the start of servicing a trigger event. Each trigger source has an associated programmable delay before beginning an initial conversion. When this field is 0, then no delay is incurred. When this field is a nonzero value, the duration for the delay is 2^{TDLY} ADCK cycles.
15 RSYNC	Trigger Resynchronization Specifies trigger resynchronization. If this field is 1, the trigger source is used as a resynchronization trigger. You can configure the resynchronization trigger to restart or abort a running trigger sequence (the target). Configure each resynchronization trigger to have a resynchronization target. When a resynchronization trigger is asserted, the target sequence is aborted, the target FIFO destination(s) are cleared, and the target sequence can optionally be restarted depending on CFG[TRES]. Multiple conditions must be met for a resync trigger to execute properly: • The resynchronization trigger must have a higher priority than the resynchronization target. • This field, RSYNC, which is the resynchronization trigger, must be 1. • The resynchronization target, specified by TCTRL*n*[TCMD], must be executing when the resynchronization trigger is asserted. See Resynchronization functionality for more information.
14-9 —	Reserved This bit always has the value 0.
8 TPRI	Trigger Priority Setting Specifies the priority of the associated trigger source. When determining ADC trigger aborts based on a higher priority trigger interrupt, ADC hardware only considers whether the new trigger (TCTRL[TPRI]) is programmed with a lower value than the running trigger. If both triggers are programmed with the same priority, no abort occurs. Also, if two or more triggers are detected with the same TCTRL[TPRI] value, ADC only takes into account the trigger number and order when determining the next active trigger. In such cases, the lower order trigger is set first. 0b - Set to highest priority, level 1 1b - Set to lower priority, level 2
7-3 —	Reserved
2 FIFO_SEL_B	SAR Result Destination for Channel B Specifies the SAR result destination for channel B. This register is used only during dual single-ended mode (CMDL*n*[CTYPE] = 0b11). In this mode, the SAR result from channel B is written to the FIFO number specified by this field. See Sampling modes for more information. 0b - Result written to FIFO 0 1b - Result written to FIFO 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 FIFO_SEL_A	SAR Result Destination for Channel A Specifies the SAR result destination for channel A. The conversion results are stored in the FIFO number specified by this field under the following conditions: • Single-ended mode: CMDL*n*[CTYPE] = 0b01 or CMDL*n*[CTYPE] = 0b00 • Differential mode: CMDL*n*[CTYPE] = 0b10 • Dual single-ended mode: CMDL*n*[CTYPE] = 0b11 This field provides the destination for all single-ended conversions, differential conversions, and A-side conversions in dual single-ended mode. 0b - Result written to FIFO 0 1b - Result written to FIFO 1
0 HTEN	Trigger Enable Enables the hardware trigger source to initiate conversion on the rising-edge of the input trigger source. **NOTE** Enabling a hardware trigger does not disable software triggers. 0b - Disable 1b - Enable

38.6.15 FIFO Control (FCTRL0 - FCTRL1)

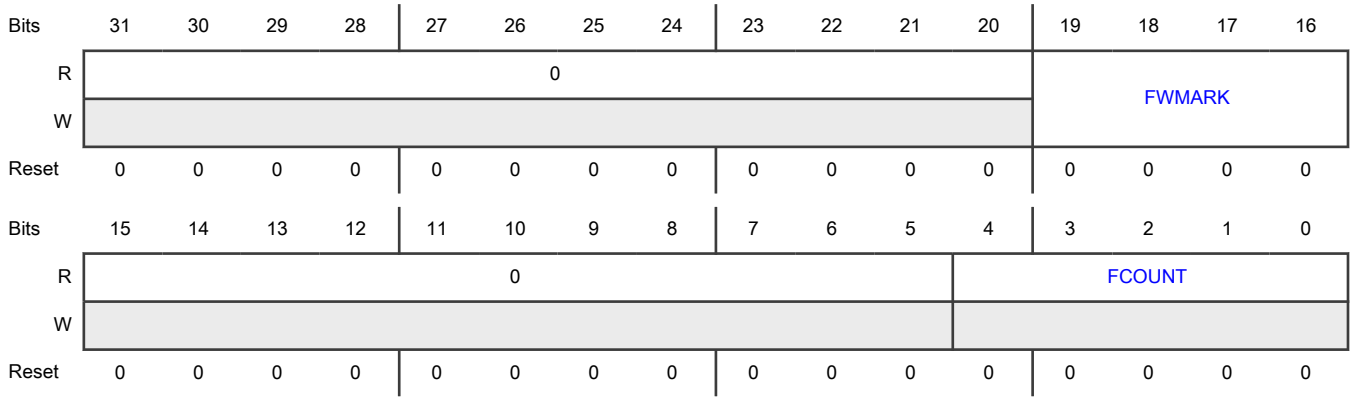
Offset

Register	Offset
FCTRL0	E0h
FCTRL1	E4h

Function

Contains control and status fields for each FIFO. You can set a programmable watermark for each FIFO that can be used to trigger an interrupt. In addition, the number of entries stored in each FIFO can be monitored by reading FCOUNT.

Diagram



Fields

Field	Function
31-20 —	Reserved This field always has the value 0.
19-16 FWMARK	Watermark Level Selection Specifies watermark level selection. FWMARK is a programmable threshold setting. When the number of datawords stored in the ADC result FIFO is greater than the value in FWMARK, the STAT[RDY] flag is asserted to indicate stored data that has reached the programmable threshold. When IE[FWMIE*n*] is 1, an interrupt request is generated. When DE[FWMDE*n*] is 1, a DMA request is generated.
15-5 —	Reserved This field always has the value 0.
4-0 FCOUNT	Result FIFO Counter Indicates the number of data words that are stored in the result FIFO. This value may be used in conjunction with PARAM[FIFOSIZE] to calculate how much room is left in the result FIFO. This field is incremented with each store of new data to the result FIFO and decrements with each read of the result FIFO. The FIFO is reset by writing to CTRL[RSTFIFO*n*], resulting in FCOUNT initialized to 0h.

38.6.16 Gain Calibration Control (GCC0 - GCC1)

Offset

Register	Offset
GCC0	F0h
GCC1	F4h

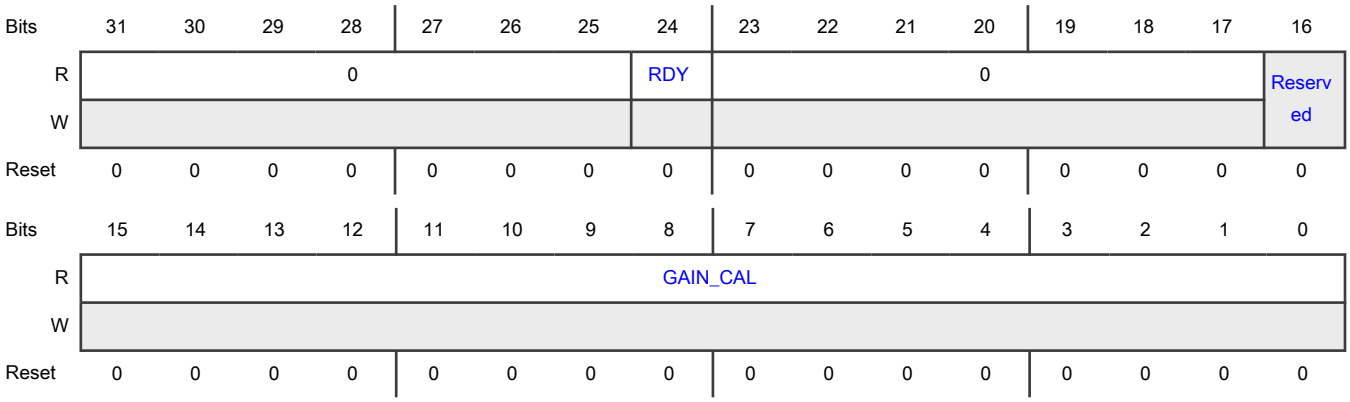
Function

Holds intermediate values that are utilized as part of the calibration steps. GCC0 is associated with the A-side converter, and GCC1 is associated with the B-side converter. ADC calculates GCC*n*[GAIN_CAL] and the field is updated automatically during the calibration steps. After the hardware calibration sequence updates GCC*n*[GAIN_CAL], GCC*n*[RDY] becomes 1 automatically.

NOTE

Requesting a calibration routine automatically writes 0 to GCC*n*[RDY] until the new value for GCC*n*[GAIN_CAL] is calculated. To complete the calibration steps, you need further software processing where you use the value of GCC*n*[GAIN_CAL] to calculate the gain adjustment. The result is stored to GCR*n*[GCALR]. For more information, see Calibration.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 RDY	Hardware Calculated GAIN_CAL Value Ready Indicates whether the data stored in GCC*n*[GAIN_CAL] is valid. If this field is 0, the value of GAIN_CAL is invalid. Run the hardware calibration routine for this value to be set. If this field is 1, the value of GAIN_CAL is valid. You must use GAIN_CAL to derive GCR*n*[GCALR]. 0b - Invalid 1b - Valid
23-17 —	Reserved
16 —	Reserved After a calibration initialization function, this field may read as 0 or 1.
15-0 GAIN_CAL	Gain Calibration Value Indicates the gain calibration value. As part of the hardware calibration steps, GAIN_CAL is automatically updated with a 16-bit signed number and is used in the gain adjustment calculations to derive the value to be written to GCR*n*[GCALR]. For more information, see Calibration.

38.6.17 Gain Calculation Result (GCR0 - GCR1)

Offset

Register	Offset
GCR0	F8h
GCR1	FCh

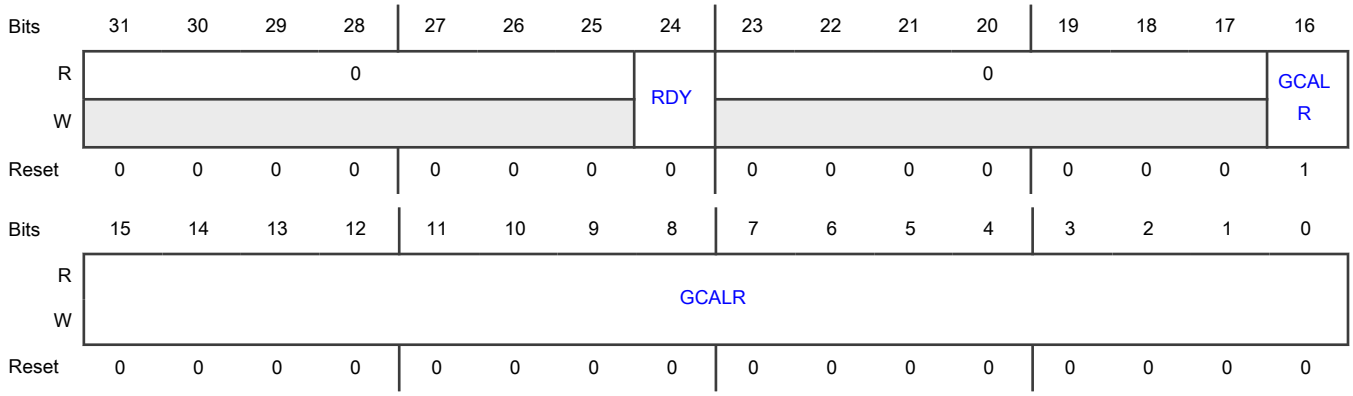
Function

Performs ADC conversions for automated gain error adjustment. GCR0 is associated with the A-side converter, and GCR1 is associated with the B-side converter. Determining the values to store in GCR0[GCALR] and GCR1[GCALR] is the result of software calculations described in Calibration.

GCR*n*[RDY] indicates whether the value of GCR*n*[GCALR] is valid. When you write a value to GCR*n*[GCALR], you must write 1 to GCR*n*[RDY] to indicate that the gain adjustment result is valid.

When you begin the calibration sequence by writing 1 to CTRL[CAL_REQ], the calibration sequence is not complete until GCR*n*[GCALR] is calculated and GCR*n*[RDY] becomes 1. The gain adjustment calculation results in a floating point value between 0 and 2. GCALR[16] holds the integer value (0 or 1), and GCALR[15:0] holds the 16-bit fractional component of the gain adjustment calculation. To convert the value of GCALR back into the gain adjustment value in decimal format, perform this calculation: 1 × GCR*n*[16] + 0.5 × GCR*n*[15] + 0.25 × GCR*n*[14] + 0.125 × GCR*n*[13] +...

Diagram



Fields

Field	Function
31-25 —	Reserved
24 RDY	Gain Calculation Ready Indicates to ADC whether the data stored in GCR*n*[GCALR] is valid and used for gain adjustment. GCALR must be written from memory or calculated each time the calibration routine is run. You must write 1 to this field after writing valid data to GCALR. This field becomes 0 automatically when requesting a new calibration sequence.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Invalid 1b - Valid
23-17 —	Reserved This field always has the value 0.
16-0 GCALR	Gain Calculation Result Holds the 17-bit value generated from the gain adjustment calculation during the calibration steps.

38.6.18 Command Low Buffer (CMDL1 - CMDL15)

Offset

For a = 1 to 15:

Register	Offset
CMDLa	F8h + (a × 8h)

Function

Consists of 15 command buffers (CMD*n*), each constructed from two 32-bit registers (CMDH*n*: CMDL*n*) that can be configured for different channel selections and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of TCTRL*n*[TCMD]. When ADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer must not be updated when ADC is active. A write to a CMD buffer when that CMD buffer is controlling the ADC operation can cause unpredictable behavior.

NOTE

The 15 command buffers are numbered [CMDH1:CMDL1] through [CMDH15:CMDL15]. In NXP-supplied header files, these are likely to be defined as two 15-element arrays that are indexed from 0. The type declaration is:

```
unsigned int CMDH[15];
unsigned int CMDL[15];
```

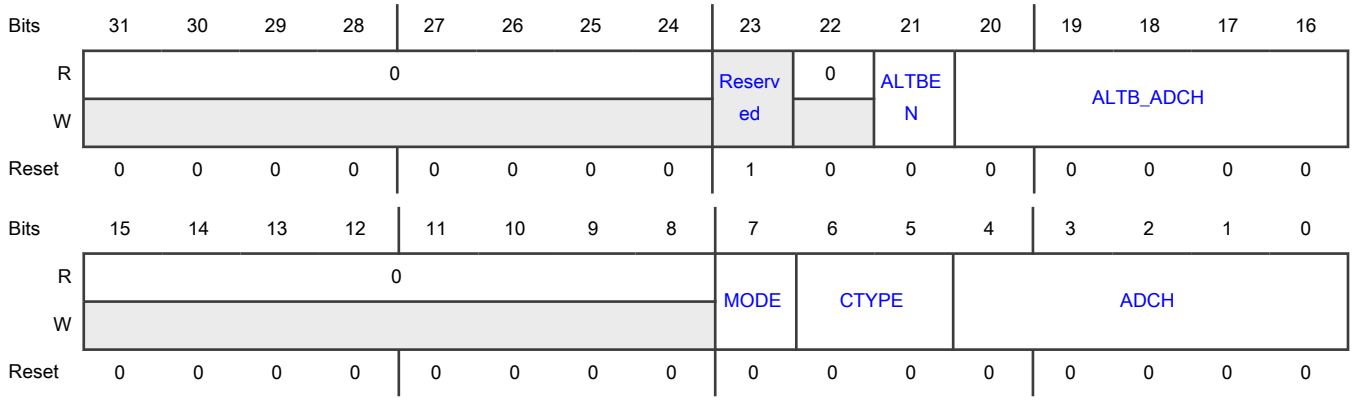
You access ADC0 CMDH1 as `ADC0->CMDH[0]` and ADC0 CMDL15 as `ADC0->CMDL[14]`.

CMDL*n*[ADCH] and CMDL*n*[CTYPE] control the selection of paired or individual input channels. Each ADC command independently makes a channel and conversion type selection. Each ADCH channel selection has an associated A-side and B-side input. You can optionally convert each ADCH pair in a differential mode, but only limited pairs are intended to be converted as differential channels (that is, adjacent pins designed with matched impedance). For the pin pairings available for differential conversions on your chip, see the chip configuration details.

NOTE

Some of the input channels are from on-chip sources such as temperature sensors and reference voltage sources and may only be connected to individual instances of ADC. Some of the input channel options in the field descriptions may not be available for your chip. See the chip-specific information for the channels supported on this chip.

Diagram



Fields

Field	Function
31-24 —	Reserved This field always has the value 0.
23 —	Reserved This field resets to the value 1.
22 —	Reserved This field always has the value 0.
21 ALTBEN	Alternate Channel B Select Enable Enables ALTB_ADCH to select the input for channel B independent of ADCH settings when CTYPE is 0b01 or 0b11. If this field is 0, ALTB_ADCH is disabled. Channel A and channel B inputs are selected based on the settings of ADCH. If this field is 1, ALTB_ADCH is enabled. Channel A inputs are selected by the ADCH setting and channel B inputs are selected by the ALTB_ADCH setting. **NOTE** Writing 10b (differential) to CMDL*n*[CTYPE] works, but is not recommended. 0b - Disable 1b - Enable
20-16 ALTB_ADCH	Alternate Channel B Input Channel Select Selects the alternate channel B input channel. If ALTBEN is 1, ALTB_ADCH selects the channel B input when CTYPE is configured as 0b01 or 0b11. **NOTE** Some of the input channels are from internal resources such as temperature sensors and bandgap voltage sources and may not be connected to individual instances of ADC. Also, some of the input channel options in the field descriptions may not be available for your chip. For the ADC channel assignments on your chip, see the chip-specific information.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0_0000b - Select CH0B 0_0001b - Select CH1B 0_0010b - Select CH2B 0_0011b - Select CH3B 0_0100b-1_1101b - Select the corresponding channel CHnB 1_1110b - Select CH30B 1_1111b - Select CH31B
15-8 —	Reserved
7 MODE	Select Resolution of Conversions Selects the ADC resolution. If this field is 0, standard resolution is selected (single-ended 12-bit conversion; differential 13-bit conversion with two's-complement output). If this field is 1, high resolution is selected (single-ended 16-bit conversion; differential 16-bit conversion with two's-complement output). 0b - Standard resolution 1b - High resolution
6-5 CTYPE	Conversion Type Specifies how a pair of A and B channels are converted. 00b - Single-ended mode (only the A-side channel is converted) 01b - Single-ended mode (only the B-side channel is converted) 10b - Differential mode (A-B) 11b - Dual-Single-Ended mode (both A-side and B-side channels are converted independently)
4-0 ADCH	Input Channel Select Selects ADCH channel pair with an associated A-side and B-side input. The ADCH setting, in conjunction with CTYPE settings, selects the input from one of the channel inputs or input pairs. **NOTE** Not all pairs are intended to be converted as differential channels. For the pin pairings available for differential conversions on your chip, see the chip-specific information. **NOTE** Some of the input channels are from internal resources such as temperature sensors and bandgap voltage sources and may only be connected to individual instances of the ADC module. Some of the input channel options in the field descriptions may not be available for your chip. See the chip-specific information for the ADC channel assignments on your chip.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0_0000b - Select CH0A or CH0B or CH0A/CH0B pair 0_0001b - Select CH1A or CH1B or CH1A/CH1B pair 0_0010b - Select CH2A or CH2B or CH2A/CH2B pair 0_0011b - Select CH3A or CH3B or CH3A/CH3B pair 0_0100b-1_1101b - Select corresponding channel CHnA or CHnB or CHnA/CHnB pair 1_1110b - Select CH30A or CH30B or CH30A/CH30B pair 1_1111b - Select CH31A or CH31B or CH31A/CH31B pair

38.6.19 Command High Buffer (CMDH1 - CMDH4)

Offset

Register	Offset
CMDH1	104h
CMDH2	10Ch
CMDH3	114h
CMDH4	11Ch

Function

Consists of 15 command buffers (CMD*n*), each constructed from two 32-bit registers (CMDH*n*: CMDL*n*) that can be configured for different channel selections and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of TCTRL*n*[TCMD]. When ADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer must not be updated while ADC is active. A write to a CMD buffer while that CMD buffer is controlling ADC operation can result in unpredictable behavior.

NOTE

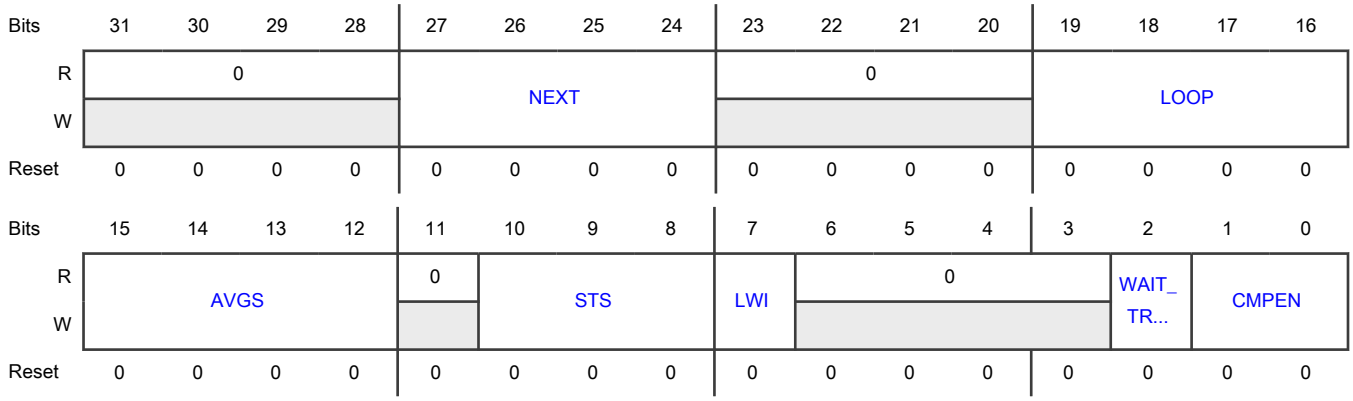
The 15 command buffers are numbered [CMDH1:CMDL1] through [CMDH15:CMDL15]. In NXP-supplied header files, these are likely to be defined as two 15-element arrays that are indexed from 0. The type declaration is:

```
unsigned int CMDH[15];

unsigned int CMDL[15];
```

You access ADC0 CMDH1 as `ADC0->CMDH[0]` and ADC0 CMDL15 as `ADC0->CMDL[14]`.

Diagram



Fields

Field	Function
31-28 —	Reserved This field always has the value 0.
27-24 NEXT	Next Command Select Selects the next command to execute after the current command completes. You can configure multiple commands in a scan configuration by linking the next command in a daisy-chain sequence. The command buffer number is not indicative of any particular order, and the order of execution is strictly controlled by this field (for example, a sequence of commands could be CMD2 to CMD1 to CMD3). Unending circular command execution is allowed by writing 1 to this field in the last command in a sequence to the first command in the sequence. It also allows a command to set the next command to itself, resulting in a continuous conversion configuration. Setting the next command to 0h causes conversions to terminate after command completion. Lower-priority trigger events cannot be serviced until a higher-priority triggered command (or sequence of commands) completes. When this field is 0, no next command is defined. The field terminates conversions at the completion of the current command. If the lower-priority trigger is pending, the command associated with the lower-priority trigger begins. 0000b - No next command defined 0001b - CMD1 command buffer register 0010b-1110b - Corresponding CMD command buffer register 1111b - CMD15 command buffer register
23-20 —	Reserved This field always has the value 0.
19-16 LOOP	Loop Count Select Selects the number of times this command executes (and stores conversion result to RESFIFO) before completion and transition to the next command or Idle state. LWI controls whether a single channel is converted on each iteration or the auto-channel increment results in a channel scanning functionality.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000b - Looping not enabled; command executes one time 0001b - Loop one time; command executes two times 0010b - Loop two times; command executes three times 0011b-1110b - Loop corresponding number of times; command executes LOOP + 1 times 1111b - Loop 15 times; command executes 16 times
15-12 AVGS	Hardware Average Select Selects the number of ADC conversions averaged to create the ADC result (2^{AVGS}). During the execution of average iterations, an internal storage buffer is used to capture temporary results. Hardware averaging is a nested loop control and does not extend across LOOP boundaries. See Functional description to understand the usage of AVGS, LOOP, and NEXT in command execution sequencing. 0000b - Single conversion 0001b - 2 conversions 0010b - 4 conversions 0011b - 8 conversions 0100b - 16 conversions 0101b - 32 conversions 0110b - 64 conversions 0111b - 128 conversions 1000b - 256 conversions 1001b - 512 conversions 1010b - 1024 conversions
11 —	Reserved This field always has the value 0.
10-8 STS	Sample Time Select Selects the total sample time. If you write 0 to this field, the minimum sample time of 3.5 ADCK cycles is selected. When STS is programmed to a nonzero value, the sample time is ($3.5 + 2^{STS}$) ADCK cycles. The shortest sample time maximizes conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be sampled accurately. You can use longer sample times to lower the overall power consumption if you configure command looping and sequencing and high conversion rates are not required. 000b - Minimum sample time of 3.5 ADCK cycles 001b - $3.5 + 2^1$ ADCK cycles; 5.5 ADCK cycles 010b - $3.5 + 2^2$ ADCK cycles; 7.5 ADCK cycles 011b - $3.5 + 2^3$ ADCK cycles; 11.5 ADCK cycles 100b - $3.5 + 2^4$ ADCK cycles; 19.5 ADCK cycles

Table continues on the next page...

Table continued from the previous page...

Field	Function
	101b - 3.5 + 2^5 ADCK cycles; 35.5 ADCK cycles 110b - 3.5 + 2^6 ADCK cycles; 67.5 ADCK cycles 111b - 3.5 + 2^7 ADCK cycles; 131.5 ADCK cycles
7 LWI	Loop with Increment Enables auto-channel increment. If this field is 0, LOOP selects the number of times the selected channel is converted consecutively. When LWI = 1, auto channel incrementing is enabled and the LOOP field defines how many consecutive channels are converted as part of the command execution. Example 1: LOOP = 8h, LWI = 0b0, CMDL*n*[CTYPE] = 0h, CMDL*n*[ADCH] = Dh. Convert on channel 13A 9 times. Example 2: LOOP = 8h, LWI = 0b1, CMDL*n*[CTYPE] = 0h, CMDL*n*[ADCH] = Dh. Run channels 13A, 14A, 15A, ... 21A each one time. Maximum channel scanning using a single command buffer is then defined by the maximum value of LOOP[16]. 0b - Disable 1b - Enable
6-3 —	Reserved
2 WAIT_TRIG	Wait for Trigger Assertion Before Execution Specifies whether commands are executed automatically or an active trigger must be received before executing this command. If WAIT_TRIG is 1, wait states are added before the command until the active trigger is asserted again, which must happen before executing this command. If WAIT_TRIG is 0, each command is automatically executed when called. 0b - Execute automatically 1b - Receive trigger before execution
1-0 CMPEN	Compare Function Enable Enables the compare function. This field guides the operation of the automatic compare function to optionally store the compare result only when the compare function is true, after an ADC channel input is sampled and converted and any averaging iterations are performed. When the compare function is enabled, the conversion result is compared to the compare value registers (CV*n*[CVH] and CV*n*[CVL]). There are multiple options on command sequencing related to the compare function. See Compare function for a detailed explanation of the compare options. **NOTE** Not all command buffers implement CMPEN, which is only available in command buffers that have a corresponding Compare Value register. 00b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - Reserved 10b - Enable; store on true 11b - Enable; repeat channel acquisition (sample, convert, and compare) until true

38.6.20 Command High Buffer (CMDH5 - CMDH15)

Offset

For a = 5 to 15:

Register	Offset
CMDHa	FCh + (a × 8h)

Function

Consists of 15 command buffers (CMD*n*), each constructed from two 32-bit registers (CMDH*n*: CMDL*n*) that can be configured for different channel selections and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of TCTRL*n*[TCMD]. When ADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer must not be updated while ADC is active. A write to a CMD buffer while that CMD buffer is controlling ADC operation can result in unpredictable behavior.

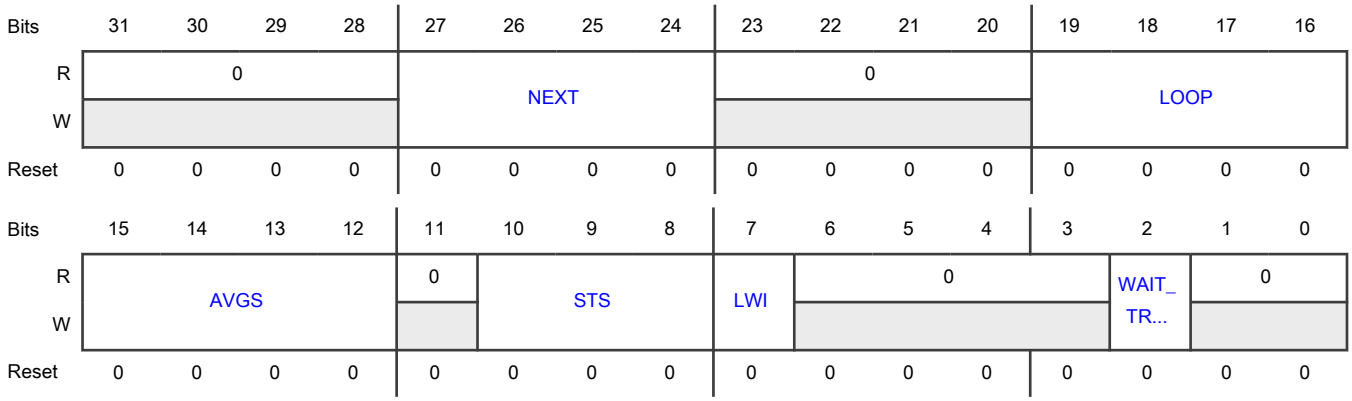
NOTE

The 15 command buffers are numbered [CMDH1:CMDL1] through [CMDH15:CMDL15]. In NXP-supplied header files, these are likely to be defined as two 15-element arrays that are indexed from 0. The type declaration is:

```
unsigned int CMDH[15];
unsigned int CMDL[15];
```

You access ADC0 CMDH1 as `ADC0->CMDH[0]` and ADC0 CMDL15 as `ADC0->CMDL[14]`.

Diagram



Fields

Field	Function
31-28 —	Reserved This field always has the value 0.
27-24 NEXT	Next Command Select Selects the next command to execute after the current command completes. You can configure multiple commands in a scan configuration by linking the next command in a daisy-chain sequence. The command buffer number is not indicative of any particular order, and the order of execution is strictly controlled by this field (for example, a sequence of commands could be CMD2 to CMD1 to CMD3). Unending circular command execution is allowed by writing 1 to this field in the last command in a sequence to the first command in the sequence. It also allows a command to set the next command to itself, resulting in a continuous conversion configuration. Setting the next command to 0h causes conversions to terminate after command completion. Lower-priority trigger events cannot be serviced until a higher-priority triggered command (or sequence of commands) completes. When this field is 0, no next command is defined. The field terminates conversions at the completion of the current command. If the lower-priority trigger is pending, the command associated with the lower-priority trigger begins. 0000b - No next command defined 0001b - CMD1 command buffer register 0010b-1110b - Corresponding CMD command buffer register 1111b - CMD15 command buffer register
23-20 —	Reserved This field always has the value 0.
19-16 LOOP	Loop Count Select Selects the number of times this command executes (and stores conversion result to RESFIFO) before completion and transition to the next command or Idle state. LWI controls whether a single channel is converted on each iteration or the auto-channel increment results in a channel scanning functionality. 0000b - Looping not enabled; command executes one time 0001b - Loop one time; command executes two times 0010b - Loop two times; command executes three times 0011b-1110b - Loop corresponding number of times; command executes LOOP + 1 times 1111b - Loop 15 times; command executes 16 times
15-12 AVGS	Hardware Average Select Selects the number of ADC conversions averaged to create the ADC result (2^{AVGS}). During the execution of average iterations, an internal storage buffer is used to capture temporary results. Hardware averaging is a nested loop control and does not extend across LOOP boundaries. See Functional description to understand the usage of AVGS, LOOP, and NEXT in command execution sequencing. 0000b - Single conversion

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0001b - 2 conversions 0010b - 4 conversions 0011b - 8 conversions 0100b - 16 conversions 0101b - 32 conversions 0110b - 64 conversions 0111b - 128 conversions 1000b - 256 conversions 1001b - 512 conversions 1010b - 1024 conversions
11 —	Reserved This field always has the value 0.
10-8 STS	Sample Time Select Selects the total sample time. If you write 0 to this field, the minimum sample time of 3.5 ADCK cycles is selected. When STS is programmed to a nonzero value, the sample time is (3.5 + 2^{STS}) ADCK cycles. The shortest sample time maximizes conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be sampled accurately. You can use longer sample times to lower the overall power consumption if you configure command looping and sequencing and high conversion rates are not required. 000b - Minimum sample time of 3.5 ADCK cycles 001b - 3.5 + 2^1 ADCK cycles; 5.5 ADCK cycles 010b - 3.5 + 2^2 ADCK cycles; 7.5 ADCK cycles 011b - 3.5 + 2^3 ADCK cycles; 11.5 ADCK cycles 100b - 3.5 + 2^4 ADCK cycles; 19.5 ADCK cycles 101b - 3.5 + 2^5 ADCK cycles; 35.5 ADCK cycles 110b - 3.5 + 2^6 ADCK cycles; 67.5 ADCK cycles 111b - 3.5 + 2^7 ADCK cycles; 131.5 ADCK cycles
7 LWI	Loop with Increment Enables auto-channel increment. If this field is 0, LOOP selects the number of times the selected channel is converted consecutively. When LWI = 1, auto channel incrementing is enabled and the LOOP field defines how many consecutive channels are converted as part of the command execution. Example 1: LOOP = 8h, LWI = 0b0, CMDL*n*[CTYPE] = 0h, CMDL*n*[ADCH] = Dh. Convert on channel 13A 9 times. Example 2: LOOP = 8h, LWI = 0b1, CMDL*n*[CTYPE] = 0h, CMDL*n*[ADCH] = Dh. Run channels 13A, 14A, 15A, ... 21A each one time.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Maximum channel scanning using a single command buffer is then defined by the maximum value of LOOP[16]. 0b - Disable 1b - Enable
6-3 —	Reserved
2 WAIT_TRIG	Wait for Trigger Assertion Before Execution Specifies whether commands are executed automatically or an active trigger must be received before executing this command. If WAIT_TRIG is 1, wait states are added before the command until the active trigger is asserted again, which must happen before executing this command. If WAIT_TRIG is 0, each command is automatically executed when called. 0b - Execute automatically 1b - Receive trigger before execution
1-0 —	Reserved

38.6.21 Compare Value (CV1 - CV4)

Offset

Register	Offset
CV1	200h
CV2	204h
CV3	208h
CV4	20Ch

Function

Contains the value used to compare the conversion result when the compare function is enabled. This register is formatted in the same way as RESFIFO*n*[D] in different modes of operation for both bit position definition and value format using unsigned or signed two's-complement. This register has a direct association with the specific command buffer register (that is, CV1 is used only during the execution of the CMD1 command).

NOTE

The compare function is only available for command buffers that have a corresponding Compare Value register.

When ADC is actively executing commands, Compare Value (CV1 - CV4) associated with the active command (CMD*n*) must not be updated. Writes to the associated CV*n* register during this time can result in unpredictable behavior.

NOTE

The 4 compare value registers are numbered [CV1] through [CV4]. In NXP-supplied header files, this is likely to be defined as a 4-element array that is indexed from 0. Therefore, the type declaration is:

```
unsigned int CV[4];
```

You must access ADC0 CV1 as `ADC0->CV[0]` and ADC0 CV4 as `ADC0->CV[3]`.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CVH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CVL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 CVH	Compare Value High Checks whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVH is used.
15-0 CVL	Compare Value Low Checks whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVL is used.

38.6.22 Data Result FIFO (RESFIFO0 - RESFIFO1)

Offset

Register	Offset
RESFIFO0	300h
RESFIFO1	304h

Function

Stores the data result of ADC conversions. RESFIFO is a 16 entry FIFO. Additionally, several tag fields of the source command and trigger information are stored along with the data. FCTRL*n*[FCOUNT] indicates the number of valid data words stored in

RESFIFO. Reading RESFIFO provides the oldest unread data word entry in the FIFO and decrements FCTRL*n*[FCOUNT]. The FIFO can be emptied by successive reads of RESFIFO. Write 1 to CTRL[RSTFIFO*n*] to reset the FIFO.

The following table describes the format of data in result FIFO in the different modes of operation. The sign bit is the MSB in signed two's-complement modes. For example, when configured for 12-bit single-ended mode, D[15] and D[2:0] become 0. When configured for 13-bit differential mode, D[15] is the sign bit and D[2:0] becomes 0.

Table 300. Data result register format description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed two's-complement, JLEFT = X
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned, 16-bit magnitude, JLEFT = X
13-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	Signed two's-complement, left-justified, zero extended, JLEFT = X
12-bit single-ended	0	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	Unsigned, JLEFT = 0, zero in D[15] and D[2:0]
12-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	0	Unsigned, JLEFT = 1, zero in D[3:0]

NOTE

S: Sign bit

D: Data, which is two's-complement data if indicated

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALID	0			CMDSRC				LOOPCNT				0			TSRC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VALID	FIFO Entry is Valid Indicates whether the FIFO entry is valid. This field becomes 1 when the FIFO is holding data. Also, the FIFO record read from RESFIFO is valid. When the FIFO is empty, the field becomes 0 and any read from RESFIFO is discarded. 0b - Not valid 1b - Valid
30-28 —	Reserved This field always has the value 0.
27-24 CMDSRC	Command Buffer Source Indicates the command buffer used as control settings for this conversion. If this field is 0, 0h is only found in the initial FIFO state prior to an ADC conversion result dataword being stored in a RESFIFO buffer. 0000b - Not a valid CMDSRC value for a dataword in RESFIFO 0001b - CMD1 0010b-1110b - Corresponding command 1111b - CMD15
23-20 LOOPCNT	Loop Count Value Indicates the loop count value during command execution that generated this result. If CMDH*n*[LOOP] is a nonzero value, the command execution stores off a result multiple times during command execution (at the loop boundary). 0000b - Initial conversion 0001b - Second conversion 0010b-1110b - LOOPCNT + 1 conversion 1111b - 16th conversion
19-17 —	Reserved This field always has the value 0.
16 TSRC	Trigger Source Indicates the trigger source that initiated a conversion and generated this result. When multiple commands are chained together using CMDH*n*[NEXT], the TSRC value for all datawords stored in Data Result FIFO (RESFIFO0 - RESFIFO1) indicates the trigger source that started the sequence of commands. 0b - Trigger source 0 1b - Trigger source 1
15-0	Data Result

Table continues on the next page...

Table continued from the previous page...

Field	Function
D	Indicates the data result. The formatting of the data in this field is summarized in Table 300.

38.6.23 Calibration General A-Side (CAL_GAR0 - CAL_GAR32)

Offset

For a = 0 to 32:

Register	Offset
CAL_GARa	400h + (a × 4h)

Function

Corrects linearity errors of the A-side converter. This set of registers contains calibration information that is automatically updated during the self-calibration sequence. See Calibration for more information on completing ADC calibration steps.

The calibration values in this register set affect the end conversion result by conditionally being subtracted from the conversion before the end result is transferred into the FIFOs. Calibration must be run each time ADC is powered down or a hard reset is issued. To reduce the latency required to run calibration, the values in this register must be stored in a nonvolatile memory after an initial calibration and recovered prior to the first ADC conversion. If these registers are set to values that are not generated by the calibration function, the linearity error specifications may not be met.

This CAL_GAR set of registers is only read-write accessible when ADC is disabled with CTRL[ADCEN] = 0b0. When writing to these registers, the access time is larger than three ADC clock cycles. Wait states are inserted on the bus to meet synchronization timing to the associated CAL_GAR register.

Each CAL_GAR register has a 16-bit signed value with a nonuniform supported range of values. The valid range of each register is summarized in Calibration general A-side and B-side register valid ranges.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAL_GAR_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved This field always has the value 0.
15-0 CAL_GAR_VAL	Calibration General A-Side Register Element Specifies a 16-bit signed value with a nonuniform supported range of values.

38.6.24 Calibration General B-Side (CAL_GBR0 - CAL_GBR32)

Offset

For a = 0 to 32:

Register	Offset
CAL_GBRa	500h + (a × 4h)

Function

Corrects linearity errors of the B-side converter. This set of registers contains calibration information that is automatically updated during the self-calibration sequence. See Calibration for more information on completing ADC calibration steps.

The calibration values in the CAL_GBR registers affect the end conversion result by conditionally being subtracting from the conversion before the end result is transferred into the FIFOs. Calibration must be run each time ADC is powered down or a hard reset is issued. To reduce the latency required to run calibration, the CAL_GBR values can be stored in a nonvolatile memory after an initial calibration and recovered prior to the first ADC conversion. If you set these registers to values that are not generated by the calibration function, the linearity error specifications may not be met.

The CAL_GBR registers are read-write accessible only when ADC is disabled with CTRL[ADCEN] = 0b0.

NOTE

- The access time when writing to these registers is greater than three ADC clock cycles. Wait states are inserted on the bus to meet synchronization timing to the associated CAL_GBR register.
- The value in each CAL_GBR register is a 16-bit signed value but has nonuniform supported range of values. The valid range of each register is summarized in Calibration general A-side and B-side register valid ranges.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAL_GBR_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved This field always has the value 0.
15-0 CAL_GBR_VAL	Calibration General B-Side Register Element Specifies a 16-bit signed value with a nonuniform supported range of values.

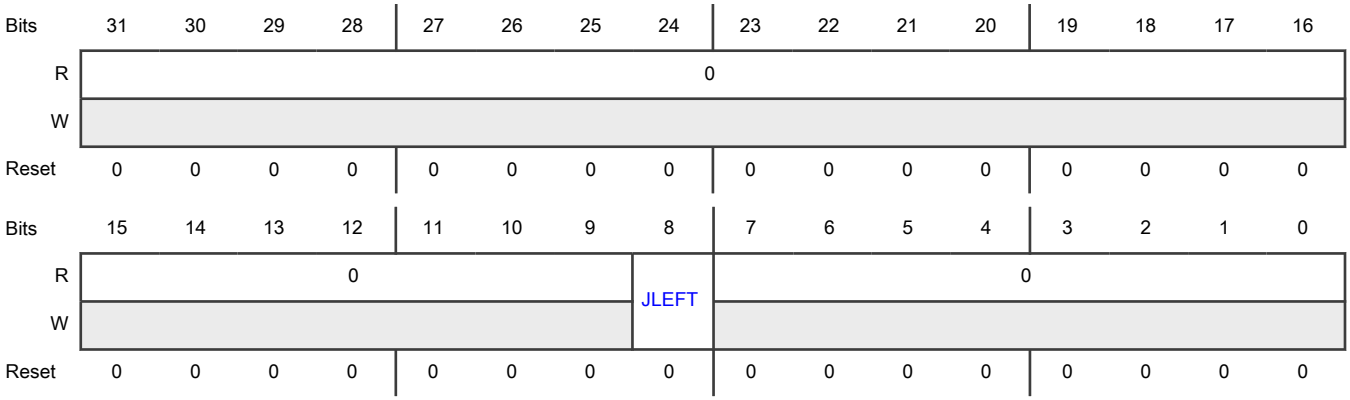
38.6.25 Configuration 2 (CFG2)

Offset

Register	Offset
CFG2	FF8h

Function
Controls ADC functions that are common to all commands.

Diagram



Fields

Field	Function
31-9 —	Reserved This field always has the value 0.
8 JLEFT	Justified Left Enable Enables the data to be stored in Data Result FIFO (RESFIFO0 - RESFIFO1) in left-justified format, in 12-bit single-ended mode. The formatting for the data in RESFIFO*n*[D] is summarized in Table 300.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When CTRL[ADCEN] is 1, you cannot change this field. Any writes are ignored. If this field is 0, for 12-bit single-ended conversions, the RESFIFO data format is in the standard format with data presented in RESFIFO*n*[D][14:3]. If this field is 1, for 12-bit single-ended conversions, the RESFIFO data format is left-justified. **NOTE** In JLEFT mode, you must program CV*n*[CVL] and CV*n*[CVH] in the left-justify format to generate the correct compare results. 0b - Standard format 1b - Left-justified format
7-0 —	Reserved This field always has the value 0.

Chapter 39
Comparator (ACMP)

39.1 Chip-specific ACMP information

Table 301. Reference links to related information

Topic	Related module	Reference
Full description	ACMP	ACMP
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

39.1.1 Module instances

This device has one instance of the ACMP module, ACMP0.

Table 302. ACMP instance

Features	ACMP0
Located in	VDD1_SENSE
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

39.1.2 External pin signals

In order to use the CMP's analog input comparison function, you need to enable the analog function of the related pins by writing 1 to the IOPCTL1_PIO10_xx[AMENA].

The following table lists the pins that can be configured as CMP analog inputs.

Table 303. ACMP external pin signals

Chip pin	Module external signal
PIO10_8/CMP0_INA	INA_1P8_1
PIO10_9/CMP0_INB	INA_1P8_2
PIO10_16/CMP0_INC	INA_1P8_3
PIO10_17/CMP0_IND	INA_1P8_4

39.1.3 Internal input 0

In this chip, the internal positive input 0 and internal negative input 0 are not connected, 000b in C1[PSEL] and C1[MSEL] is reserved.

39.1.4 DAC supply voltage reference source

The following table lists the supply voltage reference sources.

Table 304. DAC supply voltage reference source

Voltage reference (Vin)	Source
Vin1	Unused
Vin2	VDD1V8

39.1.5 Initialization

Initial configuration of a comparator peripheral is accomplished as follows:

- Enable the clock to the ACMP0 register interface by writing 1 to the CLKCTL3_PSCCTL0_COMP[ACMP0] or CLKCTL3_PSCCTL0_COMP_SET[ACMP0].
- Clear the ACMP0 reset by writing 0 to the RSTCTL3_PRSTCTL1[ACMP0] or writing 1 to the RSTCTL3_PRSTCTL1_CLR[ACMP0].

39.2 Overview

ACMP includes the following submodules:

- High-speed comparator (CMP) (see CMP block diagram)
- Digital-to-analog converter (DAC) (see DAC block diagram)
- Analog mux (ANMUX) (see ANMUX key features)

CMP provides a circuit to compare two analog input voltages. The comparator circuit operates across the full range of the supply voltage, known as rail-to-rail operation.

ANMUX provides a circuit to select an analog input signal from eight channels. DAC provides one signal. The mux circuit operates across the full range of the supply voltage.

DAC is a 256-tap resistor ladder network that provides a selectable voltage reference for applications requiring a voltage reference. It divides the supply reference, V_{in}, into 256 voltage levels. An 8-bit digital register input selects the output voltage level, which varies from V_{in} to $V_{in}/256$. You can select V_{in} from two voltage sources: V_{in1} or V_{in2}.

39.2.1 ACMP block diagram

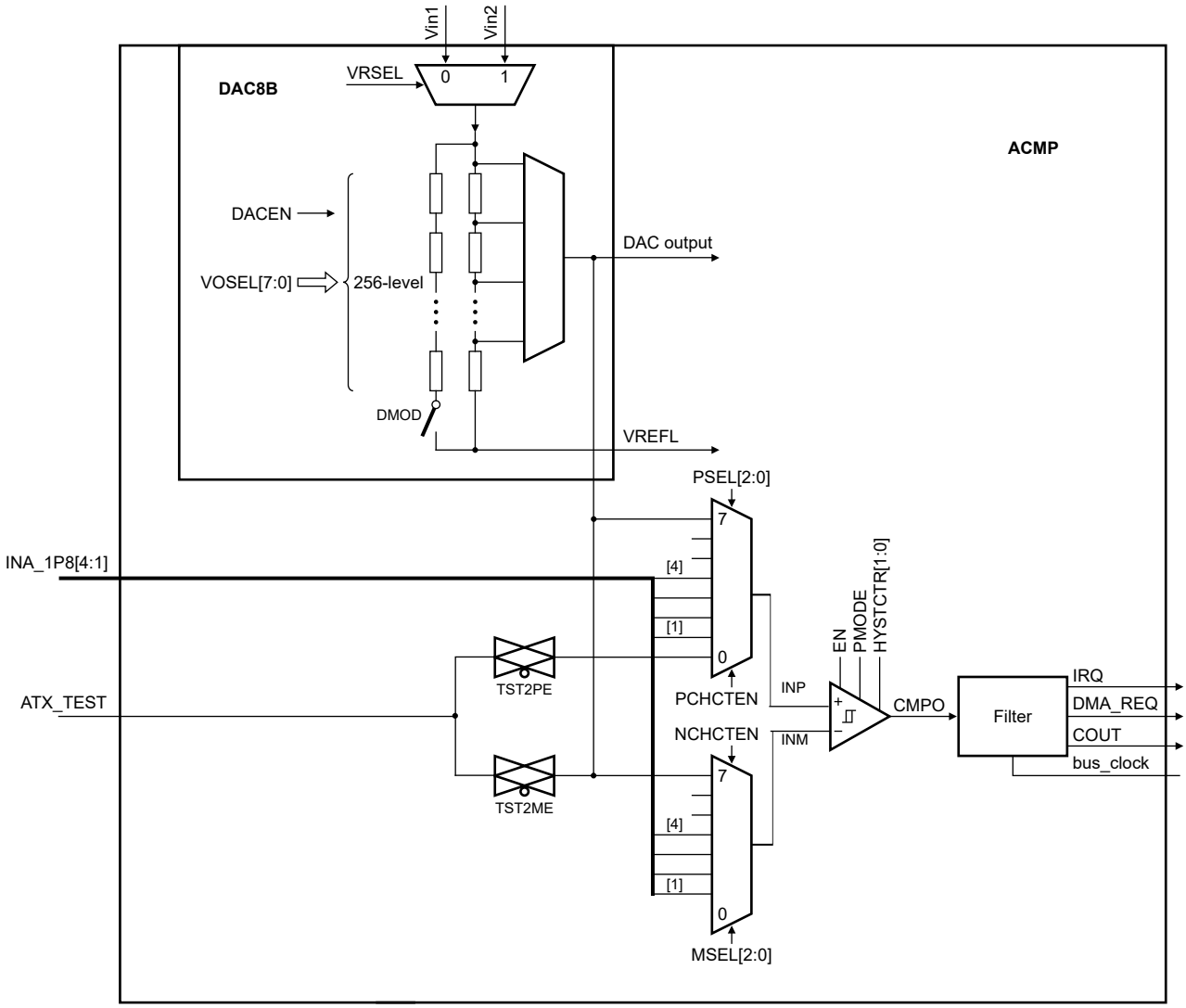


Figure 119. ACMP block diagram

39.2.2 CMP block diagram

In Figure 120 CMP:

- Bypasses the filter block when it is not in use.

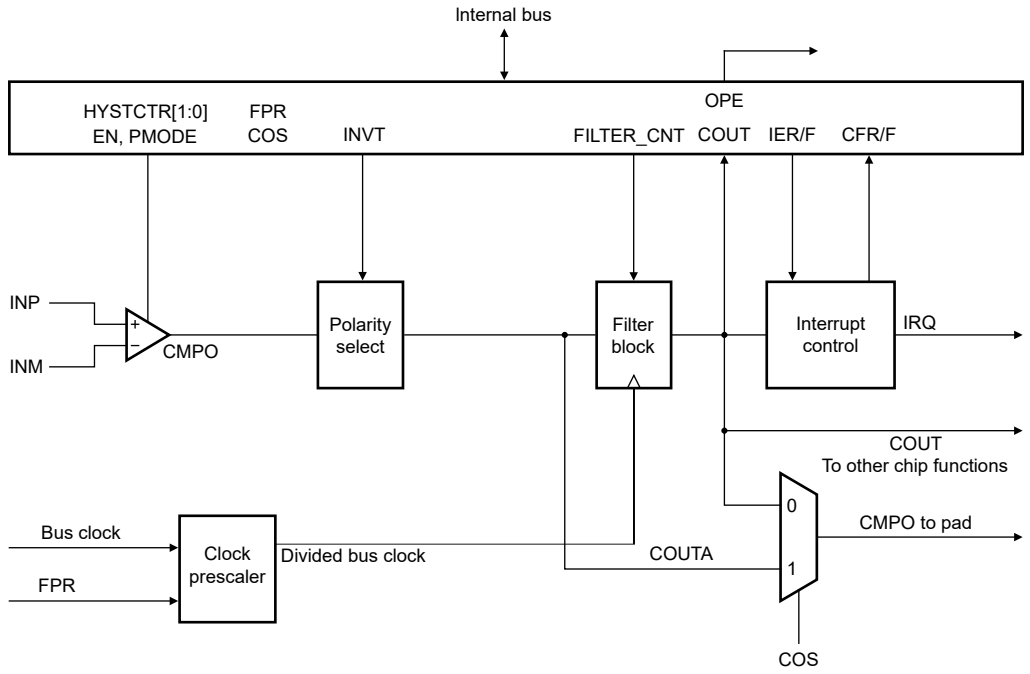


Figure 120. CMP block diagram

The filter block bypass operates as follows:

- If enabled, the filter block incurs an additional latency penalty of up to one bus clock on COUT. This is because COUT, which crosses clock domain boundaries, must resynchronize with the bus clock.

39.2.3 Features

The following subsections list the features of high-speed CMP, 8-bit DAC, and ANMUX.

39.2.3.1 CMP features

- Operates over the entire supply range
- Supports rail-to-rail inputs
- Supports programmable hysteresis control
- Provides a selectable interrupt on rising-edge, falling-edge, or both comparator outputs
- Provides a selectable inversion on comparator output
- Includes the capability to produce a wide range of outputs, such as:
  - Digitally filtered:
    - Bypass filter
- Uses external hysteresis at the same time that the output filter is used for internal functions
- Supports two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with a longer propagation delay
- Supports DMA transfer (CMP can select a comparison event to trigger a DMA transfer)
- Functional in all MCU power modes

CMP's filter functions are unavailable in Stop modes.

Other peripherals trigger the comparator.

39.2.3.2 8-bit DAC key features

- 8-bit resolution
- Selectable supply reference source
- Low-Power mode or High-Speed mode that you can configure
- Power-Down mode to conserve power when not in use
- Routing of the output of the 8-bit DAC to the internal comparator input, where you can select it as one of the CMP inputs as a reference for comparison

39.2.3.3 ANMUX key features

- Two 8-to-1 channel MUXes that can be selected as the source for comparator input
- Operation over the entire supply range

39.3 CMP functional description

CMP can compare two analog input voltages applied to INP and INM. CMPO is high when the non inverting input is greater than the inverting input, and it is low when the non inverting input is less than the inverting input. Write 1 to C0[INVT] to selectively invert this signal.

C0[IER] and C0[IEF] select the condition that causes CMP to assert an interrupt to the processor. C0[CFF] sets on a falling edge, and C0[CFR] sets on a rising edge of the comparator output. You can read the optionally filtered CMPO directly through C0[COUT].

39.3.1 CMP functional modes

CMP includes the following main sub-blocks:

- Comparator

For cases where a comparator drives a fault input, for example, for a motor-control module such as FTM or FlexPWM, it must configure to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

39.3.1.1 Disabled mode

In this mode:

- The analog comparator is nonfunctional and consumes no power.
- CMPO is 0.

39.3.1.2 Continuous mode

In this mode, the analog comparator block is powered and active. CMPO may be optionally inverted but is not subject to filtering. CMP completely bypasses the filter block and update C0[COUT] continuously. The path from comparator input pins to output pins operates in a combinational unclocked mode. COUT and COUTA are identical.

See CMP block diagram section for control configurations that result in disabling the filter block.

Figure 121 describes the comparator operation in Continuous mode.

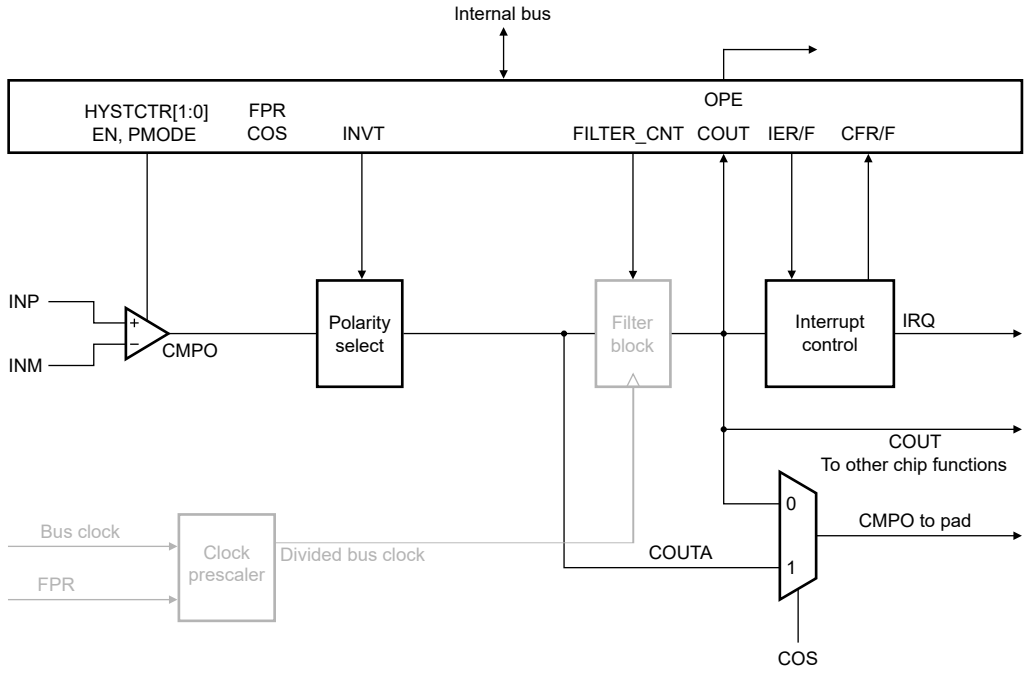


Figure 121. Comparator operation in Continuous mode

39.3.2 Low-pass filter

The low-pass filter operates on the unfiltered, unsynchronized, and optionally inverted comparator output, COUTA, and generates the filtered and synchronized output, COUT. You can configure both COUTA and COUT as module outputs, and use them for different purposes within the system.

Synchronization and edge detection always determine the status register bit values. .

The need for digital filtering and the amount of filtering depends on your requirements. Filtering can become more useful in the absence of an external hysteresis circuit, without which you can generate high-frequency oscillations at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

39.3.3 CMP clocks

This module contains the following clocks:

- Bus clock (ipg_clk/ipg_clk_s) for access to registers and filter operation

39.3.4 Interrupts

CMP generates an interrupt on either the rising-edge or falling-edge of the comparator output, or both. Assuming that C0[DMAEN] is not 1, the following table presents the conditions in which the interrupt request is asserted and deasserted.

Table 305. CMP interrupt generations

Condition	Result
C0[IER] and C0[CFR] are 1.	The interrupt request is asserted.
C0[IEF] and C0[CFF] are 1.	The interrupt request is asserted.
C0[IER] and C0[CFR] are 0 for a rising-edge interrupt.	The interrupt request is deasserted.
C0[IEF] and C0[CFF] are 0 for a falling-edge interrupt.	The interrupt request is deasserted.

39.3.5 DMA support

CMP generates a CPU interrupt if there is a change on COUT. The corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt after writing 1 to C0[DMAEN] to enable the DMA support, and writing 1 to C0[IER], C0[IEF], or both to enable the interrupt. After the DMA completes transfer, it sends a transfer-completing indicator signal that deasserts the DMA transfer request and clears the flags (C0[CFR] and C0[CFF]) to allow a subsequent change on the comparator output to occur and force another DMA request.

39.4 DAC functional description

39.4.1 DAC block diagram

The following figure shows a 256-tap resistor ladder network and a 256-to-1 multiplexer. They select an output voltage from one of the 256 distinct levels that outputs from DACO.

CMP Control 1 (C1) controls DAC. You can select its supply reference source from two sources, V_{in1} and V_{in2}. Power down or disable the module when not in use. In Disabled mode, DACO is connected to the analog ground.

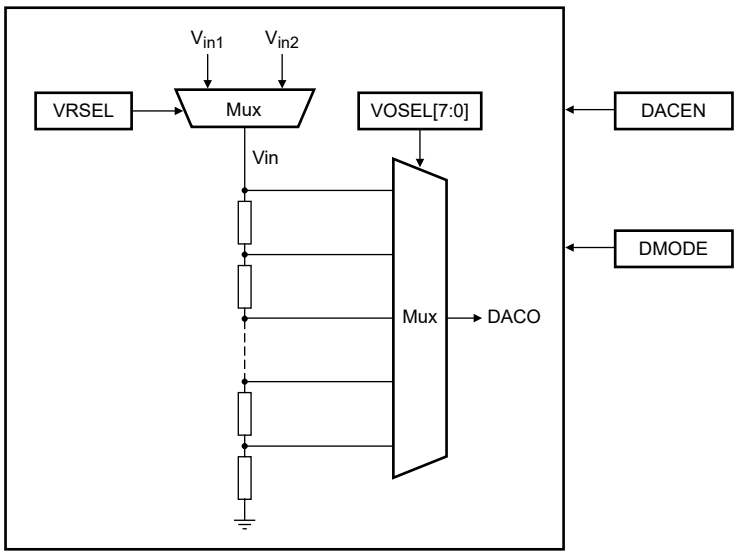


Figure 122. 8-bit DAC block diagram

39.4.2 Voltage reference source select

You must use:

- V_{in1} to connect to the primary voltage source as a supply reference of the 256-tap resistor ladder.
- V_{in2} to connect to an alternate voltage source, or primary source, if an alternate voltage source is not available.

39.4.3 DAC reset

This module contains a single reset input corresponding to the chip-wide peripheral reset.

39.4.4 DAC clocks

This module contains a single clock input: bus clock (ipg_clk/ipg_clk_s) for access to registers and filter operation.

39.4.5 DAC interrupts

This module has no interrupts.

39.5 CMP external signals

C1[PSEL] and C1[MSEL] mux the external 1.8 V domain inputs, INA_1P8 [4:1], before going to the positive port (INP) and negative port (INM) of the comparator, respectively.

Each pin can accept an input voltage that varies across the full operating range of the MCU. If the module is not enabled, you can use each pin as a digital input or output. See the specific MCU documentation for more information on what functions are shared with these analog inputs.

You can select either filtered or unfiltered comparator outputs for use on an external I/O pad.

Table 306. CMP external signals

Signal	Description	I/O
INA_1P8 [4:1]	Analog 1.8 V input channels [4:1]	I

39.6 Initialization

The time required to stabilize COUT is the power on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator and filter. Low-pass filter specifies the filter delay.

During operation, you must always consider the propagation delay of the selected data paths. It may take many bus clock cycles for COUT and C0[CFR] or C0[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.

When programmed for filtering modes, COUT initially equals 0 until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

39.7 ACMP register descriptions

39.7.1 ACMP memory map

ACMP0 base address: 4020_B000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0100_0000h
4h	Parameter (PARAM)	32	R	0000_0000h
8h	CMP Control 0 (C0)	32	RW	0000_0000h
Ch	CMP Control 1 (C1)	32	RW	0000_0000h
14h	CMP Control 3 (C3)	32	RW	1100_0000h

39.7.2 Version ID (VERID)

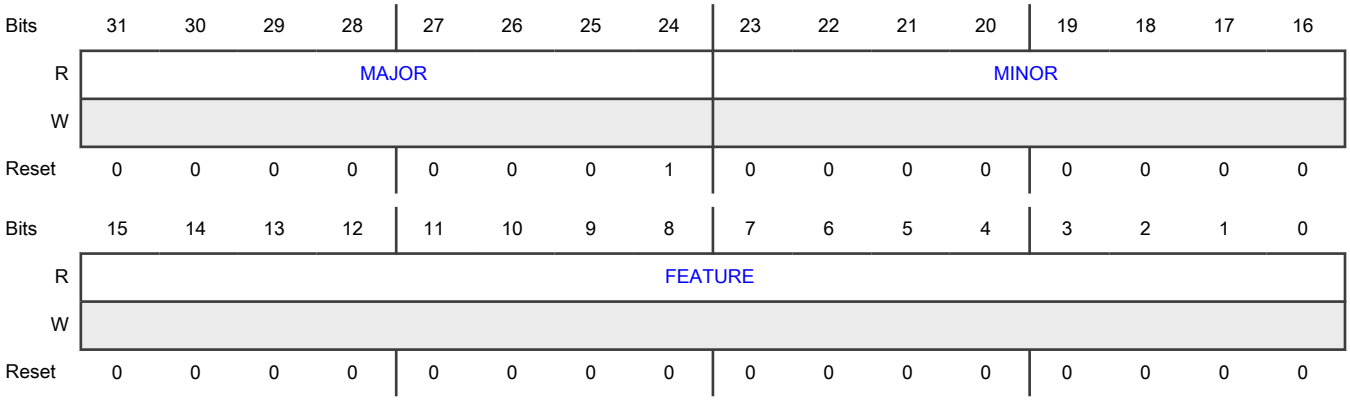
Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the chip.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number.

39.7.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the feature parameters for this instance on the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PARAM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PARAM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PARAM	Parameters Indicates the feature parameters implemented along with Version ID (VERID).

39.7.4 CMP Control 0 (C0)

Offset

Register	Offset
C0	8h

Function

Contains configuration options for the CMP function, such as DMA, interrupt control, and filter function. It also includes the hysteresis control for the analog CMP.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LINKEN	DMAEN	0	IER	IEF	CFR	CFF	COUT	FPR							
W						W1C	W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	PMODE	INVT	COS	OPE	EN	0	FILTER_CNT			0		HYSTCTR	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LINKEN	CMP to DAC Link Enable Enables the link from CMP to DAC. If this field is 1, C0[EN], instead of C1[DACEN], controls the DAC enable and disable function. 0b - Disable 1b - Enable
30 DMAEN	DMA Enable Enables the DMA transfer that CMP triggers. If this field and the corresponding interrupt enable field are 1, a DMA request is asserted when C0[CFR] or C0[CFF] is set. 0b - Disable 1b - Enable
29 —	Reserved
28 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from CMP. If this field is 1, an interrupt asserts when C0[CFR] is set. 0b - Disable 1b - Enable
27 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from CMP. If this field is 1, an interrupt asserts when C0[CFF] is set. 0b - Disable 1b - Enable
26 CFR	Analog Comparator Flag Rising Detects a rising edge on COUT, when set, during normal operation. In stop mode, this field is level sensitive. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
25	Analog Comparator Flag Falling

Table continues on the next page...

Table continued from the previous page...

Field	Function
CFF	Detects a falling edge on COUT, when set, during normal operation. In Stop mode, this field is level sensitive. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
24 COUT	Analog Comparator Output Returns the current value of the analog comparator output, when read. The field resets to 0 and reads as C0[INVT] when ACMP is disabled, that is, when C0[EN] = 0. Writes to this field are ignored.
23-16 FPR	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter. Writing 0h to this field disables the filter. See CMP functional description for more information on filter programming and latency.
15-14 —	Reserved
13 —	Reserved
12 PMODE	Power Mode Select Selects the power mode. 0b - Low-speed (LS) 1b - High-speed (HS)
11 INVT	Comparator Invert Selects the polarity of the analog comparator function. It is also driven to the COUT output (on the chip pin and as C0[COUT]) when C0[OPE] = 0. If this field is 1, the comparator output is inverted. If this field is 0, the comparator output is not inverted. 0b - Do not invert 1b - Invert
10	Comparator Output Select

Table continues on the next page...

Table continued from the previous page...

Field	Function
COS	Specifies whether CMPO must be set to equal COUT (filtered comparator output) or COUTA (unfiltered comparator output). 0b - COUT 1b - COUTA
9 OPE	Comparator Output Pin Enable Enables the path from the comparator output to a selected pin. If the value of this field is 0, the comparator output (after filter settings depending on software configuration) is not available to a packaged pin. If the value of this field is 1 and the software configures the comparator to own a packaged pin, the comparator is available in a packaged pin. 0b - Disable 1b - Enable
8 EN	Analog Comparator Module Enable Enables ACMP. When the module is not enabled, the analog part remains in the off state, and consumes no power. 0b - Disable 1b - Enable
7 —	Reserved
6-4 FILTER_CNT	Filter Sample Count Specifies the number of consecutive samples that must agree prior to the comparator output filter accepting a new output state. 000b - Filter is disabled 001b - One consecutive sample 010b - Two consecutive samples 011b - Three consecutive samples 100b - Four consecutive samples 101b - Five consecutive samples 110b - Six consecutive samples 111b - Seven consecutive samples
3-2 —	Reserved
1-0	Comparator Hard Block Hysteresis Control

Table continues on the next page...

Table continued from the previous page...

Field	Function
HYSTCTR	Selects the level of internally generated hysteresis for the hard block output. See chip data sheet to get the actual hystersis value with each level. 00b - Level 0 01b - Level 1 10b - Level 2 11b - Level 3

39.7.5 CMP Control 1 (C1)

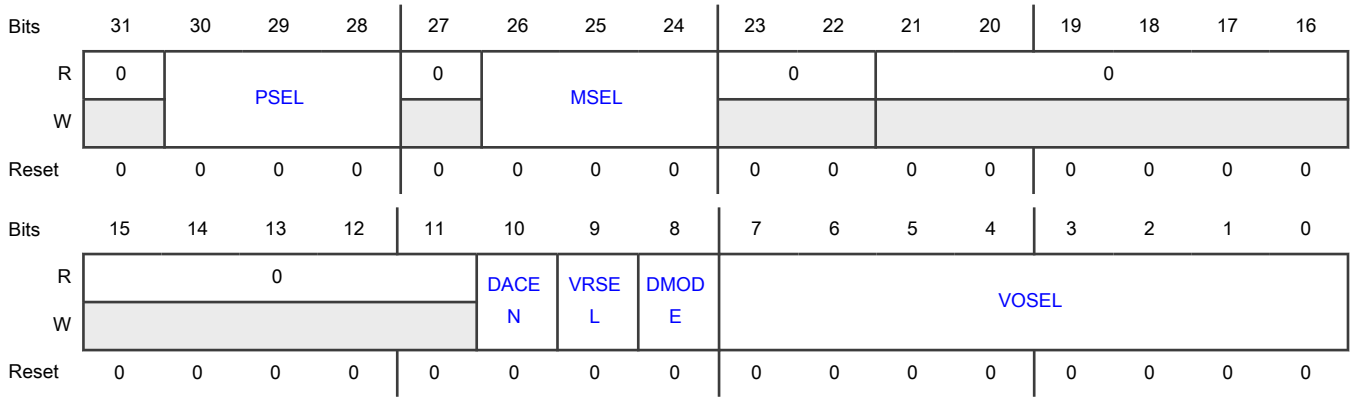
Offset

Register	Offset
C1	Ch

Function

Contains configuration options for CMP channel selection as well as for analog DAC control.

Diagram



Fields

Field	Function
31 —	Reserved
30-28 PSEL	Plus Input MUX Control Determines which input is selected for the positive mux.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• For more information about the external inputs, see CMP external signals and chip-specific ACMP information. • For connections about the internal positive input 0, see chip-specific ACMP information. The 8-bit DACoutput connects internally. **NOTE** In Round-Robin mode of operation, you must exclude internal positive input 0 from the active and fixed channel ports. C1[MSEL] and this field must have different values. 000b - Internal positive input 0 for plus channel (internal plus input) 001b - External input 1 for plus channel (INA_1P8[1]) 010b - External input 2 for plus channel (INA_1P8[2]) 011b - External input 3 for plus channel (INA_1P8[3]) 100b - External input 4 for plus channel (INA_1P8[4]) 101b - Reserved 110b - Reserved 111b - Internal 8-bit DAC output
27 —	Reserved
26-24 MSEL	Minus Input MUX Control Determines which input is selected for the negative mux. • For more information about the external inputs, see CMP external signals and chip-specific ACMP information. • For connections about the internal negative input 0, see chip-specific ACMP information. The 8-bit DAC output connects internally. **NOTE** In Round-Robin mode of operation, you must exclude internal negative input 0 from the active and fixed channel ports. This field and C1[PSEL] must have different values. 000b - Internal negative input 0 for minus channel (internal minus input) 001b - External input 1 for minus channel (INA_1P8[1]) 010b - External input 2 for minus channel (INA_1P8[2]) 011b - External input 3 for minus channel (INA_1P8[3]) 100b - External input 4 for minus channel (INA_1P8[4]) 101b - Reserved 110b - Reserved 111b - Internal 8-bit DAC output

Table continues on the next page...

Table continued from the previous page...

Field	Function
23-22 —	Reserved
21-16 —	Reserved
15-11 —	Reserved
10 DACEN	DAC Enable Enables the DAC. After the DAC is disabled, it is powered down to conserve power. 0b - Disable 1b - Enable
9 VRSEL	Supply Voltage Reference Source Select Selects the source of resistor ladder network supply reference (Vin). See chip-specific ACMP information for the actual DAC voltage reference sources of Vin1 and Vin2. 0b - Vin1 1b - Vin2
8 DMODE	DAC Mode Select Selects a DAC mode. 0b - Low-Speed and Low-Power mode 1b - High-Speed and High-Power mode
7-0 VOSEL	DAC Output Voltage Select Selects an output voltage from one of 256 distinct levels. DACO = (Vin/256) × (VOSEL[7:0] + 1), so the DACO range is from Vin/256 to Vin.

39.7.6 CMP Control 3 (C3)

Offset

Register	Offset
C3	14h

Function

Contains configuration options for MUX gating only when you remove the 3 V domain.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			PCHCTEN	0			NCHCTEN	0			0	0			0
W																
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0			0	0			0	0			0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28 PCHCTEN	Positive Channel Enable Enables the plus channel. 0b - Disable 1b - Enable
27-25 —	Reserved
24 NCHCTEN	Negative Channel Enable Enables the minus channel. 0b - Disable 1b - Enable
23-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 —	Reserved
15	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
14-12 —	Reserved
11 —	Reserved
10-8 —	Reserved
7 —	Reserved
6-4 —	Reserved
3-0 —	Reserved

Chapter 40
Ultra Secured Digital Host Controller (uSDHC)

40.1 Chip-specific uSDHC information

Table 307. Reference links to related information

Topic	Related module	Reference
Full description	uSDHC	uSDHC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

40.1.1 Module instances

This device has two instances of the uSDHC module, uSDHC0 and uSDHC1.

Table 308. uSDHC instances

Features	uSDHC0	uSDHC1
Located in	VDDN_MEDIA	VDDN_MEDIA
Access (LP)	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
Support eMMC5.0/5.1 standard	Yes	No

40.1.2 External signals

The following table lists the mapping of signal names used in this chapter and Pinout chapters.

Table 309. External signals mapping

External signals in this chapter	Pin signals
CLK	SDHCy_CLK
CMD	SDHCy_CMD
DATx	SDHCy_DATAx
CD_B	SDHCy_CARD_DET_N
WP	SDHCy_WR_PRT
RESET_B	SDHCy_RESET_N

Table continues on the next page...

Table 309. External signals mapping (continued)

External signals in this chapter	Pin signals
VSELECT	-
STROBE	SDHCy_DS

40.2 Overview

uSDHC provides the interface between the host system and the eMMC, SD card, and SDIO as shown in Figure 124. The module acts as a bridge, passing host bus transactions to the eMMC, SD card, and SDIO by sending commands and performing data accesses to/from the cards. It handles the SD card/SDIO/eMMC protocols at the transmission level.

Table 310. Updated terms

Updated term	Deprecated term
Target	Slave
uSDHC controller	Master (related to device)
Initiator	Master (related to AXI bus)
Placeholder	Dummy

40.2.1 Block diagram

The following figure illustrates the block diagram for uSDHC. Dual port SRAM is FIFO for write/read. For more information, see Data buffer.

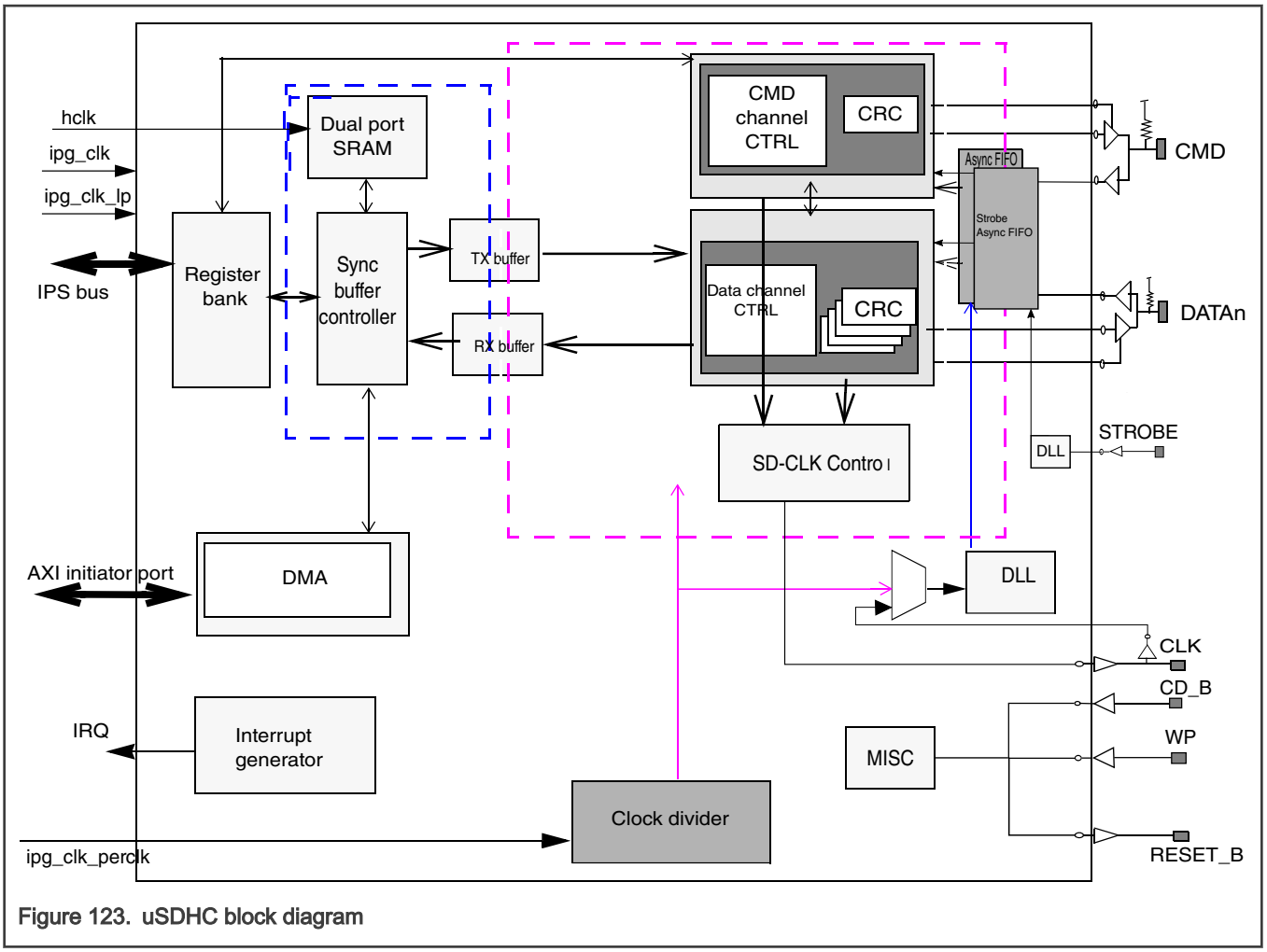


Figure 123. uSDHC block diagram

The figure below shows the System connection of uSDHC:

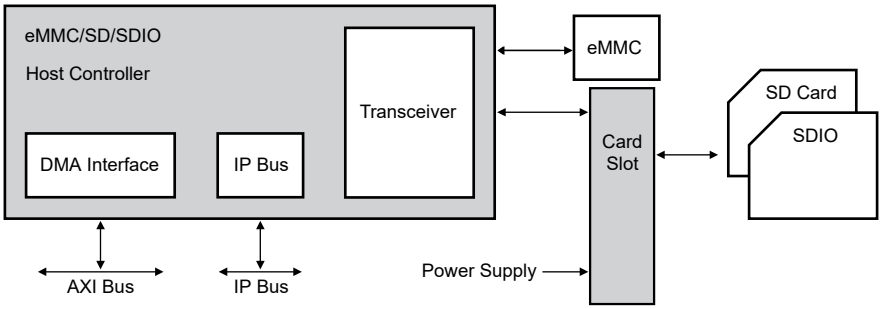


Figure 124. System connection of uSDHC

The following are brief descriptions of the cards supported by uSDHC:

- The Embedded MultiMediaCard (eMMC) is a universal low-cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous eMMC were based on a 7-pin serial bus with a single data pin, while the new high speed eMMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.
- The Secure Digital Card (SD) is an evolution of the old eMMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic

devices. The physical form factor, pin assignment, and data transfer protocol are backward compatible with the old eMMC (with some additions).

- Under the SD protocol, system connection can be categorized into memory card, I/O card, and combo card, which have both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI (Secure Digital Music Initiative) standard. The I/O card, which is also known as SDIO, provides high-speed data I/O with low-power consumption for mobile electronic devices.

40.2.2 Features

The list given below shows the features of the uSDHC module:

- Conforms to the SD Host Controller Standard Specification version 2.0/3.0
- Compatible with the eMMC System Specification version 4.2/4.3/4.4/4.41/4.5/5.0/5.1
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Specification version 2.0/3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, eMMC, MMC plus, and MMC RS cards
- For card bus clock frequency: See the product Data Sheet for the clock frequency of each supported mode.
- Supports 1-bit/4-bit SD and SDIO modes, and 1-bit/4-bit/8-bit eMMC modes
  - Up to 832 Mbps of data transfer for SDIO using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDIO using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 832 Mbps of data transfer for SDXC cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDXC card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 1600 Mbps of data transfer for eMMC using eight parallel data lines in the Single Data Rate (SDR) mode
  - Up to 3200 Mbps of data transfer for eMMC using eight parallel data lines in the Dual Data Rate (DDR) mode
- Supports single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer for memory device
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes; also supports interrupt period
- Supports internal DMA capabilities
- Supports voltage selection by configuring vendor-specific register bit VEND_SPEC[VSELECT]
- Supports Advanced DMA to perform linked memory access
- Supports Command Queue mechanism

NOTE

This block can support all the above-listed speed mode and maximum data throughput. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

40.3 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA AXI interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

40.3.1 Modes and operations

40.3.1.1 Data transfer modes

The uSDHC module can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- eMMC 1-bit
- eMMC 4-bit
- eMMC 8-bit
- Identification mode (up to 400 kHz)
- eMMC full-speed mode (up to 26 MHz)
- eMMC high-speed mode (up to 52 MHz)
- eMMC HS200 mode (up to 200 MHz)
- eMMC HS400 mode (200 MHz both edges)
- eMMC DDR mode (52 MHz both edges)
- SD card or SDIO full-speed mode (up to 25 MHz)
- SD card or SDIO high-speed mode (up to 50 MHz)
- SD card or SDIO UHS-I mode (up to 208 or 100 MHz in SDR mode, up to 50 MHz in the DDR mode)

NOTE

This block can support all the above listed speed mode and maximum clock frequency. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

40.3.2 Data buffer

The uSDHC module uses one configurable data buffer to transfer data between the system bus (IP bus or advanced extensible interface (AXI) bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the AXI initiator clock). The buffer is used as a temporary storage for transferring data between the host system and the card. The watermark levels for read and write are both configurable and can range between 1 to 128 words. When using internal DMA mode, The watermark level for read should be configured to 16 and watermark level for write should be configured to integral multiple of 16 to get max AXI bus throughput. The next figure provides the uSDHC buffer scheme as buffer control and buffer RAM wrapper.

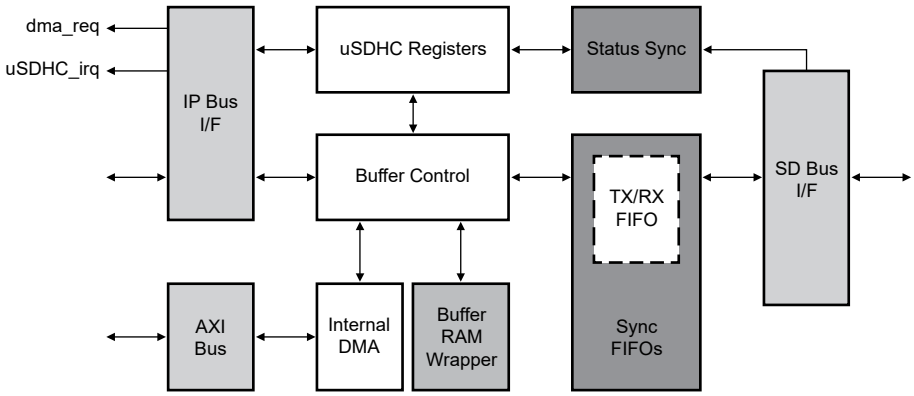


Figure 125. uSDHC buffer scheme

Here are 2 transfer modes to access the data buffer:

- CPU polling mode:
  - For a host-read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the host driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar. For more information on the process of writing operation, see Write operation sequence.
- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AXI bus.

For a read operation, when there are more words in the buffer than the amount set in WTMK_LVL[RD_WML], the internal DMA starts fetching data over the AXI bus.

The Write operation functions in a similar manner—sequential and contiguous access is necessary to ensure that the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, the byte order is swapped after the data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, the byte order is swapped before the data is stored in the buffer.

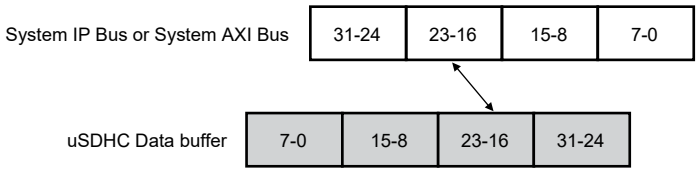


Figure 126. Data swap between system bus and uSDHC data buffer in the byte little endian mode

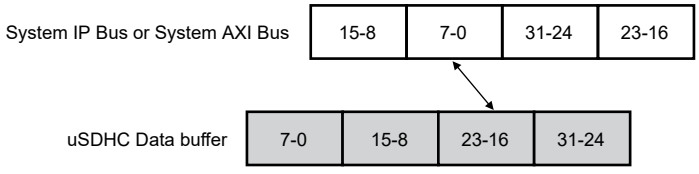


Figure 127. Data swap between system bus and uSDHC data buffer in half word big endian mode

40.3.2.1 Write operation sequence

There are 2 ways to write data into the buffer when the user transfers data to the card:

- Processor core polling through the BWR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, the DMAEN bit in the Transfer Type register is not set when the command is sent, uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in WTMK_LVL[WR_WML] and is ready for receiving new data. At the same time, uSDHC sets the BWR bit. The buffer write ready interrupt is generated if it is enabled by software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC stops the data transfer and abandons the current block. The host driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the write operation from that address. It is recommended that a software reset for Data be applied before the transfer is restarted.

The uSDHC module does not start data transmission until the buffer has been filled with the number of words set in the WR_WML register. If the buffer is empty and the host system does not write data in time, uSDHC stops the CLK to avoid the data buffer underrun situation.

40.3.2.2 Read operation sequence

There are 2 ways to read data from the buffer when the data is received from the card:

- Processor core polling through the BRR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, which is available and ready for system fetching data. At the same time, uSDHC sets the BRR bit. The buffer read ready interrupt is generated if it is enabled by the software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC stops the data transfer and abandons the current block. The host driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the read operation from that address. It is recommended that a software reset (in register RSTD) for data be applied before the transfer is restarted.

For any read transfer mode, uSDHC does not start data transmission until the number of words set in the RD_WML register are in buffer. If the buffer is full and the host system does not read data in time, uSDHC stops the CLK to avoid the data buffer overrun situation.

40.3.2.3 Data buffer and block size

In the uSDHC module, the data buffer can hold up to 256 words (32-bit) and the watermark levels for write and read can be configured accordingly. The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way. The host driver may configure the value according to the system and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes). That means, the largest block size is 512 bytes.

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned), stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the software side must write twice for each block. For each block, the ending byte is abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

40.3.2.4 Dividing large data transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in multiples of block size. If the total data length cannot be divided evenly into a multiple of the block size, then there are two options to transfer the data. These two options depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add placeholder data in the last block to fill the block size. For option 2, the card must manage the removal of the placeholder data. Only for write, uSDHC sends data to the card.

See Figure 128 for an example showing the division of large data transfers, assuming a kind of WLAN SDIO that only supports a block size of up to 64 bytes. Although uSDHC supports a block size of up to 4096 bytes, SDIO can only accept a block size of less than 64 bytes, so the data must be divided (see the example below).

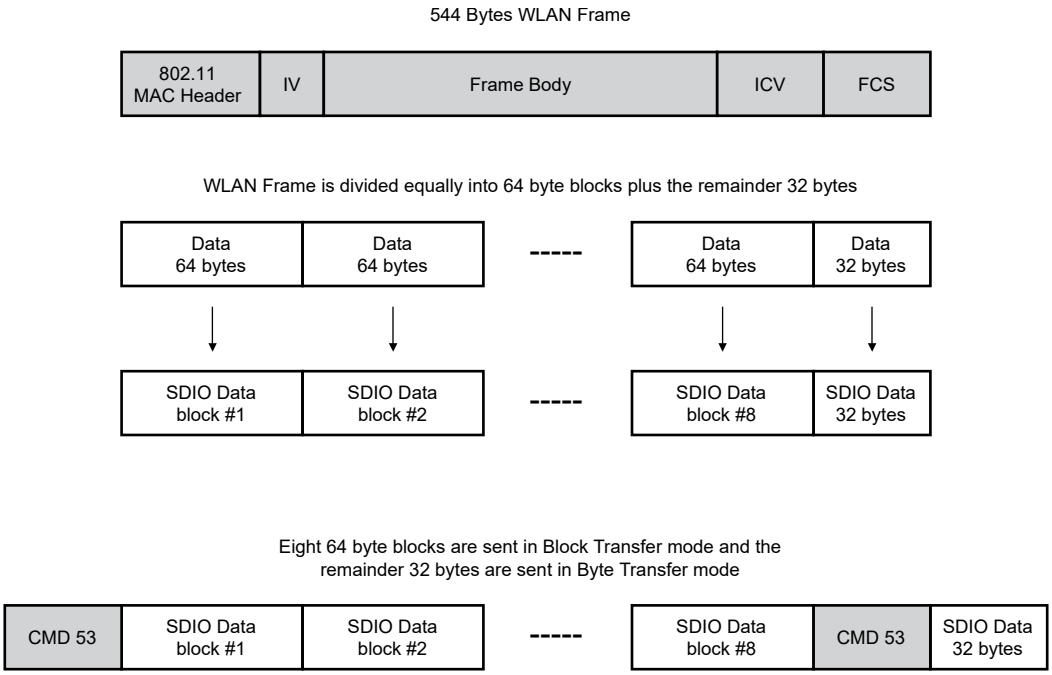


Figure 128. Example for dividing large data transfers

40.3.3 DMA AXI interface

The internal DMA implements a DMA engine and the AXI initiator. When the internal DMA is enabled, but the BWR and BRR bits are set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register. See Figure 129 for an illustration of the DMA AXI interface block.

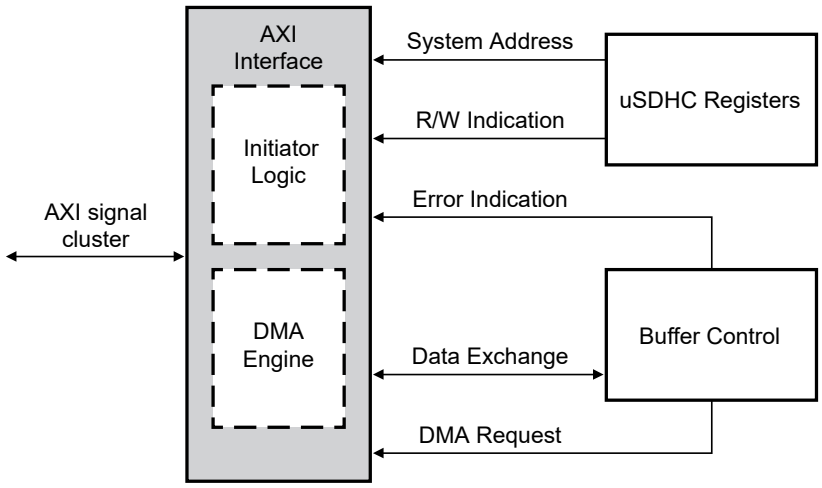


Figure 129. DMA AXI interface block

40.3.3.1 Internal DMA request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the data buffer block sends a DMA request to AXI interface.

The delay in response from the internal DMA engine depends on the system AXI bus loading and the priority assigned to uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request if the data buffer space (for write) or bytes in data buffer is smaller than the watermark level. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of the current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always automatically set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the watermark level set to six words (24 bytes). After the first burst transfer, if there are greater than or equal to two words in the buffer, which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to transfer for the current block is ceiling((31 - 6 * 4) / 4) = 2. The uSDHC module reads two words out of the buffer, with seven valid bytes and one stuffed byte.

40.3.3.2 AXI initiator interface

It is possible that the internal AXI DMA engine could fail during the data transfer. Upon detection of an AXI bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state. At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register is generated to host CPU to report a bus error condition.

After the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and restart the transfer from this address to recover the corrupted block.

40.3.3.3 ADMA engine

In the SD host controller standard, a new DMA transfer algorithm called the Advanced DMA (ADMA) is defined. For simple DMA, after the page boundary is reached, a DMA interrupt is generated and the new system address is programmed by the host driver. The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention is not be needed during long DMA-based data transfers.

There are two types of ADMA in host controller: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory, and ADMA2 eliminates the restriction so that the data of any location (word aligned) and any size (less than 64 KB) can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors defined in SD host controller Standard, and if the "End" flag is detected in the descriptor, ADMA stops after this descriptor is processed.

40.3.3.3.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Rsv descriptor (new in ADMA2)
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA starts read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before the tran type descriptor. Every tran type descriptor triggers a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, the tran type descriptor contains both data length and transfer data address, so the tran type descriptor itself can start a data transfer.

See Figure 130 for the format of the descriptor table for ADMA1.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 130. Format of the ADMA1 descriptor table

See Figure 131 for the concept and access method of the descriptor table for ADMA1.

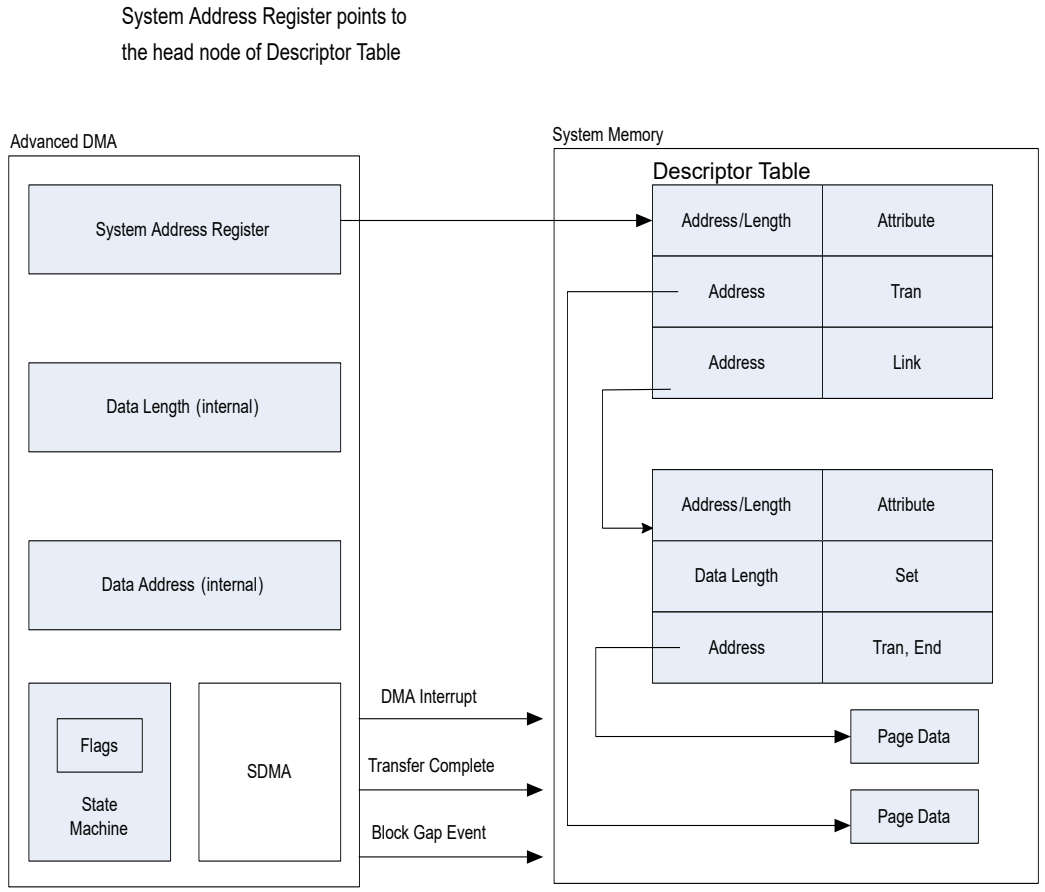


Figure 131. Concept and access method of the ADMA1 descriptor table

The figure below explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it ignores the higher 32-bit. The Address field should be set to word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 132. Format of the ADMA2 descriptor table

See Figure 133 for the concept and access method of the descriptor table for ADMA2.

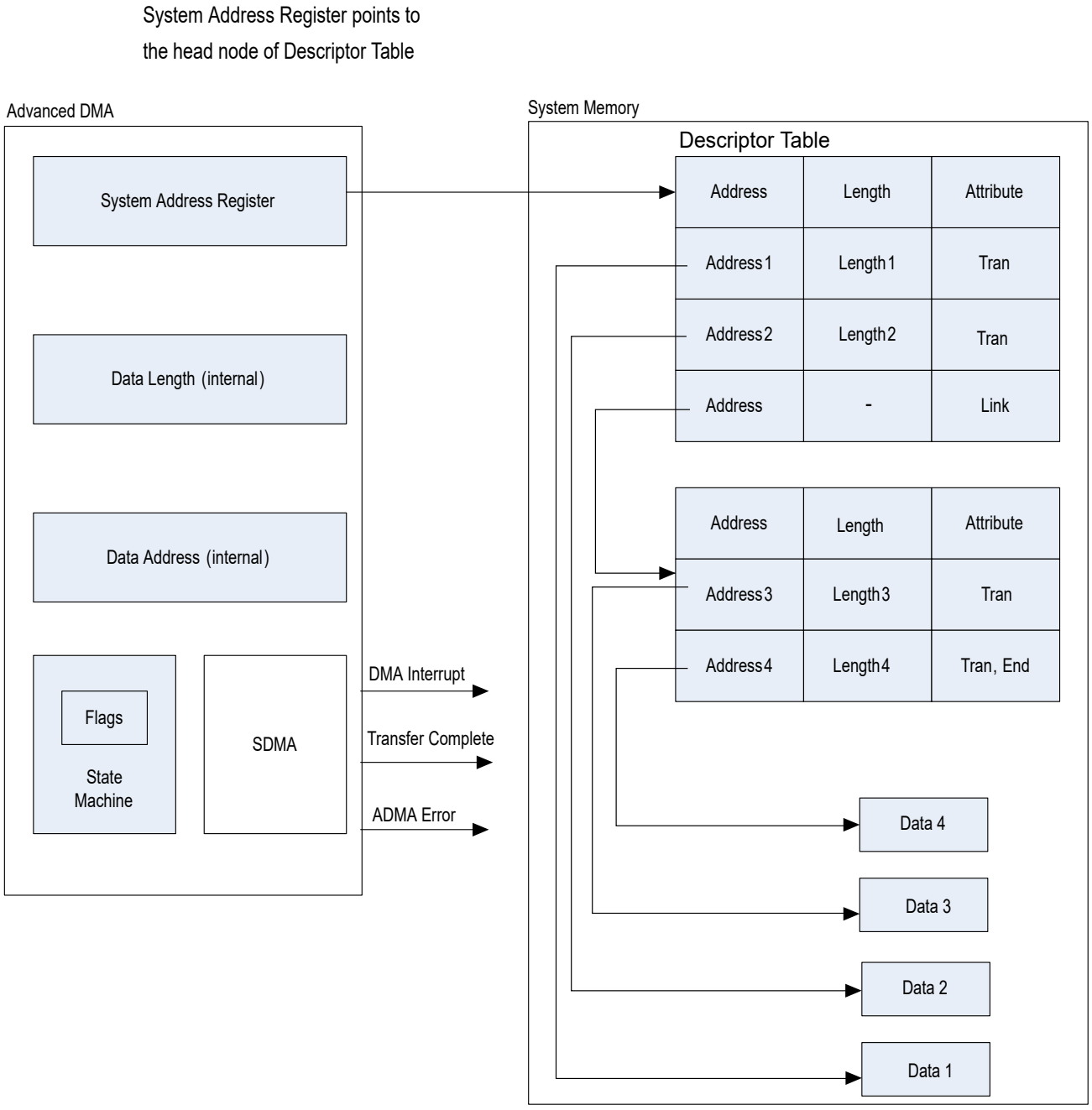


Figure 133. Concept and access method of the ADMA2 descriptor table

40.3.3.3.2 ADMA interrupt

If the interrupt flag descriptor is set, ADMA generates an interrupt according to various types of descriptors.

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer descriptor is complete, which means the data communication between the host system and uSDHC is finished.
- Link type of descriptor: interrupt is generated when new descriptor address is set.

- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer descriptor is complete, which means the data communication between the host system and uSDHC is finished.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

40.3.3.3.3 ADMA error

The ADMA stops whenever an error is encountered. These errors include:

- Fetching descriptor error
- AXI response error
- Data length mismatch error

An ADMA descriptor error is generated when it fails to detect a "valid" flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the "Interrupt" flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in register BLK_ATT must be equal to the whole data length set in descriptor, otherwise data length mismatch error is generated.

When BLKCNTEN bit is not set, then the whole data length set in descriptor is a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors occur.

40.3.4 Register bank with IP bus interface

Register accesses through the IP bus interface are on the register bank. See Figure 134 for the block diagram.

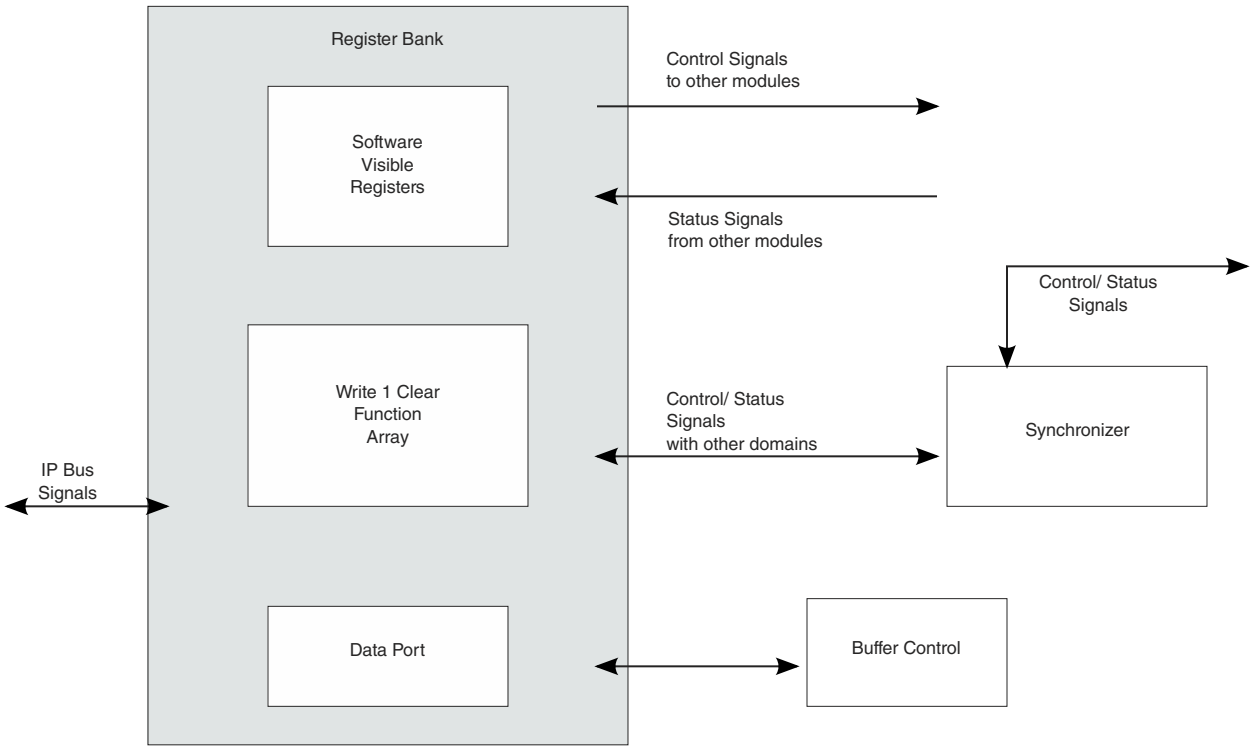


Figure 134. Register bank diagram

Only a 32-bit access is allowed, and no partial read/write is supported; therefore, all accesses are word aligned.

Access to an unimplemented address within the register memory map space does not generate a transfer error.

40.3.4.1 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DAT lines
- Monitors the interrupt from SDIO
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four sub-modules:

- SD control misc
- Command control
- Data control
- Clock control

40.3.4.2 SD control miscellaneous

In the SD control miscellaneous unit:

- The card detection (including CD_B and DATA3 for card detection) and card interrupt are implemented.
- The module monitors the signal level on all the eight data lines and the command lines. It directly routes the level values into the register bank.
- The module also detects the Write Protect (WP) line. If WP is activated as '1', writes to memory and combo cards are ignored by cards. In that case, there is no need to send write command anymore.

40.3.4.3 SD clock control

If the internal data buffer is near full (for read) or near empty (for write), the SD clock must be gated off to avoid buffer over/under-run. This module asserts the gate of the output SD clock to shut the clock off. After the buffer has space (for read) or has data (for write), the clock gate of this module opens, and the SD clock is active again.

40.3.4.4 Command control

The Command Control module deals with the transactions on the CMD line.

See Figure 135 for an illustration of the structure for the Command CRC shift register.

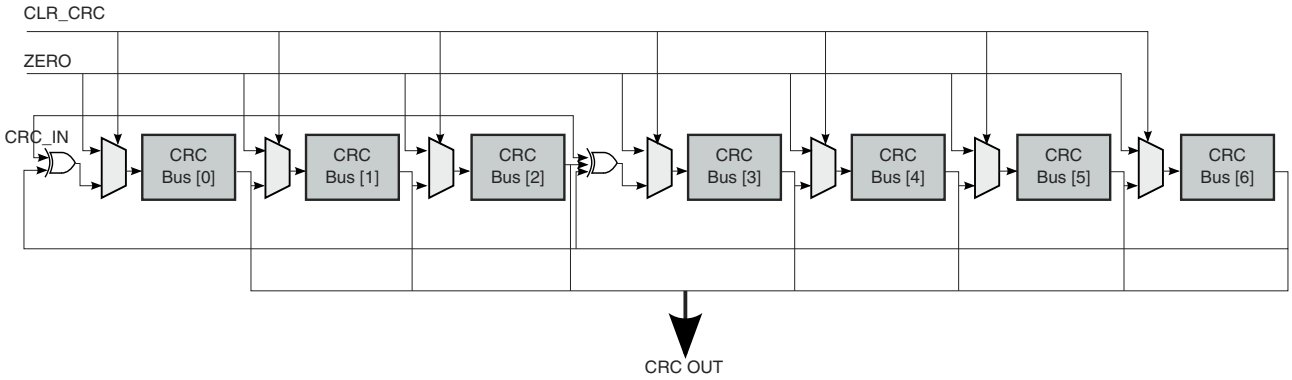


Figure 135. Command CRC shift register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

40.3.4.5 Data control

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line and generates the read wait state by the request from the transceiver.

The CRC polynomials for the data are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1

M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0

CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

40.3.5 Card insertion and removal detection

The uSDHC module uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. It is controlled by SoC pad or other logic. When there is no card on the eMMC/SD bus, the DATA3 is pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, uSDHC sends an interrupt (if enabled) to inform the Host system that a card is inserted.

40.3.6 Power management and wakeup events

When there is no operation between uSDHC and the card through the SD bus, the user can completely disable the peripheral clock and base clock in the chip-level clock control module to save power. When the user needs to use uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to uSDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC module can generate these interrupts even when there are no clocks enabled. The three interrupts that can be used as wakeup events are these:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO (not available in multiple block data transfers)

These three wakeup events (or wakeup interrupts) can also be used to wakeup the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to uSDHC are disabled or when the entire system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (PROT_CTRL) for more information on the uSDHC Protocol Control register.

40.3.6.1 Setting wakeup events

For uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters the sleep mode. Before the software disables the host clock, it should ensure that all the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

40.3.7 eMMC fast boot

The Embedded MultiMediaCard (eMMC4.3 or later) specification adds a fast boot feature that requires hardware support. There are two types of fast boot modes in the eMMC4.3 or later specification: boot operation and alternative boot operation. Each type also has with-acknowledge and without-acknowledge modes.

In the boot operation mode, the uSDHC controller (eMultiMediaCard host) can read boot data from the target (eMMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for target), before issuing CMD1.

NOTE

For the eMMC4.3 setting, please see the eMMC4.3 specification.

40.3.7.1 Boot operation

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the target recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after the CMD line goes low, the target starts to send the first boot data to the uSDHC controller on the DATA line(s). The uSDHC controller must keep the CMD line low to read all of the boot data.

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If boot acknowledge is enabled, the target has to send acknowledge pattern '010' to the uSDHC controller within 50ms after the CMD line goes low. If boot acknowledge is disabled, the target does not send out acknowledge pattern '010'.

The uSDHC controller can terminate the boot mode with the CMD line high.

The boot operation is terminated when all the contents of the enabled boot data are sent to the uSDHC controller. After the boot operation is executed, the target gets ready for the CMD1 operation and the uSDHC controller needs to start a normal eMMC initialization sequence by sending CMD1.

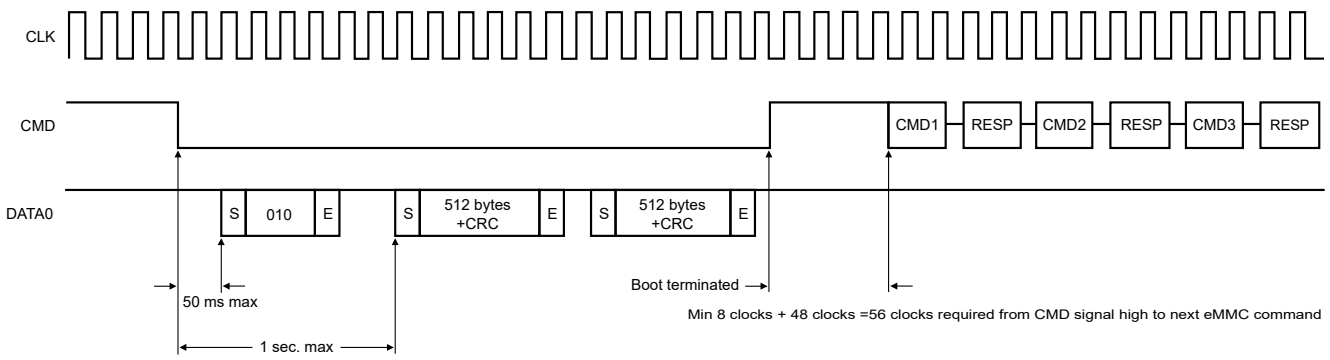


Figure 136. Embedded MultiMediaCard state diagram (normal boot mode)

40.3.7.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the target recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after CMD0 with the argument of 0xFFFFFFFA is issued, the target starts to send the first boot data to the uSDHC controller on the DATA line(s).

If boot acknowledge is enabled, the target must send the acknowledge pattern '010' to the uSDHC controller within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the target does not send out acknowledge pattern '010'.

The uSDHC controller can terminate the boot mode by issuing CMD0 (Reset).

Boot operation is terminated when all the contents of the enabled boot data are sent to the uSDHC controller. After the boot operation is executed, the target gets ready for the CMD1 operation and the uSDHC controller needs to start a normal eMMC initialization sequence by sending CMD1.

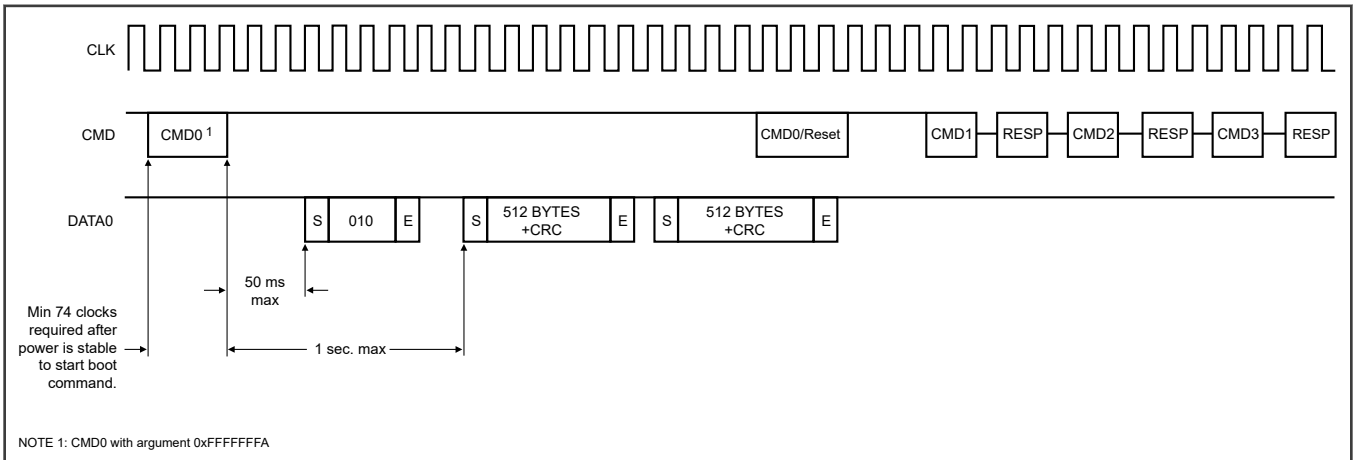


Figure 137. Embedded MultiMediaCard state diagram (alternative boot mode)

40.3.8 Commands for SD card, SDIO, and eMMC

A table containing the list of commands for the eMMC/SD card/SDIO is provided here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the SD card, SDIO, and eMMC:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DATA
- Addressed (point-to-point) data transfer commands (adtc)

Response: A response is a token that is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Table 311. Commands for eMMC/SD card/SDIO

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all eMMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all eMMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:16] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO in idle state to send them operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The Embedded MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission. Only supported by memory device.
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] reserved bits(all 0)	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	128 clocks of tuning pattern (64 byte in 4 bit mode or 128 byte in 8 bit mode) is sent for HS200 optimal sampling point detection.
CMD22	Reserved				
CMD23	ac	[31] reliable write request [30:16] set to 0 [15:0] number of blocks	R1	SET_BLOCK_COUNT	Defines the number of blocks (read/write) and the reliable writer parameter (write) for a block read or write command.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD39	ac	[31:16] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the eMMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 is used to read and write these registers. MMC command defined by CE-ATA Digital Protocol.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer. MMC command defined by CE-ATA Digital Protocol.
CMD62-63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.

Table continues on the next page...

Table 311. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for eMMC and SD cards. For eMMC, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed eMMC and high-speed SD cards. Command SWITCH_FUNC is for high-speed SD cards.
3. Command SWITCH is for high speed eMMC. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 312.
4. ACMDs is preceded with the APP_CMD command. Commands listed are used for SD only, other SD commands not listed are not supported on this module.

The access bits for the EXT_CSD access modes are listed in the following table.

Table 312. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd Set field of the argument.
01	Set bits	The bits in the pointed byte are set, according to the bits set to 1 in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the bits set to 1 in the Value field.
11	Write byte	The Value field is written into the pointed byte.

40.3.9 SDIO interrupt

Information on interrupts in 1-bit mode, interrupts in 4-bit mode, and card interrupt handling are detailed in the sections below.

40.3.9.1 Interrupts in 1-bit mode

In this case, the DATA1 pin provides the interrupt function. An interrupt is asserted by pulling the DATA1 low from SDIO, until the interrupt service is finished to clear the interrupt.

40.3.9.2 Interrupt in 4-bit mode

As the interrupt and data line 1 share pin 8 in a 4-bit mode, an interrupt is only sent by the card and recognized by the host during a specific time. This is known as the interrupt period. The uSDHC module only provides samples the level on pin 8 during the interrupt period. At all other times, the host treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in a 4-bit mode, there is only a limited period of time that the interrupt period can be active because of the limited period of data line availability between the multiple blocks of data. This requires stricter definition of the interrupt period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line holds low for one clock cycle, then pulls high for the next clock cycle . On completion of the interrupt period, the card releases the DATA1 line into the high Z state. The uSDHC module provides sample of the DATA1 during the interrupt period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Specification v1.10 for further information about the SDIO interrupt.

40.3.9.3 Card interrupt handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, uSDHC clears the interrupt request to the host system. The host driver should clear this bit before servicing the SDIO interrupt, and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from SDIO does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt source from the external card followed by writing 1 to this bit. In a 1-bit mode, uSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In a 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from SDIO and the interrupt to the Host System Interrupt Controller. When the SDIO status is set and the host driver needs to service this interrupt, the SDIO bit in the Interrupt Control Register of SDIO is cleared. This is required to clear the SDIO interrupt status latched in uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1 and uSDHC starts sampling the interrupt signal again.

See Figure 138 for an illustration of the SDIO interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

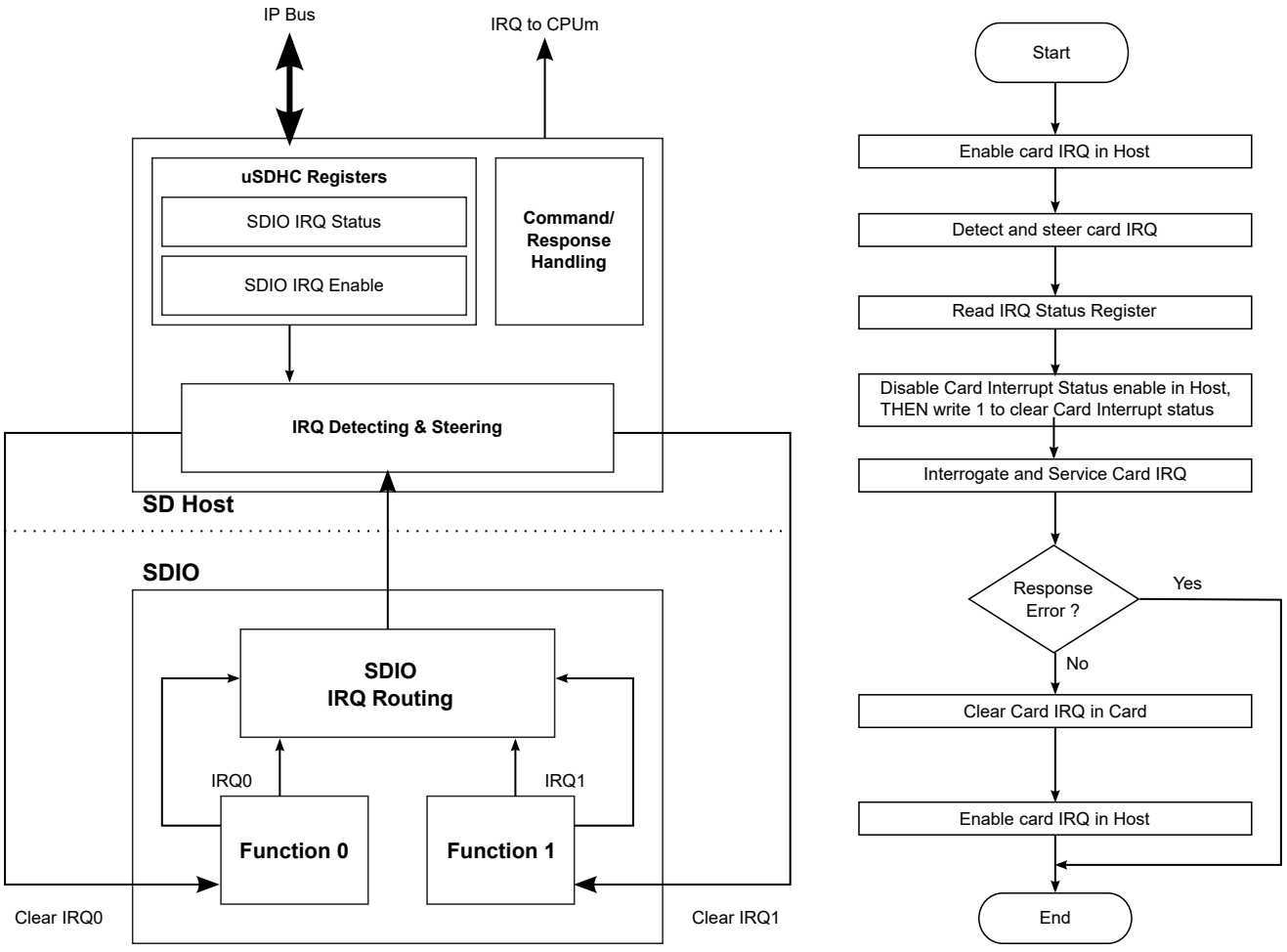


Figure 138. Card interrupt scheme, card interrupt detection, and handling procedure

40.3.10 Speed mode selection

To switch to HS200 or HS400 from High Speed (26 MHz-52 MHz), see the procedure below.

NOTE

While the actual timing change is done, the behavior of any command given (such as CMD13) cannot be guaranteed due to the asynchronous operation. Therefore, you must not use CMD13 to check the busy completion of the timing change indication. In case, CMD13 is used, the host must ignore CRC errors, if appear.

The max frequency for HS200 mode and HS400 mode could be slower than 200 MHz based on the chip implementation, you must set the target frequency for these two speed modes accordingly. Following sections describe the mode selections for HS200 mode and HS400 mode in detail.

40.3.10.1 HS200 mode selection

To switch to HS200, the Driver must perform the following steps:

1. Select the chip (through sending CMD7) and make sure it is unlocked (through CMD42).
2. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate if the chip supports HS200 at the IO voltage appropriate for both host and chip (through CMD8).
3. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported chip Driver Strengths (through CMD8). The default setting is 50 Ohm. The Host Designer might simulate its specific system, using a device driver models. Host can select the optimal Driver Type that might drive the host system load at the required operating frequency with the minimal noise generated.

   NOTE

   This step can be skipped if changes of Driver strength is not needed.

4. Check the setting of PROT_CTRL[DTW], make sure the data transfer width is set as expected.

   Make sure the I/O pad voltage is 1.8V and corresponding drive strength is set to proper value.

   Set HS200 bit and Driver Strength value in the HS_TIMING [185] field of the Extended CSD register by issuing CMD6. If the host attempts to write an invalid value, the HS_TIMING byte is not changed, the HS200 interface timing is not enabled, the Driver Strength is not changed, and the SWITCH_ERROR field is set. After the chip responds with R1, it might assert Busy signal. Once the busy signal gets de-asserted, the host may send a SEND_STATUS Command (CMD13) using the HS200 timing and after it receives a Trans State indication and No Error it means that the chip is set to HS200 timing and the Driver Strength is set to the selected settings.
5. At this point, the host can set the frequency to ≤ 200 MHz.
6. The host might invoke the HS200 standard tuning procedure, by sending CMD21 to the chip.

NOTE

The host must switch to the required bus width before starting the tuning operation to allow the tuning sequence to be done using the proper bus operating conditions.

40.3.10.2 Standard tuning procedure

By default, lower frequency operation, a fixed sampling clock is used to receive signals on CMD and DAT[3:0]. Before using the HS200, HS400, SDR104, or SDR50 modes, the Host Driver executes the tuning procedure at the mode switch sequence.

1. Issue uSDHC SW reset, set SYS_CTRL[RSTT] to 1.
2. Clear VEND_SPEC[FRC_SDCLK_ON] to 0.
3. Set TUNING_CTRL to 0x01312880 to enable the standard tuning and disable the CMD line check.
4. Start the tuning procedure by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] to 1.

5. Issue CMD19(SD) / CMD21(eMMC) with the proper Command Transfer Type (CMD_XFR_TYP) and Mixer Control (MIX_CTRL) settings.
6. Wait for uSDHC BRR (Buffer Read Ready) interrupt signal is 1.
7. Check AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]. If AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] = 1, write 1 to clear INT_STATUS[BRR] and then repeat steps 6~7. If AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] = 0, standard tuning has completed, or the tuning has not completed within 40 attempts. The Host Driver might abort this loop if the number of loops exceeds 40 or 150 ms timeout occurs. In this case, a fixed sampling clock should be used (AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0).
8. Sampling Clock Select (AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]) is valid after AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] has changed from 1 to 0.
   AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 1, indicates tuning procedure passed.
   AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0, indicates tuning procedure failed. The tuning result is applied to the delay chain, CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) [30:16], upon the successful tuning procedure completion.
9. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

While the tuning sequence is being performed, the Host Controller does not generate interrupts (including Command Complete), except Buffer Read Ready. CMD19 response errors are not indicated.

Writing AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] to 0 forces the Host Controller to use a fixed sampling clock and resets the tuning circuit of the Host Controller.

NOTE

- There could be slight difference on delay cell delay in each project, and this difference can lead to different loop number needed. TUNING_CTRL[TUNING_STEP] can be used to control how may taps are to be added for each step.
- Manual tuning might be required in cases where standard tuning is not able to close passing window with CMD19/21 transfer fail.

40.3.10.3 Manual tuning procedure

Manual tuning controls more details in the tuning procedure, and some corner cases. Manual tuning provides control to tuning start point, tuning steps, and delay chains. To start manual tuning:

1. Clear TUNING_CTRL[STD_TUNING_EN] to 0.
2. Clear VEND_SPEC[FRC_SDCLK_ON] and MIX_CTRL[FBCLK_SEL] to 0, set MIX_CTRL[SMP_CLK_SEL] and MIX_CTRL[EXE_TUNE] to 1 to start tuning.
3. Set SYS_CTRL[RSTA] to 1, and wait for self-clear.
4. Config delay chain, CLK_TUNE_CTRL_STATUS[DLY_CELL_SET_PRE] with the valueA.

NOTE

The valueA must start from 0.

5. Set SYS_CTRL[RST_FIFO] to 1 and wait for self-clear.
6. Send CMD19/21 to card.
7. Poll for CC, CIE, CEBE, CCE and CTOE of Interrupt Status (INT_STATUS) assertion.
8. Poll for TC, DEBE, DCE and DTOE of Interrupt Status (INT_STATUS) assertion.
9. Check TP of Interrupt Status (INT_STATUS); TP = 1 & no command/data error indicates the current CMD19/21 transfer is passed; otherwise, indicates failed.
10. If TP = 0 or command/data error is observed, increase the valueA with the required tuning step(1 to 127) and repeat steps 3~9.

11. If TP = 1 & no command/data error is met after a loop with TP=0 (or command/data error), save the current valueA as value_start. Increase valueA with the required tuning step, and repeat steps 3~9 until TP = 0 (or command/data error) again. Save the current valueA as value_end.
12. Clear MIX_CTRL[EXE_TUNE] to 0.
13. Set SYS_CTRL[RSTA] to 1, and wait for self-clear.
14. Set MIX_CTRL[SMP_CLK_SEL] to 1.
15. Set bit[14:0] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) with the output of expression {[(((value_start+value_end)/2)< < 8)& 0xffffff00] -0x300 } | 0x33.
16. Poll bit[30:16] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) until the value written in the previous step is read.
17. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

The manual tuning procedure defines a passing window, inside which the tuned clock can capture the correct data, and delay chain is to be configured as centered within the passing window.

It is not a fixed flow as the standard tuning. Any step can be changed based on the real case. In cases where passing window is not closed with a tuning fail, the center of the passing region must be applied for the delay chain setting. In cases where two passing windows separated by tuning fail(s) is observed, the center of the first passing window must be applied for the delay chain setting.

NOTE

Both the standard tuning and manual tuning are only for tuning of SD_CLK, cannot be used to tune STROBE clock of HS400 and HS400 enhanced mode.

40.3.10.4 Switch to HS400 mode

The valid IO voltage for HS400 is 1.8 V. The bus width is set to only DDR 8bit in HS400 mode. HS400 supports the same commands as DDR52. To switch to HS400 mode, the host must perform the following steps:

1. Initialize chip with “Backward Compatible Timings", after power-on or software reset (CMD0), the interface timing of the chip is set as the default "Backward Compatible Timing".
2. Select the chip with CMD7.
3. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate whether the chip supports HS400 by sending CMD8.
4. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported chip's Driver Strengths by sending CMD8.

NOTE

You may skip this step if changes of driver strength is not needed.

5. Set the "Selected Driver Strength" parameter in the HS_TIMING [185] field of the Extended CSD register to the appropriate driver strength for HS400 operation and set the "Timing Interface" parameter to 0x2 to switch to HS200 mode with CMD6. Send CMD13 to make sure mode switch to HS200 mode has been successfully done.
6. Perform the tuning procedure (standard or manual) at the HS400 target operating frequency, save the tuning result (CLK_TUNE_CTRL_STATUS[30:16]), clear TUNING_CTRL[STD_TUNING_EN] to 0, set SYS_CTRL[RSTT] to 1 and wait for self-clear.

NOTE

Tuning procedure in HS200 mode is required to synchronize the command response on the CMD line to CLK for HS400 operation.

7. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x1 to switch to High Speed mode with CMD6 and then set the clock frequency to a value not greater than 52 MHz.

8. Set BUS_WIDTH[183] to 0x06 to select the dual data rate x8 bus mode. Check the setting of PROT_CTRL[DTW]. Make sure the data transfer width is set as expected.
9. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch to HS400 mode with CMD6. Set the MIX_CTRL[DDR_EN] and MIX_CTRL[HS400_MODE] to 1 to enable HS400 mode in controller.
10. Set bit[14:0] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) with saved tuning result value of 0x68 from standard tuning flow in step 6).
11. Poll bit[30:16] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) until the value written in the step 10 is read.
12. Set MIX_CTRL[SMP_CLK_SEL] to 1'b1 to apply the delay on capture clock of CMD channel, then set the clock frequency to ≤ 200 MHz.

    Ensure that uSDHC function clock source is 2x HS400 mode target frequency, and both SDCLKFS and DVS are configured to 0.
13. Set Strobe DLL control (STROBE_DLL_CTRL) to 0x00400039 to have strobe clock automatically tuned to 1/4 cycle of input ipp_strobe to capture data. Ensure that both STROBE_DLL_STATUS[STROBE_DLL_STS_REF_LOCK] and STROBE_DLL_STATUS[STROBE_DLL_STS_SLV_LOCK] are set before any transfer is started.

NOTE

For speed mode switch like from High Speed mode to HS200 mode or HS400(enhanced) mode, you must calibrate the right configuration of slew rate and drive strength for chip pad that are used for uSDHC.

40.3.10.5 Switch to HS400 enhanced mode

This selection flow describes how to initialize the eMMC device in HS400 mode while enabling enhanced strobe without the need for tuning procedure. To switch to HS400 mode with enhanced strobe, the host must perform the following steps:

1. Initialize device with "Backward Compatible Timings", after power-on or software reset (CMD0), the interface timing of the device is set as the default "Backward Compatible Timing".
2. Select the device with CMD7
3. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate whether the device supports HS400 with CMD8.
4. Read the STROBE_SUPPORT[184] field of the Extended CSD register to validate whether the device supports enhanced strobe with CMD8.
5. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x1 to switch to High Speed mode by sending CMD6 and then set the clock frequency to a value not greater than 52 MHz.
6. Set BUS_WIDTH[183] to 0x86 to select the dual data rate x8 bus mode and enable enhanced strobe (this is active only after HS400 mode is selected) with CMD6. Check the setting of PROT_CTRL[DTW]. Make sure the data transfer width is set as expected.
7. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported device Driver Strengths with CMD8.

NOTE

You may skip this step if changes of driver strength is not needed, if needed host may change the device Driver Strength.

8. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch to HS400 mode with CMD6.
9. Set the MIX_CTRL[DDR_EN], MIX_CTRL[HS400_MODE], and MIX_CTRL[EN_HS400_MODE] bits to 1 to enable enhanced HS400 mode in controller.

10. Host may set the clock frequency to a value not greater than 200 MHz. Ensure uSDHC function clock source is 2x HS400 mode target frequency, and both SDCLKFS and DVS are configured to 0 .
11. Set Strobe DLL control (STROBE_DLL_CTRL) to 0x00400039 to have strobe clock automatically delayed 1/4 cycle of input ipp_strobe to capture data. Make sure both STROBE_DLL_STATUS[STROBE_DLL_STS_REF_LOCK] and STROBE_DLL_STATUS[STROBE_DLL_STS_SLV_LOCK] are set before any transfer is started.

NOTE

For speed mode switch like from high speed mode to HS200 mode or HS400(enhanced) mode, you must calibrate the right configuration of slew rate and drive strength for chip pad that are used for uSDHC.

40.3.11 Software restrictions

40.3.11.1 Initialization active

The driver cannot set INITA bit in System Control register when any of the command line or data lines are active, so the driver must ensure both CDIHB and CIHB bits are cleared.

40.3.11.2 Software polling procedure

For polling read or write, after the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in the Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block. For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

40.3.11.3 Suspend operation

To suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by SDIO, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such a 'suspend' command, uSDHC treats the current transfer is stopped and change the BLKCNT register to its original value, instead of retaining the remaining number of blocks.

40.3.11.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

ADMA1 is 4KB aligned.

40.3.11.5 (A)DMA address setting

To configure the ADMA1/ADMA2/DMA address register, when TC bit is set, the register always updates itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring the ADMA1/ADMA2/DMA address register.

40.3.11.6 Data port access

Data port does not support parallel access. For example, during an internal DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or internal DMA. Otherwise the data is corrupted inside the uSDHC buffer.

40.3.11.7 Change clock frequency

The uSDHC module does not automatically gate off the card clock when the host driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing the clock divisor value (SDCLKFS or DVS in System Control Register) or setting the RSTA bit.

Also, before changing the clock divisor value, the host driver should make sure that the SDSTB bit is high.

40.3.11.8 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for eMMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode. In this case, the card may not respond to this extra abort command and uSDHC gets response timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

40.3.12 Clocking

The table found here describes the clock sources for uSDHC. For more information on clocking, see the Clocking chapter.

Table 313. Clocks

Clock name	Description
hclk	Bus clock
ipg_clk	Peripheral clock
ipg_clk_perclk	Base clock
ipg_clk_lp	Low power clock

40.3.12.1 Clock and reset manager

This module controls all four kinds reset signals within uSDHC:

- Hardware reset
- Software reset for all logic
- Software reset for the data logic
- Software reset for the command logic

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

40.3.12.2 Clock generator

The clock generator generates the card CLK by peripheral source clock in two stages. The clock divisor can be configured through register SYS_CTRL[SDCLKFS] for prescaler configuration while the [DVS] for divisor configuration. Details can be found in the register function description. See the following figure for the structure of the divider. The term "Base" represents the frequency of peripheral source clock (see ipg_clk_perclk in Table 313).

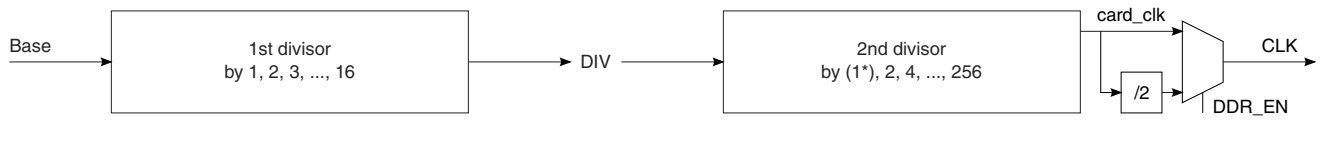


Figure 139. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV) that can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler and outputs the actual internal working clock (card_clk). This clock is the driving clock for all the sub modules of the SD protocol unit, and helps in syncing FIFOs (see Figure 125) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage can be DIV, DIV/2, DIV/4,..., or DIV/256. Therefore, the highest frequency of the card_clk is base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the duty cycle of the base clock is 50%, the duty cycle of card_clk is also 50%, even when the compound divisor is an odd value.

NOTE

CLK is different for the SDR and DDR modes.

- In the SDR mode, CLK is equal to the internal working clock (card_clk).

$$f_{card_clk} = \frac{f_{Base}}{SYS_CTRL[DVS] \times SYS_CTRL[SDCLKFS]}$$

$$f_{CLK} = f_{card_clk}$$

Equation 2. Equation for f_{card_clk} in SDR mode

For HS200 mode, if the source clock f_{Base} = 400 MHz, and the target clock f_{CLK} = 200 MHz, then configuring DVS to 'h00 and SDCLKFS to 'h01, the mapping value of SYS_CTRL[DVS] is 1 and the mapping value of SYS_CTRL[SDCLKFS] is 2.

$$f_{card_clk} = \frac{400MHz}{1x(2)} = 200MHz$$

$$f_{CLK} = f_{card_clk} = 200\ MHz$$

- In the DDR mode, CLK is equal to card_clk/2

$$f_{card_clk} = \frac{f_{Base}}{SYS_CTRL[DVS] \times (SYS_CTRL[SDCLKFS]/2)}$$

$$f_{CLK} = \frac{f_{card_clk}}{2}$$

Equation 3. Equation for f_{card_clk} in DDR mode

For HS400 mode, if the source clock f_{Base} = 400 MHz, and the target clock f_{CLK} = 200 MHz, then configuring DVS to 'h00 and SDCLKFS to 'h00, the mapping value of SYS_CTRL[DVS] is 1 and the mapping value of SYS_CTRL[SDCLKFS] is 2.

$$f_{card_clk} = \frac{400MHz}{1x(2/2)} = 400MHz$$

$$f_{CLK} = \frac{f_{card_clk}}{2} = \frac{400MHz}{2} = 200MHz$$

40.3.13 Command Queuing (CQE) operation

To facilitate command queuing in eMMC, the device manages an internal task queue that the host can queue data transfer tasks. Initially, the task queue is empty. Every task is issued by the host and initially queued as pending. The device controller works to prepare the pending tasks for execution. The software flow is described below:

1. Read CMDQ_SUPPORT[308] and CMDQ_DEPTH[307] of the extended CSD register (through CMD8) to verify eMMC chip support of command queue.
2. Set CMDQ_MODE_EN in CMD_MODE_EN[15] field of the extended CSD by issuing CMD6.

3. Send CMD13 to ensure a successful mode switch.
4. Define Task Descriptor List (TDL) comprised of task descriptor and transfer descriptor sets.
    a. 0x0010002b,0x00000000,0x20000023,0x34300000, //Write example
    b. 0x0010102b,0x00000000,0x20000023,0x34500000, //Read example
5. Set PROT_CTRL[DMASEL] to 2 to select ADMA2. CMDQ descriptors are in ADMA2 format.
6. Set CQSSC2[SSC2] to RCA.
7. Set Command Queuing Task Descriptor List Base Address (CQTDLBA) and Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU) to TDL start address.
8. Set CQCFG[CQUE] to 1 to enable the command queuing.
9. Set CQTDBR[TDBR] field(s) to trigger the required task(s).
10. Read CQTCN[TCN] field(s) to confirm the task completion(s).
11. Set CQTCN[TCN] field(s) to 1 to clear status.

40.3.14 Abort transaction

Abort command

There are two types of abort command: CMD12 and I/O abort.

- CMD12 is used for memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).
- I/O abort is used for SDIO card, SDCombo (I/O portion). I/O abort stops an infinite block transfer (block count = 0) by writing the ASx bit in CCCR.

If an exact number of blocks are to be transferred, it is recommended that the uSDHC issues a block command with the correct block count, rather than using an infinite count and aborting the data at the correct time.

Asynchronous and synchronous abort

There are two ways to issue an abort command (CMD12 or I/O abort): asynchronous abort and synchronous abort.

- In an asynchronous abort sequence, uSDHC can issue an abort command (CMD12 or I/O abort) at anytime unless Command Inhibit (PRES_STATE[CIHB]) in the Present State register is set to 1.
- In a synchronous abort sequence, uSDHC shall issue an abort command (CMD12 or I/O abort) after data transfer is stopped by using Stop At Block Gap Request (SABGREQ).

Read wait

This function uses DATA2 to allow uSDHC to signal the SDIO or Combo Card to temporarily halt the sending of read data by a card during the block gap.

40.4 External signals

The following table describes the uSDHC external signals:

Table 314. uSDHC external signals

Signal	Description	Direction
CLK	Is an Internally generated clock used to drive the eMMC, SD card, and SDIO.	O

Table continues on the next page...

Table 314. uSDHC external signals (continued)

Signal	Description	Direction
CMD	Is used to send commands and receive responses to and from the card.	I/O
DAT7	DAT7 line in the 8-bit mode — Not used in other modes	I/O
DAT6	DAT6 line in the 8-bit mode — Not used in other modes	I/O
DAT5	DAT5 line in the 8-bit mode — Not used in other modes	I/O
DAT4	DAT4 line in the 8-bit mode — Not used in other modes	I/O
NOTE If uSDHC needs to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.		
DAT3	DAT3 line in the 4/8-bit mode or configured as card detection pin. The bit may be configured as card detection pin in the 1-bit mode.	I/O
DAT2	DAT2 line or Read Wait in the 4-bit mode Read Wait in 1-bit mode	I/O
DAT1	DAT1 line in the 4/8-bit mode Also, used to detect interrupt in 1/4-bit mode	I/O
DAT0	DAT0 line in all the modes Also, used to detect busy state	I/O
CD_B	Card detection signal directly routed from the socket. Low value (0) indicates there is a card inserted. In the case the pin is not used (for the embedded memory), tie low. Optional for system implementation.	I
WP	Write protection signals directly routed from the socket. Low value (0) indicates it is not write protected. In the case the pin is not used (for the embedded memory), tie low. Optional for system implementation.	I
RESET_B	Is used to reset the eMMC or control Card power. Active Low. • eMMC: connect eMMC RST_n pin.	O

Table continues on the next page...

Table 314. uSDHC external signals (continued)

Signal	Description	Direction
	• Card: instead of hardware reset to enable/disable Card power during warm reset.	
VSELECT	Is used to change the voltage of the external power supplier. Optional for system implementation.	O
STROBE	Input clock for eMMC HS400 mode. **NOTE** If the uSDHC does not support the HS400 mode, STROBE can also be optional and tied to low.	I

40.5 Application information

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, and ALL_SEND_CID. In the Broadcast mode, eMMC are in the open-drain mode to avoid bus contention. See Commands for SD card, SDIO, and eMMC for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter the standby mode. Addressed type commands are used from this point. In this mode, for eMMC, the CMD/DATA I/O pads turns to the push-pull mode to have the driving capability for maximum frequency operation. See Commands for SD card, SDIO, and eMMC for the commands of ac and adtc categories.

40.5.1 Command send and response receive basic operation

Assuming that the data type WORD is an unsigned 32-bit integer, the flow indicated below presents a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)

{

    WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner

    wCmd = (<cmd_index> & 0x3f) << 24; // set the first 8 bits as '00'+<cmd_index>

    set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL according to the command index;

    if (internal DMA is used) wCmd |= 0x1;

    if (multi-block transfer) {

        set MSBSEL bit;

        if (finite block number) {

            set BCEN bit;

            if (auto12 command is to use) set AC12EN bit;
```

```
        }

    }

    write_reg(CMDARG, <cmd_arg>); // configure the command argument

    write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command

}

wait_for_response(cmd_index)

{

    while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set

    read IRQ Status register and check if any error bits about Command are set

    if (any error bits are set) report error;

    write 1 to clear CC bit and all Command Error bits;

}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure that the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and move to the Standby state no response to the Host when CMD2 is sent. The host driver deals with "fake" errors like this with caution.

40.5.2 Card identification mode

When a card is inserted to the socket or the card is reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for eMMC.

40.5.2.1 Card detect

See Figure 140 for a flow diagram showing the detection of SD card and SDIO using uSDHC.

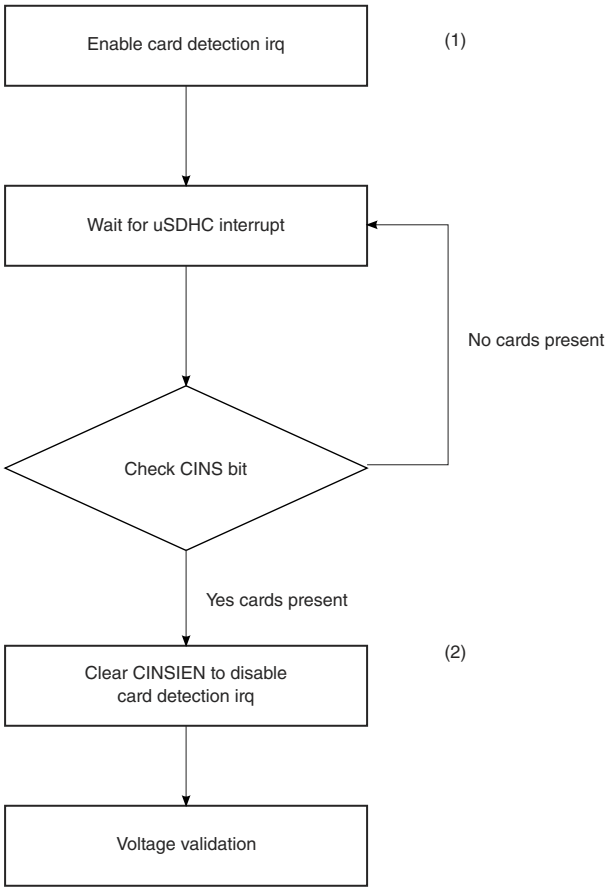


Figure 140. Flow diagram for card detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt.
- When an interrupt from uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion.
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards.

40.5.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) that is driven by Power On Reset (POR).
- Software reset (Host only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (Card only): The command, "Go_Idle_State" (CMD0), is the software reset command for all types of eMMC and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SDIO, CMD52 is used to write an I/O reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both uSDHC and the card.

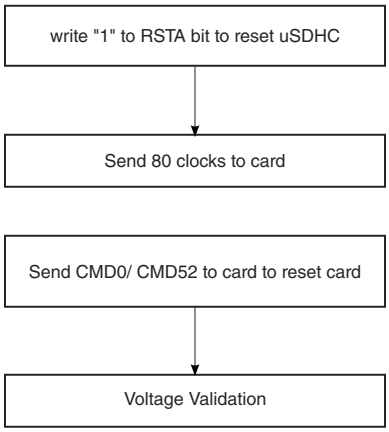


Figure 141. Flow chart for resetting uSDHC and SD card or SDIO

```
software_reset() {
    set_bit(SYSCTRL, RSTA); // software reset the Host set DTOCV and SDCLKFS fields to get the CLK of
    //frequency around 400kHz
    //configure IO pad to set the power voltage of external card to around 3.0V
    //poll bits CIHB and CDIHB
    //bits of PRSSTAT to wait both fields are cleared
    set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
    send_command(CMD_GO_IDLE_STATE, <other parameter>); // reset the card with CMD0 or
    send_command(CMD_IO_RW_DIRECT, <other parameter>);
}
```

40.5.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for VDD are defined in the Operation Conditions Register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer VDD conditions. This means that if the host and card have non-common VDD ranges, the card is neither able to complete the identification cycle nor able to send CSD data.

Therefore, special commands Send_Op_Cont (CMD1 for eMMC), SD_Send_Op_Cont (ACMD41 for SD Memory), and IO_Send_Op_Cont (CMD5 for SD I/O) are used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards that do not match the VDD range(s) desired by the host. This is accomplished when the host sends the desired VDD voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive state. This query should be used if the host is able to select a common voltage range or if a notification is sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_argument) {
    label the card as UNKNOWN;
    send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO operation
voltage, command argument is zero
    if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
        if (0 < number of IO functions) {
            label the card as SDIO;
            IORDY = 0;
```

```
            while (!(IORDY in IO OCR response)) { // set voltage range for each IO function
                send_command(IO_SEND_OP_COND, <voltage range>, <other parameter>);
                wait_for_response(IO_SEND_OP_COND);
            } // end of while ...
        } // end of if (0 < ...
        if (memory part is present inside SDIO)
            Label the card as SDCombo; //this is an SD-Combo card
    } // end of if (RESP_TIMEOUT ...
    if (the card is labeled as SDIO)
        return; // card type is identified and voltage range is set, so exit the function;
    send_command(APP_CMD, 0x0, <other parameter>); // CMD55, Application specific CMD prefix
    if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
        send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage range for
memory part or SD card
        wait_for_response(SD_APP_OP_COND); // voltage range is set
        if (card type is UNKNOWN)
           label the card as SD;
        return; //
    } // end of if (no error ...
    else if (errors other than time-out occur) { // command/response pair is corrupted
        deal with it by program specific manner;
    } // of else if (response time-out
    else { // CMD55 is refuse, it must be eMMC
        if (card is already labeled as SDCombo) { // change label
            re-label the card as SDIO;
            ignore the error or report it;
            return; // card is identified as SDIO
        } // end of if (card is ...
        send_command(SEND_OP_COND, <voltage range>, <...>);
        if(RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
            either label the card as UNKNOWN;
            return;
        } // end of if (RESP_TIMEOUT ...
    } // end of else
}
```

40.5.2.4 Card registry

This section briefly describes the registry flow. For details, please refer to the card specifications. Card registry for the eMMC and SD/SDIO/SD combo cards are different. For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the card spec). Currently, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host requests the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command is sent to all the new cards in the system. Incompatible cards are put into the Inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA is used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in the system.

For eMMC operation, the host starts the card identification process in the open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line to allow parallel card operation during card identification. After the bus is activated, the host requests the cards to send their valid operation conditions

(CMD1). The response to CMD1 is the "wired AND " operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in the Ready state) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. As the CID is unique for each card, only one card can be successfully sent its full CID to the host. This card then goes into the Identification state. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign a relative card address (RCA) to the card. After the RCA is received, the state of the card changes to standby, and the card does not react in further identification cycles. Also, its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

The following steps show how to perform an operation using eMMC:

```
card_registry() {
    do { // decide RCA for each card until response time-out
        if(card is labelled as SDCombo or SDIO) { // for SDIO like device
            send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO to publish its RCA
            retrieve RCA from response;
        } // end if (card is labelled as SDCombo ...
        else if (card is labelled as SD) { // for SD card
            send_command(ALL_SEND_CID, <...>);
            if(RESP_TIMEOUT == wait_for_response(ALL_SEND_CID))
                break;
            send_command(SET_RELATIVE_ADDR, <...>);
            retrieve RCA from response;
        } // else if (card is labelled as SD ...
        else if (card is labelled as eMMC) {
            send_command(ALL_SEND_CID, <...>);
            rca = 0x1; //arbitrarily set RCA, 1 here for example
            send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper 16 bits
        } // end of else if (card is labelled as eMMC...
    } while (response is not time-out);
}
```

40.5.3 Card access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

40.5.3.1 Block write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

40.5.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card indicates that the failure on the DATA line. The transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and stops the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation also stops if the host tries to write over a write-protected area.

For eMMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO at any time, and the card responds with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby state and release the DATA line without interrupting the write operation. When re-selecting the card, it reactivates the busy indication by pulling data to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card that incorporates the internal DMA and the write operation is a multi-block write. For the other methods (CPU polling status with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving, if the device is memory device: such as eMMC, SD Card, SD Combo Card (memory portion).

40.5.3.1.2 DDR write

uSDHC supports the dual data rate mode.

The software flow to write to a card in the DDR mode is described as below:

1. Check the card status and wait until the card is ready for data.

   NOTE

   For eMMC, block length can only be set to 512 Byte.
2. Set the uSDHC number block register (NOB), where nob is 5, for instance.
3. Set the eMMC, SD card to high-speed mode and use SWITCH (CMD6); or set SDIO to high-speed mode and use IO_RW_DIRECT (CMD52).
4. Set the eMMC bus or SD with 4-bit/8-bit DDR mode and use SWITCH (CMD6); or set SDIO to with 4-bit DDR mode and use IO_RW_DIRECT (CMD52).
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. Set DDR_EN to 1.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.

7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).

40.5.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Similar to the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status and wait until the card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).

The number of blocks left during the data transfer is accessible by reading the contents of BLK_ATT[BLKCNT]. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The host driver reads the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt (INT_STATUS[TC]) is received.

It is also possible that the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is the end of the transfer. These types of requests are ignored, the driver should treat these as a non-pause transfer, and deal with it as a common write operation.

40.5.3.2 Block read

Information about Normal read, DDR read, Read with Pause, and Delay Line (DLL) in Read Path are detailed in the sections below.

40.5.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks are continuously transferred until a stop command is issued.

The software flow to read from a card that incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other methods (CPU polling status with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status and wait until card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo Cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer read ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or any other error occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

40.5.3.2.2 DDR read

The uSDHC module supports dual data rate mode.

The software flow to write to a card in the DDR mode is described below:

1. Check the card status and wait until the card is ready for data.

   NOTE

   For eMMC, block length can only be set to 512 Bytes.
2. Set the uSDHC number block register (NOB) where nob is 5, for instance.
3. Set the eMMC, SD card to high-speed mode and use SWITCH (CMD6); or set SDIO to high-speed mode and use IO_RW_DIRECT (CMD52).
4. Set the eMMC bus or SD with 4-bit/8-bit DDR mode and use SWITCH (CMD6); or set SDIO to with 4-bit DDR mode and use IO_RW_DIRECT (CMD52).
5. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. Set DDR_EN to 1.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.

7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

40.5.3.2.3 Read with Pause

The read operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in Control register to pause the transfer between the data blocks.

Only SDIO and SD Combo Card support Read Wait feature while Read with Pause. Read Wait feature means that it uses DATA2 to allow uSDHC to signal the SDIO or SD Combo Card to temporarily halt the sending of read data by a card during the block gap. Before setting SABGREQ to 1, ensure to set the RWCTL bit to 1 after recognizing the read wait capability of SDIO (SRW bit in CCCR register is 1).

Similar to the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCCR register on SDIO or SD Combo Card (I/O portion) to confirm that the card supports the Read Wait mode.
2. If SDIO or Combo Card (I/O portion) supports Read Wait, set RWCTL to 1.

   If the device is meemory device: such as eMMC, SD Card, or SD Combo Card (Memory portion), set RD_DONE_NO_8CLK to 1.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo Cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bitfield in the CCCR register (for function 0) or FBR register (for functions 1~7).
5. Set the uSDHC block length register to be the same as the block length set for the card in step 4.
6. Set the uSDHC number block register (NOB), where nob is 5, for instance.
7. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer read ready interrupt, configure the DMA setting, and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or any other error occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

Similar to the Write operation, it is possible to meet the ending block of the transfer when paused. In this case, uSDHC ignores the Stop At Block Gap Request and treats it as a command read operation.

If the Suspend command is sent and the transfer is later resumed by means of a Resume command, uSDHC takes the command as a normal one accompanied with data transfer. It is left for the host driver to set all the relevant registers before the transfer

is resumed. For momory device, if there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. Then uSDHC automatically sends CMD12 to mark the end of multi-block transfer.

40.5.3.2.4 Delay Line (DLL) in read path

The DLL is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

The reasons why DLL is needed for uSDHC are these:

- The path of read data traveling from card to host varies.
- In the eMMC and SD cards DDR mode, the minimum input setup and hold time are both at 2.5 ns.

The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in the override mode. The override value set is the number of delay cells. In the override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL automatically adjusts the number of delay cells according to the target value set by the user and PVT changes. Be aware that the target value is in units of 1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns; setting target value of 16 means 5ns =(16/32)*10ns. The software can disable automatic update by the setting dll_gate_update bit.

As you might change the frequency of card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. If DLL is to be used, make sure SDCLKFS is used (at least be set to divided by 2) so that the REF_CLK generated are the same frequency as card_clk. There are two DLLs, PARTS DLL and STROBE DLL. PARTS DLL controls the SD_CLK loop back delay for which DLL (Delay Line) Control (DLL_CTRL) and DLL Status (DLL_STATUS) are used. STROBE DLL controls the STROBE loop back delay for which Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) are used. For PARTS DLL, below flow can be taken as reference. For STROBE DLL, similar register field with prefix STROBE in Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) must be used.

Step 1: Set the DLL_CTRL_RESET and DLL_CTRL_ENABLE fields

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: Clear the DLL_CTRL_RESET field

Step 5: Wait until both the DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: Set the DLL_CTRL_SLV_FORCE_UPD

Step 7: Clear the DLL_CTRL_SLV_FORCE_UPD

NOTE

The software should make sure that the DLL_CTRL_SLV_FORCE_UPD lasts for at least one card_clk. So, the software may need to add some delay between step 6 and step 7.

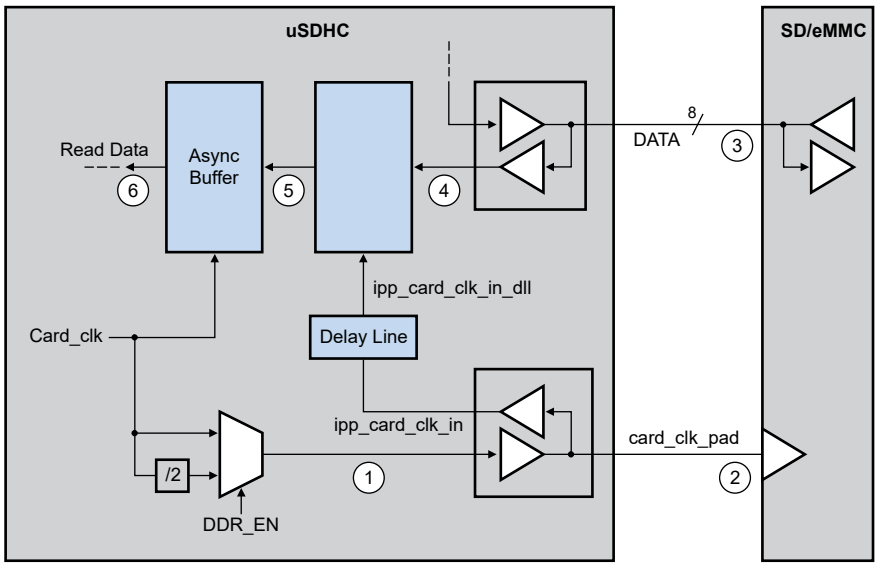


Figure 142. DLL in read path

40.5.3.3 Suspend Resume

The uSDHC module supports the Suspend Resume operations of SDIO and Combo Card, although slightly different than the suggested implementation of Suspend in the SDIO specification. Both Read and Write support Suspend and Resume, however ADMA operation does not support this function.

40.5.3.3.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of SDIO and Combo Card. The uSDHC module does not monitor the content of the response, so it does not know if the Suspend command succeeded or not. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the host driver does not mark the Suspend command as "Suspend", i.e., setting the CMDTYP bits to 01. Instead, the driver sends this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform uSDHC that the current transfer is suspended. Here is the sequence for the Suspend operation:

1. Check CCCR[SRW] on SDIO or SD Combo Card (I/O portion) to confirm that the card supports the Read Wait mode. If SDIO or Combo Card (I/O portion) supports Read Wait, set the RWCTL bit to 1.
2. Set the SABGREQ bit to 1 to pause the current data transfer at block gap. Wait for the BGE bit set as 1, and wait for the Transfer Complete interrupt. Save the context registers in the system memory for later use, including the DMA System Address Register DS_ADDR (for internal DMA operation), and the Block Attribute Register BLK_ATT.
3. Send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.

   The following steps show an example of CMD52 function0 suspend with CMDTYP=2'b00:

   a. Set CMD_ARG to 0x88001802

   b. Set MIX_CTRL to 0x80000000

   c. Set CMD_XFR_TYP to 0x341A0000
4. Check the BS bit of the CCCR in the response (CMD_RSP0[0]). If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
5. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so uSDHC can detect this special setting and be informed that the paused operation has successfully suspended.

The following steps show an example of CMD52 function0 suspend with CMDTYP=2'b01:

a. Set CMD_ARG to 0x00001800

b. Set MIX_CTRL to 0x80000000

c. Set CMD_XFR_TYP to 0x345A0000

6. Clear the BGE bit to 0, and clear the SABGREQ bit to 0. Begin operation for another function on SDIO and Combo Card (I/O portion).

40.5.3.3.2 Resume

To resume the data transfer, a Resume command is issued:

1. To resume the suspended function, restore the context registers (DS_ADDR and BLK_ATT) with the saved value in the step #2 of the Suspend operation.
2. Send the Resume command: In the Transfer Type register, all the bitfields are set to the value as if this were another ordinary data transfer instead of a transfer resume (except the CMDTYP is set to 2'b10).

   The following steps show an example of CMD52 function0 resume with CMDTYP=2'b10:

   a. Set CMD_ARG to 0x88001A01

   b. Set MIX_CTRL either to 0x80000023 (Suspend Write, Host to Card) or to 0x80000033 (Suspend Read, Card to Host)

   c. Set CMD_XFR_TYP to 0x34BA0000

3. If the Resume command (INT_STATUS[CC]) has responded, the data transfer is resumed.

40.5.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command. The steps to prepare the correct descriptor chain are as follows:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4KB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors matches the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.
8. Once the ADMA transfer is complete, clear PROT_CTRL[DMASEL] to 00b.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

40.5.3.5 Transfer error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 errors are detailed in the sections below.

40.5.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received is discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one. For a multi-block transfer, the host driver issues a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, uSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 is sent by uSDHC. In this case, the host driver re-sends or re-obtains the last block with a single block transfer.

40.5.3.5.2 Internal DMA error

During the data transfer with internal Simple DMA, if the DMA engine encounters an error on the AXI bus, the DMA operation stops, and the DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the host driver calculates the start address of data block in which the error occurs. The start address can be calculated by either:

- Reading the DMA System Address register: The error occurs during the previous burst. Considering the block size, and the start address of the next burst transfer, it is straight forward to obtain the start address of the corrupted block.
- Reading the BLKCNT field of the Block Attribute register: Considering the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined.To use this method, MIX_CTRL[BCEN] bit has to be set to enable Block Attribute register update.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

40.5.3.5.3 Transfer ADMA error

There are three kinds of possible ADMA errors: The AXI transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

- AXI transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or to the next block if no block is corrupted.
- Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and recreate the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

40.5.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, uSDHC automatically sends a CMD12 to the card to stop the transfer. When errors with this command occur, it is recommended to the host driver to deal with the situations in the following manner:

- Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The host driver clears the auto CMD12 error status bits and re-send CMD12 until it is accepted by the card.
- Auto CMD12 response CRC error: As the card responds to CMD12, it stops the transfer. The host driver may ignore the error and clear the error status bit.
- Auto CMD12 conflict error or not sent: The command is not sent; therefore, the host driver sends a CMD12 manually.

40.5.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For SDIO, it can be the low-level on the DATA1 line during some special period. The uSDHC module only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by uSDHC, and the host system is informed by uSDHC asserting the uSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt source is controlled by the external card, the interrupt from SDIO must be serviced before the CINT bit is cleared. Refer to Card interrupt handling for the card interrupt handling flow.

40.5.4 Switch function

A switch command is issued by the host driver to enable new features added to the eMMC and SD cards. The eMMC and SD cards can transfer data at bus widths other than 1-bit. Different speed modes are also defined. To enable these features, a switch command is issued by the host driver.

For SDIO, the high-speed mode /DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed as high. For SD cards, the high-speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For eMMC, the high-speed mode HS200 or HS400 is queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit, and DDR8-bit width of eMMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudo-code examples do not show current capability check, which is recommended in the function switch process.

40.5.4.1 Query, enable, and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report SDIO does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of the desired value below 25MHz;
(data transactions like normal peers)
}
```

40.5.4.2 Query, enable, and disable SD high-speed mode/DDR50 /SDR50 /SDR104

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;(high speed
mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
```

```
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
      send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of around 50MHz for high speed mode, 100MHz for SDR50, 200MHz for SDR104, 50MHz for DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

40.5.4.3 Query, enable, and disable eMMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support high speed mode and return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this eMMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report eMMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

40.5.4.4 Set eMMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support multiple bit width and return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-bit,
x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

40.5.5 ADMA operation

Here are the codes for the ADMA1 and ADMA2 operations.

40.5.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

40.5.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
```

```
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

40.5.6 Fast boot operation

40.5.6.1 Normal fast boot flow

Here are the steps for normal fast boot flow:

1. Software must configure the SYS_CTRL[INITA] bit to make sure that 74 card clocks are finished.
2. Software must configure the eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode. If the data is sent through the DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set Data Transfer Width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN retain the default value, where DPSEL bit is set to 1, DTDSEL is set to 1 and MSBSEL is set to 1.
7. DMAEN should be configured as 0 in the polling mode and if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When step 6 is configured, the boot process begins. The software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt is triggered, and the software must configure eMMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure eMMC Boot Register bit 6 to 0 to disable boot. This render CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin the normal initialization process.
10. You must reset the host and begin the normal process.

40.5.6.2 Alternative fast boot flow

Here are the steps for alternative fast boot flow:

1. Software needs to configure SYS_CTRL[INITA] to make sure 74 card clocks are finished.
2. Software needs to configure eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through the DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in the DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software needs to configure the Protocol control register to set the data transfer width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with the 0xFFFFFFFA argument. In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. Note DMAEN should be configured as 0 in the polling mode, and if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in the DDR fast boot mode, DDR_EN needs to be set to 1.
7. When step 6 is configured, the boot process begins. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host sends out the interrupt and the software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process.
8. If there is no time out, the software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed, configure the eMMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the target (card) is ready for the identification step.
9. You must reset the host and begin the normal process.

40.5.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during eMMC fast boot.

In fast boot, the host can use Advanced DMA2 (ADMA2) with two destinations.

The detailed flow is described below:

1. The software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. The software needs to configure the eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1). Here VALUE1 is the value of the block count that needs to transfer the first time, so that the host stops at the block gap when the uSDHC controller gets VAULE1 blocks from the device. Also, configure bits[3-0] to select the ack timeout value according to the SD clock frequency.
3. The software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. The software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enable ADMA2 by configuring Protocol Control Register bits [9-8].
6. The software needs to set at least three pairs of ADMA2 descriptor in boot memory (that is, in IRAM, at least six words). The first pair descriptor defines the start address (that is, IRAM) and data length (that is,512byte*VALUE1) of the first part boot code. The software also needs to set the second pair descriptor, the second start address (any value that

is writable), and data length is suggested to set 1~2word (record as VALUE2). Note that the second couple desc also transfers useful data even at lease 1 word, because our ADMA2 cannot support 0 data_length data transfer descriptor.

7. The software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot and do not need to be set in normal fast boot.
8. The software needs to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in the DMA mode. And, if BCEN is configured as 1, then configure blk no in Bock Attributes Register to the max value. And, if in the DDR fast boot mode, DDR_EN needs to be set to 1.
9. When step 8 is configured, boot process begins, the first VALUE1 block number data gets transferred. The software needs to poll the TC bit (bit1 in Interrupt Status Register) to determine first transfer is ended. Also, the software needs to polling the BGE bit (bit2 in Interrupt Status Register) to determine if the first transfer stops at the block gap.
10. When TC and BGE bits are set to 1, the software can analyze the first code of VALUE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of the boot code (VALUE3, the remain boot code block). Remember to set the last descriptor with END.
11. The software needs to configure the eMMC Boot Register (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the ack mode or not. In the DMA mode, configure bit 7 to 1 for enabling the automatically stop at block gap feature. Also, configure bit31-bit16 to set the (BLK_CNT - (VALUE1+1+VALUE3)), that host stops at block gap when the uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequency. Note that the software does not need to configure the BLK_CNT again, because it is counted down automatically by the uSDHC controller.
12. The software needs to clear the TC and BGE bits and the software needs to clear SABGREQ (bit 16 in the Protocol control register) and set CREQ (bit17 in the Protocol control register) to 1 to resume the data transfer. Host transfers the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. The software needs to do poll BGE bit to determine if the fast boot is over.

Note:

- When ADMA boot flow starts, for uSDHC, it is like a normal ADMA read operation. So, set ADMA2 descriptor as the normal ADMA2 transfer.
- Need a few words length memory to keep descriptor.
- For the 1~2-word data in second descriptor setting, it is a useful data, so the software needs to deal the data because of the application case.

40.6 uSDHC memory map and register definition

40.6.1 USDHC register descriptions

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map of uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

40.6.1.1 USDHC memory map

USDHC0 base address: 4041_2000h

USDHC1 base address: 4041_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DMA System Address (DS_ADDR)	32	RW	0000_0000h
4h	Block Attributes (BLK_ATT)	32	RW	0001_0000h
8h	Command Argument (CMD_ARG)	32	RW	0000_0000h
Ch	Command Transfer Type (CMD_XFR_TYP)	32	RW	0000_0000h
10h	Command Response0 (CMD_RSP0)	32	R	0000_0000h
14h	Command Response1 (CMD_RSP1)	32	R	0000_0000h
18h	Command Response2 (CMD_RSP2)	32	R	0000_0000h
1Ch	Command Response3 (CMD_RSP3)	32	R	0000_0000h
20h	Data Buffer Access Port (DATA_BUFF_ACC_PORT)	32	RW	0000_0000h
24h	Present State (PRES_STATE)	32	R	See section
28h	Protocol Control (PROT_CTRL)	32	RW	0880_0020h
2Ch	System Control (SYS_CTRL)	32	RW	0080_800Fh
30h	Interrupt Status (INT_STATUS)	32	RW	0000_0000h
34h	Interrupt Status Enable (INT_STATUS_EN)	32	RW	0000_0000h
38h	Interrupt Signal Enable (INT_SIGNAL_EN)	32	RW	0000_0000h
3Ch	Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)	32	RW	0000_0000h
40h	Host Controller Capabilities (HOST_CTRL_CAP)	32	RW	07F3_B407h
44h	Watermark Level (WTMK_LVL)	32	RW	0810_0810h
48h	Mixer Control (MIX_CTRL)	32	RW	8000_0000h
50h	Force Event (FORCE_EVENT)	32	RW	0000_0000h
54h	ADMA Error Status (ADMA_ERR_STATUS)	32	R	0000_0000h
58h	ADMA System Address (ADMA_SYS_ADDR)	32	RW	0000_0000h
60h	DLL (Delay Line) Control (DLL_CTRL)	32	RW	0000_0000h
64h	DLL Status (DLL_STATUS)	32	R	0000_0200h
68h	CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)	32	RW	0000_0000h
70h	Strobe DLL control (STROBE_DLL_CTRL)	32	RW	0000_0000h
74h	Strobe DLL status (STROBE_DLL_STATUS)	32	R	0000_0200h
C0h	Vendor Specific Register (VEND_SPEC)	32	RW	3000_7809h
C4h	eMMC Boot (MMC_BOOT)	32	RW	0000_0000h
C8h	Vendor Specific 2 Register (VEND_SPEC2)	32	RW	0001_9006h
CCh	Tuning Control (TUNING_CTRL)	32	RW	0021_2800h
100h	Command Queuing Version (CQVER)	32	R	0000_0510h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
104h	Command Queuing Capabilities (CQCAP)	32	RW	0000_310Ah
108h	Command Queuing Configuration (CQCFG)	32	RW	0000_0000h
10Ch	Command Queuing Control (CQCTL)	32	RW	0000_0000h
110h	Command Queuing Interrupt Status (CQIS)	32	RW	0000_0000h
114h	Command Queuing Interrupt Status Enable (CQISTE)	32	RW	0000_0000h
118h	Command Queuing Interrupt Signal Enable (CQISGE)	32	RW	0000_0000h
11Ch	Command Queuing Interrupt Coalescing (CQIC)	32	RW	0000_0000h
120h	Command Queuing Task Descriptor List Base Address (CQTDLBA)	32	RW	0000_0000h
124h	Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)	32	RW	0000_0000h
128h	Command Queuing Task Doorbell (CQTDBR)	32	RW	0000_0000h
12Ch	Command Queuing Task Completion Notification (CQTCN)	32	RW	0000_0000h
130h	Command Queuing Device Queue Status (CQDQS)	32	R	0000_0000h
134h	Command Queuing Device Pending Tasks (CQDPT)	32	R	0000_0000h
138h	Command Queuing Task Clear (CQTCLR)	32	RW	0000_0000h
140h	Command Queuing Send Status Configuration 1 (CQSSC1)	32	RW	0001_1000h
144h	Command Queuing Send Status Configuration 2 (CQSSC2)	32	RW	0000_0000h
148h	Command Queuing Command Response for Direct-Command Task (CQCRDCT)	32	R	0000_0000h
150h	Command Queuing Response Mode Error Mask (CQRMEM)	32	RW	FDF9_A080h
154h	Command Queuing Task Error Information (CQTERRI)	32	R	0000_0000h
158h	Command Queuing Command Response Index (CQCRI)	32	R	0000_0000h
15Ch	Command Queuing Command Response Argument (CQCRA)	32	R	0000_0000h

40.6.1.2 DMA System Address (DS_ADDR)

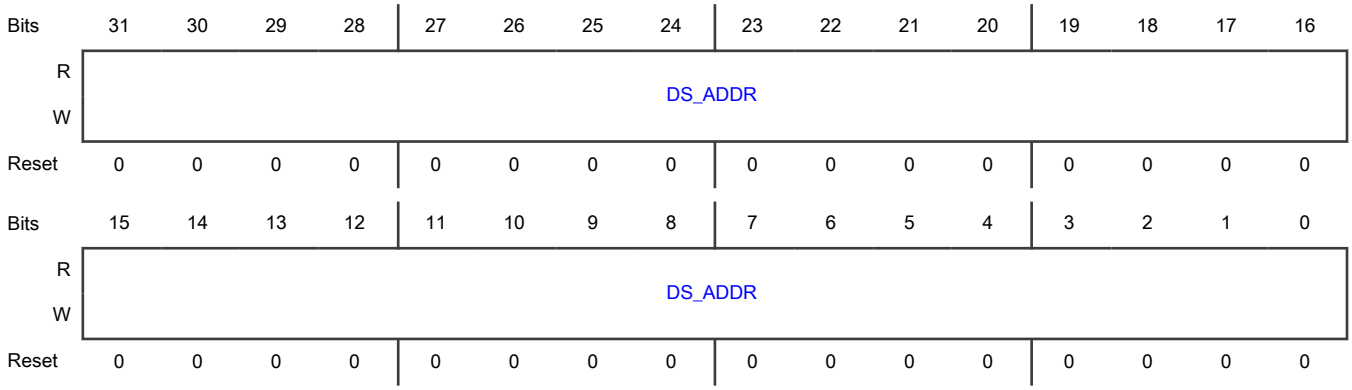
Offset

Register	Offset
DS_ADDR	0h

Function

This register contains the physical system memory address used for DMA transfers.

Diagram



Fields

Field	Function
31-0 DS_ADDR	System address **DMA system address / argument 2** When ACMD23_ARGU2_EN is set to 0, SDMA uses this register as system address and supports only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When ACMD23_ARGU2_EN is set to 1, SDMA uses ADMA System Address register (05Fh – 058h) instead of this register to support both ADMA1 and ADMA2. This register is used only for Argument2 and SDMA may use Auto CMD23. 1. **SDMA system address** Because the address must be word (4 bytes) aligned, the least two bits are reserved and always set to 0. When uSDHC stops a DMA transfer, this register points out the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The host driver initializes this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver waits until the DLA field in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions. 2. **Argument 2** This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

40.6.1.3 Block Attributes (BLK_ATT)

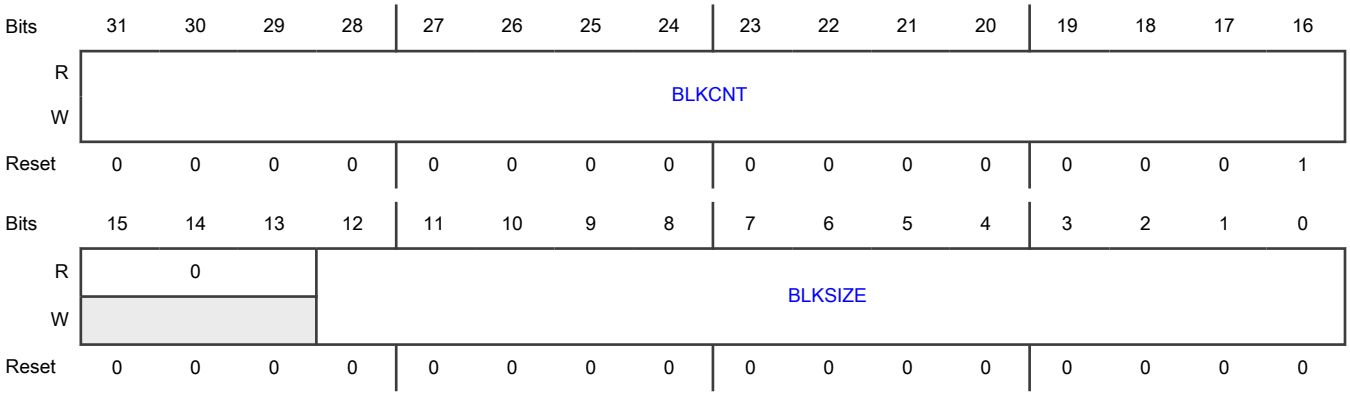
Offset

Register	Offset
BLK_ATT	4h

Function

This register is used to configure the number of data blocks and the number of bytes in each block.

Diagram



Fields

Field	Function
31-16 BLKCNT	Blocks count for current transfer This field is enabled when the Block Count Enable field in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this field always reads as 1. The host driver sets this field to a value between 1 and the maximum block count. The uSDHC module decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to zero results in no data blocks being transferred. This field should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this field may return an invalid value and write operations are ignored. When saving transfer content because of a Suspend command, the number of blocks yet to be transferred can be determined by reading this field. The reading of this field should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when the Suspend command is sent out, uSDHC treats the current transfer as stopped and change the BLKCNT field back to its original value instead of keeping the dynamical indicator of the remaining block count. When restoring transfer content prior to issuing a Resume command, the host driver restores the previously saved block count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL field is indicating a single block transfer, the read value of BLKCNT is always 1. 0000_0000_0000_0000b - Stop count 0000_0000_0000_0001b - 1 block 0000_0000_0000_0010b - 2 blocks 1111_1111_1111_1111b - 65535 blocks
15-13 —	Reserved
12-0 BLKSIZE	Transfer block size This field specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored. 0_0000_0000_0000b - No data transfer 0_0000_0000_0001b - 1 byte 0_0000_0000_0010b - 2 bytes 0_0000_0000_0011b - 3 bytes 0_0000_0000_0100b - 4 bytes 0_0001_1111_1111b - 511 bytes 0_0010_0000_0000b - 512 bytes 0_1000_0000_0000b - 2048 bytes 1_0000_0000_0000b - 4096 bytes

40.6.1.4 Command Argument (CMD_ARG)

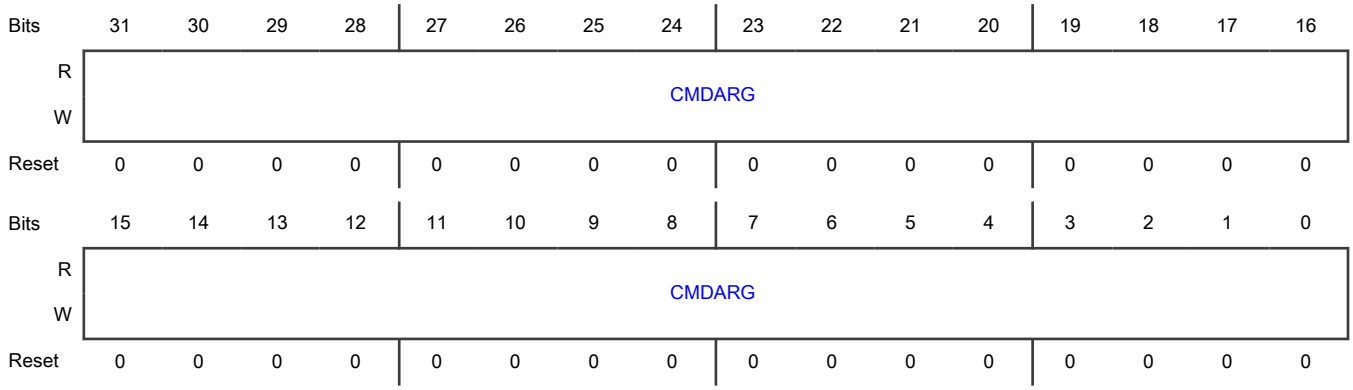
Offset

Register	Offset
CMD_ARG	8h

Function

This register contains the SD/eMMC command argument.

Diagram



Fields

Field	Function
31-0 CMDARG	Command argument The SD/eMMC command argument is specified as bits 39-8 of the command format in the SD or eMMC specification. This field is write protected when the Command Inhibit (CMD) field in the Present State register is set.

40.6.1.5 Command Transfer Type (CMD_XFR_TYP)

Offset

Register	Offset
CMD_XFR_TYP	Ch

Function

This register is used to control the operation of data transfers. The host driver sets this register before issuing a command followed by a data transfer or before issuing a Resume command. To prevent data loss, uSDHC prevents writing to the bits, which are involved in the data transfer of this register, when data transfer is active. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver checks the Command Inhibit DAT field (PRES_STATE[CDIHB]) and the Command Inhibit CMD field (PRES_STATE[CIHB]) in the Present State register before writing to this register. When the CDIHB field in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB field is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as a single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC ignores the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (PRES_STATE[WPSPL] field of Present State register is '1'); otherwise, WP is activated as '1', write to memory and combo cards are ignored by cards (in that case, there is no need to send write command anymore).

If the commands with data transfer do not receive the response in 64 clock cycles, that is, if response time-out happens, uSDHC treats the external device, does not accept the command, and stops the data transfer. In this scenario, the driver should issue the command again to retry the transfer. It is also possible that for some reason the card responds to the command but uSDHC does

not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 315. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

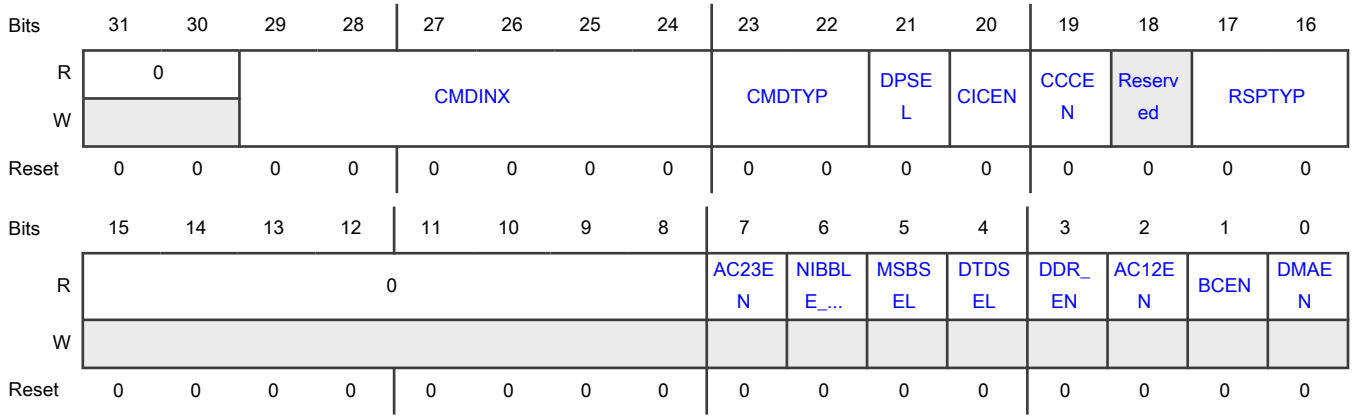
The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, regarding the Response Type bits as well as the name of the response type.

Table 316. Relationship between parameters and the name of the response type

Response type	Index check enable	CRC check enable	Name of response type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification, but R5b is defined in this specification to specify that uSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command is used with R5b.
- The CRC fields for R3 and R4 are expected to be all 1 bits. The CRC check is disabled for these response types.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 CMDINX	Command index These fields are set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Specification.
23-22 CMDTYP	Command type There are three types of special commands: Suspend, Resume, and Abort. These fields are set to 00b for all other commands. • Suspend command: If the Suspend command succeeds, uSDHC assumes that the card bus has been released and that it is possible to issue the next command that uses the DATA line. Because uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the host driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform uSDHC that a Suspend command was successfully issued. See Suspend Resume for more details. After the end bit of command is sent, uSDHC deasserts Read Wait for read transactions and stops checking busy for write transactions. In a 4-bit mode, the interrupt cycle starts. If the Suspend command fails, uSDHC maintains its current state, and the host driver restarts the transfer by setting the Continue Request field in the Protocol Control register. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the Suspend command and then sends the Resume command. The uSDHC module checks for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, uSDHC stops reads to the buffer. If this command is set when executing a write transfer, uSDHC stops driving the DATA line. After issuing the Abort command, the host driver should issue a software reset (Abort Transaction). 00b - Normal other commands 01b - Suspend CMD52 for writing bus suspend in CCCR 10b - Resume CMD52 for writing function select in CCCR 11b - Abort CMD12, CMD52 for writing I/O Abort in CCCR
21 DPSEL	Data present select This field is set to 1 to indicate that data is present and is transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b (for example, CMD38)) **NOTE** In resume command, this field is set, and other bits in this register is set the same as when the transfer was initially launched. When the write protect switch is on, (that is, the WPSPL field is active as '0'), any command with a write operation ignored. When this field is set, while the DTDSEL field is 0, writes to the register Transfer Type are ignored.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No data present 1b - Data present
20 CICEN	Command index check enable If this field is set to 1, uSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this field is set to 0, the Index field is not checked. 0b - Disable command index check 1b - Enables command index check
19 CCCEN	Command CRC check enable If this field is set to 1, uSDHC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this field is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. See RSPTYP[1:0] and Command Transfer Type (CMD_XFR_TYP). 0b - Disables command CRC check 1b - Enables command CRC check
18 —	Reserved
17-16 RSPTYP	Response type select 00b - No response 01b - Response length 136 10b - Response length 48 11b - Response length 48, check busy after response
15-8 —	Reserved
7 AC23EN	AC23EN This field is read when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is tied to '0'. When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register. 0b - Disable 1b - Enable
6 NIBBLE_POS	NIBBLE_POS This field indicates the nibble position in the DDR 4-bit mode. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
5 MSBSEL	MSBSEL This field enables multiple block DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0b - Disable 1b - Enable
4 DTDSEL	DTDSEL This field defines the direction of DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. 0b - Disable 1b - Enable
3 DDR_EN	DDR_EN Dual data rate mode selection. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. 0b - Disable 1b - Enable
2 AC12EN	AC12EN Multiple block transfers for memory require a CMD12 to stop the transaction. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0b - Disable 1b - Enable
1 BCEN	BCEN This field is used to enable the Block Count register, which is only relevant for multiple block transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0b - Disable 1b - Enable
0	DMAEN

Table continues on the next page...

Table continued from the previous page...

Field	Function
DMAEN	This field enables DMA functionality. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0b - Disable 1b - Enable

40.6.1.6 Command Response0 (CMD_RSP0)

Offset

Register	Offset
CMD_RSP0	10h

Function

This register is used to store part 0 of the response bits from the card.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CMDRSP0	Command response 0 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

40.6.1.7 Command Response1 (CMD_RSP1)

Offset

Register	Offset
CMD_RSP1	14h

Function

This register is used to store part 1 of the response bits from the card.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CMDRSP1	Command response 1 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

40.6.1.8 Command Response2 (CMD_RSP2)

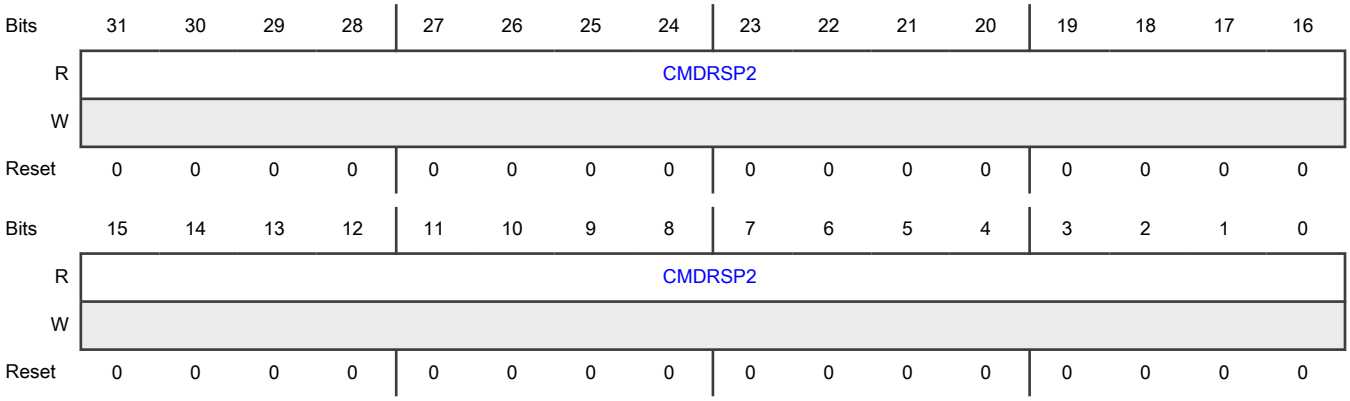
Offset

Register	Offset
CMD_RSP2	18h

Function

This register is used to store part 2 of the response bits from the card.

Diagram



Fields

Field	Function
31-0 CMDRSP2	Command response 2 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

40.6.1.9 Command Response3 (CMD_RSP3)

Offset

Register	Offset
CMD_RSP3	1Ch

Function

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD bus to Command Response registers for each response type. In this table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 317. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}

Table continues on the next page...

Table 317. Response bit definition for each response type (continued)

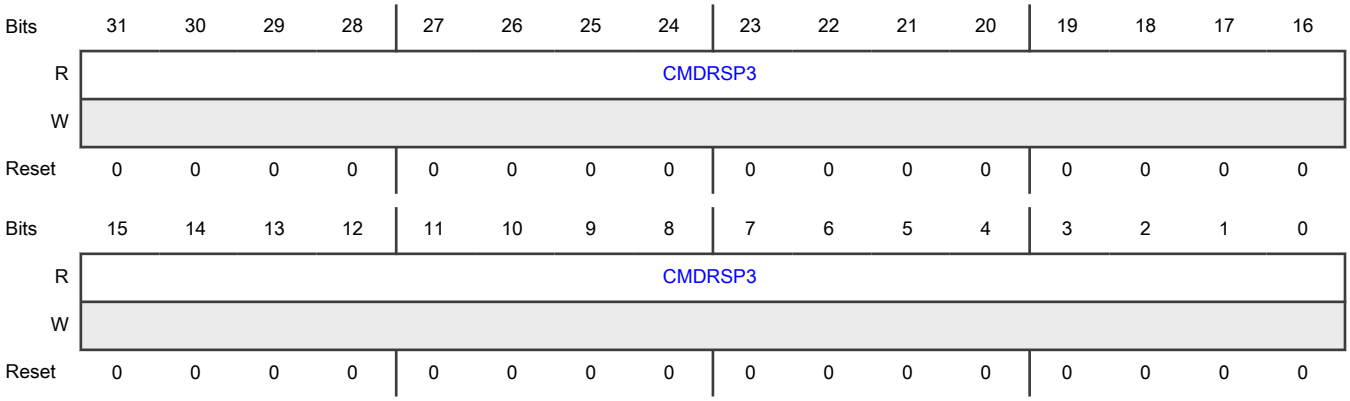
Response type	Meaning of response	Response field	Response register
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, uSDHC only stores part of the response data in the Command Response registers. This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, uSDHC checks R[47:1], and if the response length is 136 the uSDHC checks R[119:1].

Because uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows uSDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

Diagram



Fields

Field	Function
31-0 CMDRSP3	Command response 3 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

40.6.1.10 Data Buffer Access Port (DATA_BUFF_ACC_PORT)

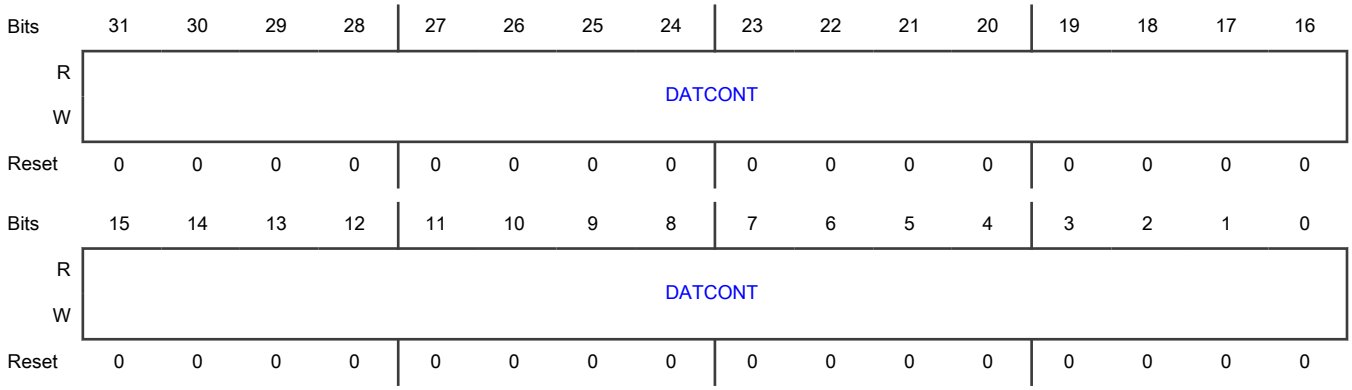
Offset

Register	Offset
DATA_BUFF_ACC_PORT	20h

Function

The Buffer Data Port register is for 32-bit data access by the Arm platform. When the internal DMA is enabled, any write to this field is ignored, and any read from this field always yields 0s.

Diagram



Fields

Field	Function
31-0 DATCONT	Data content This field is used to access the internal buffer.

40.6.1.11 Present State (PRES_STATE)

Offset

Register	Offset
PRES_STATE	24h

Function

The host driver can get status of uSDHC from this 32-bit read only register.

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands are issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

NOTE

The reset value of Present State register depends on board connectivity.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL								CLSL	Reserved			Reserved	WPSPL	CDPL	0	CINST
W																	
Reset	u	u	u	u	u	u	u	u	u	0	0	u	u	u	0	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSCD	0		RTR	BREN	BWEN	RTA	WTA	Reserved				SDSTB	DLA	CDIHB	CIHB
W																
Reset	1	0	0	0	0	0	0	0	1	0	0	0	u	0	0	0

Fields

Field	Function
31-24 DLSL	DATA[7:0] line signal level This field is used to check the DATA line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. 0000_0001b - Data 0 line signal level 0000_0010b - Data 1 line signal level 0000_0100b - Data 2 line signal level 0000_1000b - Data 3 line signal level 0001_0000b - Data 4 line signal level 0010_0000b - Data 5 line signal level 0100_0000b - Data 6 line signal level 1000_0000b - Data 7 line signal level
23 CLSL	CMD line signal level This field is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this field after reset is 1'b1, when the command line is pulled up.
22-21 —	Reserved
20	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
19 WPSPL	Write protect switch pin level The Write Protect switch is supported for memory and combo cards. This field reflects the inverted value of the WP pin of the card socket. A software reset does not affect this field. The reset value is affected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this field is high and write is enabled. 0b - Write protected (WP = 1) 1b - Write enabled (WP = 0)
18 CDPL	Card detect pin level This field reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this field. This field may be valid, but is not guaranteed, because of propagation delay. Use of this field is limited to testing because it must be debounced by software. A software reset does not affect this field. A write to the Force Event Register does not affect this field. The reset value is affected by the external card detection pin. This field shows the value on the CD_B pin (that is, when a card is inserted in the socket, it is 0 on the CD_B input, and consequently, the CDPL reads 1. 0b - No card present (CD_B = 1) 1b - Card present (CD_B = 0)
17 —	Reserved
16 CINST	Card inserted This field indicates whether a card has been inserted. The uSDHC module debounces this signal so that the host driver does not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not affect this field. The Software Reset For All in the System Control register does not affect this field. A software reset does not affect this field. 0b - Power on reset or no card 1b - Card inserted
15 TSCD	Tap select change done This field indicates the delay setting is effective after write CLK_TUNE_CTRL_STATUS register. 0b - Delay cell select change is not finished. 1b - Delay cell select change is finished.
14-13 —	Reserved
12	Re-Tuning Request (only for SD3.0 SDR104 mode, and eMMC HS200 mode)

Table continues on the next page...

Table continued from the previous page...

Field	Function
RTR	Host controller may request host driver to execute re-tuning sequence by setting this field when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This field is cleared when a command is issued with setting Execute Tuning field in MIX_CTRL register. Changing of this field from 0 to 1 generates Re-Tuning Event. See Interrupt status registers for more detail. This field isn't set to 1 if Sampling Clock Select in the MIX_CTRL register is set to 0 (using fixed sampling clock). 0b - Fixed or well tuned sampling clock 1b - Sampling clock needs re-tuning
11 BREN	Buffer read enable This status field is used for non-DMA read transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this field is high, valid data greater than the watermark level exist in the buffer. A change of this field from 1 to 0 occurs when some reads from the buffer (read DATPORT (Base + 0x20)) are made and the buffer hasn't valid data greater than the watermark level. A change of this field from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 0b - Read disable 1b - Read enable
10 BWEN	Buffer write enable This status field is used for non-DMA write transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this field is 1, valid data greater than the watermark level can be written to the buffer. A change of this field from 1 to 0 occurs when some writes to the buffer (write DATPORT (Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this field from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 0b - Write disable 1b - Write enable
9 RTA	Read transfer active This status field is used for detecting completion of a read transfer. This field is set for either of the following conditions: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer A transfer complete interrupt is generated when this field changes to 0. This field is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from uSDHC internal buffer.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent because of the Stop At Block Gap Request being set to 1. 0b - No valid data 1b - Transferring data
8 WTA	Write transfer active This status field indicates a write transfer is active. If this field is 0, it means no valid write data exists in uSDHC. This field is set in either of the following cases: • After the end field of the write command • When writing 1 to the Continue Request field in the Protocol Control register to restart a write transfer This field is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request During a write transaction, a Block Gap Event interrupt is generated when this field is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the host driver in determining when to issue commands during Write Busy state. 0b - No valid data 1b - Transferring data
7-4 —	Reserved
3 SDSTB	SD clock stable This status field indicates that the internal card clock is stable. This field is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON field in System Control register to remove glitches on the card clock when the frequency is changing. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. 0b - Clock is changing frequency and not stable. 1b - Clock is stable.
2 DLA	Data line active This status field indicates whether one of the DATA lines on the SD bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field is set in either of the following cases: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer This field is cleared in either of the following cases: • When the end field of the last data block is sent from the SD bus to uSDHC. • When the Read Wait state is stopped by a Suspend command and the DATA2 line is released. The uSDHC module waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait to use the suspend / resume function. This field remains 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the write command • When writing to 1 to the Continue Request field in the Protocol Control register to continue a write transfer This field is cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, uSDHC also detects if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, uSDHC assumes the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and because of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This field is cleared when the DATA0 line is released. 0b - DATA line inactive 1b - DATA line active
1 CDIHB	Command Inhibit Data (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / eMMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Can issue command that uses the DATA line 1b - Cannot issue command that uses the DATA line
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / eMMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0b - Can issue command using only CMD line 1b - Cannot issue command

40.6.1.12 Protocol Control (PROT_CTRL)

Offset

Register	Offset
PROT_CTRL	28h

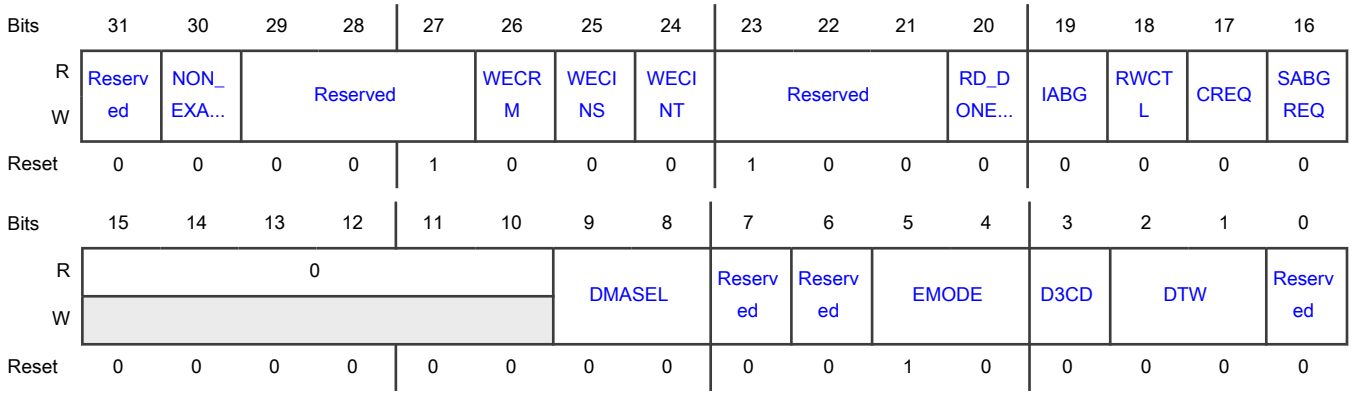
Function

This register controls three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether uSDHC issues a Suspend command or the SD card accepts the Suspend command.

- If the host driver does not issue a Suspend command, the Continue request is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card does not accept it, the Continue request is used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver waits for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the host driver clears the Stop At Block Gap Request before or simultaneously.

Diagram



Fields

Field	Function
31 —	Reserved Always write as 0
30 NON_EXACT_BLK_RD	Non-exact block read Current block read is non-exact block read. It is only used for SDIO. 0b - The block read is exact block read. Host driver does not need to issue abort command to terminate this multi-block read. 1b - The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read.
29-27 —	Reserved
26 WECRM	Wakeup event enable on SD card removal This field enables a wakeup event, via a card removal, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Removal Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Removal Status and uSDHC interrupt. 0b - Disables wakeup event enable on SD card removal 1b - Enables wakeup event enable on SD card removal
25 WECINS	Wakeup event enable on SD card insertion This field enables a wakeup event, via a card insertion, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Insertion Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Insertion Status and uSDHC interrupt. 0b - Disable wakeup event enable on SD card insertion 1b - Enable wakeup event enable on SD card insertion

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 WECINT	Wakeup event enable on card interrupt This field enables a wakeup event, via a card interrupt, in the Interrupt Status register. This field can be set to 1 if FN_WUS (Wakeup Support) in CIS is set to 1. When this field is set, the Card Interrupt Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Interrupt Status and uSDHC interrupt. 0b - Disables wakeup event enable on card interrupt 1b - Enables wakeup event enable on card interrupt
23-21 —	Reserved Always write as 3'b100
20 RD_DONE_NO_8CLK	Read performed number 8 clock According to the SD/eMMC spec, for read data transaction, 8 clocks are needed after the end field of the last data block. So, by default(RD_DONE_NO_8CLK=0), eight clocks are active after the end field of the last read data transaction. However, these 8 clocks should not be active if user wants to use stop at block gap (include the auto stop at block gap in boot mode) feature for read and the RWCTL field (bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid these 8 clocks. Otherwise, the device might send extra data to uSDHC while uSDHC ignores these data. In a summary, this field should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.
19 IABG	Interrupt at block gap This field is valid only in 4-bit mode, of SDIO, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If SDIO cannot signal an interrupt during a multiple block transfer, this field should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO insertion, it sets this field according to the CCCR of the card. 0b - Disables interrupt at block gap 1b - Enables interrupt at block gap
18 RWCTL	Read wait control The read wait function provided by this field is optional for SDIO. If the card supports read wait, set this field to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise, uSDHC has to stop the SD clock to hold read data, which restricts commands generation. When the host driver detects an SDIO insertion, it sets this field according to the CCCR of the card. If the card does not support read wait, this field should never be set to 1; otherwise, DATA line conflicts might occur. If this field is set to 0, stop at block gap during read operation is also supported, but uSDHC stops the SD clock to pause reading operation. 0b - Disables read wait control and stop SD clock at block gap when SABGREQ field is set 1b - Enables read wait control and assert read wait without stopping SD clock at block gap when SABGREQ field is set
17	Continue request

Table continues on the next page...

Table continued from the previous page...

Field	Function
CREQ	This field is used to restart a transaction which was stopped using the stop at block gap request. When a suspend operation is not accepted by the card, it is also by setting this field to restart the paused transfer. To cancel stop at the block gap, set stop at block gap request to 0 and set this field to 1 to restart the transfer. The uSDHC module automatically clears this field, therefore it is not necessary for the host driver to set this field to 0. If both stop at block gap request and this field are 1, the continue request is ignored. 0b - No effect 1b - Restart
16 SABGREQ	Stop at block gap request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver leaves this field set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC module supports the stop at block gap request for write transfers, but for read transfers it requires that SDIO support read wait. Therefore, the host driver does not set this field during read transfers unless SDIO supports Read Wait and has set the read wait control to 1; otherwise, uSDHC stops the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the Data Port register, the host driver sets this field after all block data is written. If this field is set to 1, the host driver does not write data to the Data Port register after a block is sent. Once this field is set, the host driver does not clear this field before the Transfer Complete field in Interrupt Status register is set, otherwise uSDHC's behavior is undefined. This field effects read transfer active, write transfer active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 0b - Transfer 1b - Stop
15-10 —	Reserved
9-8 DMASEL	DMA select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00b - No DMA or simple DMA is selected. 01b - ADMA1 is selected. 10b - ADMA2 is selected. 11b - Reserved
7 —	Reserved
6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
5-4 EMODE	Endian mode This field supports all three endian modes in data transfer. See Data buffer for more details. 00b - Big endian mode 01b - Half word big endian mode 10b - Little endian mode 11b - Reserved
3 D3CD	DATA3 as card detection pin If this field is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 might set the card into the SPI mode, which uSDHC does not support. 0b - DATA3 does not monitor card insertion 1b - DATA3 as card detection pin
2-1 DTW	Data transfer width This field selects the data width of the SD bus for a data transfer. The host driver sets it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00b - 1-bit mode 01b - 4-bit mode 10b - 8-bit mode 11b - Reserved
0 —	Reserved

40.6.1.13 System Control (SYS_CTRL)

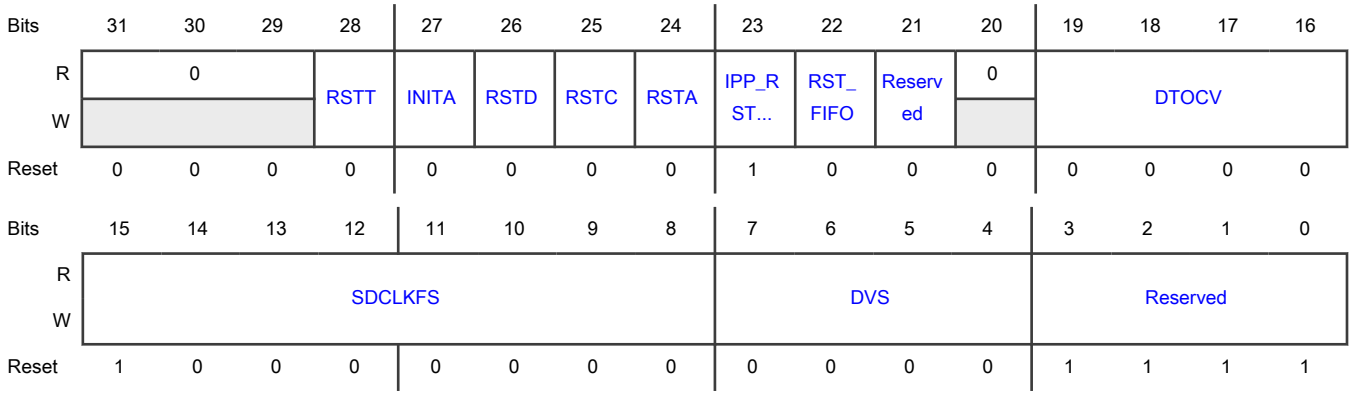
Offset

Register	Offset
SYS_CTRL	2Ch

Function

This register provides control of the system. See detail in the field description.

Diagram



Fields

Field	Function
31-29 —	Reserved
28 RSTT	Reset tuning When set this field to 1, it resets tuning circuit. After tuning circuits are reset, bit value is 0. Clearing execute_tuning field in AUTOCMD12_ERR_STATUS also sets this field to 1 to reset tuning circuit
27 INITA	Initialization active When this field is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this field is self cleared. This field is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this field is already 1 has no effect. Writing 0 to this field at any time has no effect. When either of the CIHB and CDIHB fields in the Present State register are set, writing 1 to this field is ignored (that is, when command line or data lines are active, write to this field is not allowed). On the other-hand, when this field is set, that is, during initialization active period, it is allowed to issue command, and the command bit stream appears on the CMD pad after all 80 clock cycles are done. So, when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs to send 80 cycles to the card and does not want to wait till this field is self cleared.
26 RSTD	Software reset for data line Only part of the data circuit is reset. DMA circuit is also reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Data Port register — Buffer is cleared and initialized • Present State register — Buffer read enable — Buffer write enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	— Read transfer active — Write transfer active — DATA line active — Command Inhibit (DATA) • Protocol Control register — Continue request • Interrupt Status register — Buffer read ready — Buffer write ready — DMA interrupt — Block gap event — Transfer complete **NOTE** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0b - No reset 1b - Reset
25 RSTC	Software reset for CMD line Only part of the command circuit is reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Present State Register — Command Inhibit (CMD) • Interrupt Status Register — Command Complete 0b - No reset 1b - Reset
24 RSTA	Software reset for all This reset affects the entire host controller except for the card detection circuit. RSTA resets all the registers that can be reset by RSTC/RSTD. During its initialization, the host driver is set this field to 1 to reset uSDHC. The uSDHC module resets this field to 0 when the capabilities registers are valid and the host driver can read them. Additional use of Software Reset For All does not affect the value of the capabilities registers. After this field is set, it is recommended that the host driver reset the external card and re-initialize it. After this field is set, the software should wait for self-clear. In tuning process, after every CMD19 is finished, this field is set to retest uSDHC.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0b - No reset 1b - Reset
23 IPP_RST_N	Hardware reset This field has alternative function as below: • eMMC: be used to reset eMMC, through pad directly to eMMC RST_n pin. • Card: be used to control SD Card power enable or disable during warm reset, through pad directly to hardware reset on board.
22 RST_FIFO	Reset the async FIFO Reset the async FIFO between card interface and the internal logic. After this field is set, the software waits for self-clear.
21 —	Reserved
20 —	Reserved
19-16 DTOCV	Data timeout counter value This value determines the interval by which DAT line timeouts are detected. See the Data Timeout Error field in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency is generated by dividing the base clock SDCLK value by this value. The host driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 0000b - SDCLK x 2^{32} 0001b - SDCLK x 2^{33} 0010b - SDCLK x 2^{18} 0011b - SDCLK x 2^{19} 1101b - SDCLK x 2^{29}, recommend to use for supported speed modes except HS200, HS400, SDR104 mode 1110b - SDCLK x 2^{30}, recommend to use for HS200 and SDR104 mode 1111b - SDCLK x 2^{31}, recommend to use for HS400 mode
15-8 SDCLKFS	SDCLK frequency select

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this field holds the prescaler (of this register) and divisor (next register) of the Base Clock Frequency register. In Single Data Rate mode (DDR_EN field of MIXERCTRL is '0') Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16 04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 While in Dual Data Rate mode (DDR_EN field of MIXERCTRL is '1') Only the following settings are allowed: 80h) Base clock divided by 512 40h) Base clock divided by 256 20h) Base clock divided by 128 10h) Base clock divided by 64 08h) Base clock divided by 32 04h) Base clock divided by 16 02h) Base clock divided by 8 01h) Base clock divided by 4 00h) Base clock divided by 2 When the software changes the DDR_EN field, SDCLKFS might need to be changed also. In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h yields 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The reset value of this field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD clock after reset is 375 kHz. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. If setting SDCLKFS and DVS can generate the same clock frequency,(for example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.), SDCLKFS is highly recommended. For HS400 mode, SDCLKFS must be used so that REF_CLK, generated for STROBE DLL, is at the same frequency as STROBE clock. For HS200 mode, if DLL is to be used, make sure SDCLKFS is used so that REF_CLK, generated for PARTS DLL, is at the same frequency as SD_CLK. For SDR104 mode, if DLL is to be used, make sure SDCLKFS is used so that REF_CLK, generated for PARTS DLL, is at the same frequency as SD_CLK.
7-4 DVS	Divisor This field is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisors without deterioration of duty cycle. Before changing clock divisor value (SDCLKFS or DVS), Host driver should make sure the SDSTB field is high. The settings are as follows: 0000b - Divide-by-1 0001b - Divide-by-2 1110b - Divide-by-15 1111b - Divide-by-16
3-0 —	Reserved Always write as 1.

40.6.1.14 Interrupt Status (INT_STATUS)

Offset

Register	Offset
INT_STATUS	30h

Function

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status fields is set to 1. For all fields, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition; otherwise, the CINT field is asserted again.

The table below shows the relationship between the command timeout error and the command complete.

Table 318. uSDHC status for command timeout error/command complete bit combinations

Command complete	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the transfer complete and the data timeout error.

Table 319. uSDHC status for data timeout error/transfer complete bit combinations

Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC error and command timeout error.

Table 320. uSDHC status for command CRC error/command timeout error bit combinations

Command CRC error	Command timeout error	Meaning of the status
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAE	0	TNE	0	AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W		W1C		W1C		W1C		W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_INT...	CQI	TP	RTE	0	Reserved		CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		W1C	W1C	W1C		W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-29 —	Reserved
28 DMAE	DMA error Occurs when an Internal DMA transfer has failed. This field is set to 1, when some error occurs in the data transfer. This error can be caused by either simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Because any error corrupts the whole data block, the host driver restarts the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 0b - No error 1b - Error
27 —	Reserved
26 TNE	Tuning error: (only for SD3.0 SDR104 mode and eMMC HS200 mode) This field is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, host driver needs to abort a command executing and perform tuning. This field is relevant in combination with the bus error only and must be cleared prior to all the write and read transactions.
25 —	Reserved
24 AC12E	Auto CMD12 error Occurs when detecting that one of the fields in the Auto CMD12 Error Status register has changed from 0 to 1. This field is set to 1, not only when the errors in Auto CMD12 occur, but also, when the Auto CMD12 is not executed due to the previous command error. 0b - No error 1b - Error
23 —	Reserved
22 DEBE	Data end bit error Occurs either when detecting 0 at the end field position of read data that uses the DATA line, or at the end field position of the CRC. This field is not asserted in tuning process.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No error 1b - Error
21 DCE	Data CRC error Occurs when detecting a CRC error when transferring read data that uses the DATA line, or when detecting the Write CRC status having a value other than 010. This field is not asserted in tuning process. 0b - No error 1b - Error
20 DTOE	Data timeout error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out. This field is not asserted in tuning process. 0b - No error 1b - Time out
19 CIE	Command index error Occurs if a command index error occurs in the command response. This field is not asserted in tuning process. 0b - No error 1b - Error
18 CEBE	Command end bit error Occurs when detecting that the end field of a command response is 0. This field is not asserted in tuning process. 0b - No error 1b - End bit error generated
17 CCE	Command CRC error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this field is set when detecting a CRC error in the command response. • The uSDHC module detects a CMD line conflict by monitoring the CMD line when a command is issued. If uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then uSDHC stops the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error should also be set to 1 to distinguish CMD line conflict.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field is not asserted in tuning process. 0b - No error 1b - CRC error generated
16 CTOE	Command timeout error Occurs only if no response is returned within 64 SDCLK cycles from the end field of the command. If uSDHC detects a CMD line conflict, in which case a Command CRC Error is also be set (as shown in Interrupt Status (INT_STATUS)), this field is set without waiting for 64 SDCLK cycles. This is because the command is stopped by uSDHC. This field is not asserted in tuning process. 0b - No error 1b - Time out
15 ERR_INT_STATUS	Error Interrupt Status This bit is set when any of the error status bit DMAE, AC12E, DEBE, DCE, DTOE, CIE, CEBE, CCE and CTOE is set.
14 CQI	Command queuing interrupt This interrupt is asserted when at least one of the bits in CQIS register is set. This interrupt is cleared only by clearing the source interrupt in CQIS register.
13 TP	Tuning pass:(only for SD3.0 SDR104 mode and eMMC HS200 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.
12 RTE	Re-tuning event: (only for SD3.0 SDR104 mode and eMMC HS200 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. host controller requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0b - Re-tuning is not required. 1b - Re-tuning should be performed.
11 —	Reserved
10-9 —	Reserved
8 CINT	Card interrupt This status field is set when an interrupt signal is detected from the external card. In 1-bit mode, uSDHC detects the Card Interrupt without the SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from SDIO and the interrupt to the host system. Writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	this field to 1 can clear this field, but as the interrupt source from SDIO does not clear, this field is set again. to clear this field, it is required to reset the interrupt source from the external card followed by a writing 1 to this field. When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset the interrupt sources in SDIO and the interrupt signal might not be asserted), write 1 to clear this field, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0b - No card interrupt 1b - Generate card interrupt
7 CRM	Card removal This status field is set if the Card Inserted field in the Present State register changes from 1 to 0. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if no card is inserted. to leave it cleared, clear the Card Removal Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or inserted 1b - Card removed
6 CINS	Card insertion This status field is set if the Card Inserted field in the Present State register changes from 0 to 1. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if a card is inserted. to leave it cleared, clear the Card Inserted Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or removed 1b - Card inserted
5 BRR	Buffer read ready This status field is set if the Buffer Read Enable field, in the Present State register, changes from 0 to 1. See the Buffer Read Enable field in the Present State register for additional information. This field indicates that cmd19 is finished in tuning process. 0b - Not ready to read buffer 1b - Ready to read buffer
4 BWR	Buffer write ready This status field is set if the Buffer Write Enable field, in the Present State register, changes from 0 to 1. See the Buffer Write Enable field in the Present State register for additional information. 0b - Not ready to write buffer 1b - Ready to write buffer

Table continues on the next page...

Table continued from the previous page...

Field	Function
3 DINT	DMA interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this field does not be set. Instead, the DMAE field is set. Either Simple DMA or ADMA finishes data transferring, this field is set. 0b - No DMA interrupt 1b - DMA interrupt is generated.
2 BGE	Block gap event If the Stop At Block Gap Request field in the Protocol Control register is set, this field is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this field is not set to 1. In the case of a Read Transaction: This field is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD bus timing). The Read Wait must be supported to use this function. In the case of Write Transaction: This field is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 0b - No block gap event 1b - Transaction stopped at block gap
1 TC	Transfer complete This field is set when a read or write transfer is completed. In the case of a Read Transaction: This field is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the host system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request field in the Protocol Control register (after valid data has been read to the host system). In the case of a Write Transaction: This field is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request field in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this field is set when busy is deasserted. This field is not asserted in tuning process. 0b - Transfer does not complete 1b - Transfer complete
0 CC	Command complete This field is set when you receive the end field of the command response (except auto CMD12). See the Command Inhibit (CMD) in the Present State register. This field is not asserted in tuning process. 0b - Command not complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Command complete

40.6.1.15 Interrupt Status Enable (INT_STATUS_EN)

Offset

Register	Offset
INT_STATUS_EN	34h

Function

Setting any bit in this register to 1 enables the corresponding interrupt status be requested to the system. If any bit is set to 0, the corresponding interrupt request gets blocked.

- Depending on IABG field setting, uSDHC might be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There are some delays on the Card Interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAESEN	0	TNESEN	0	AC12ESEN	0	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CQISEN	TPSEN	RTESEN	0	Reserved		CINTSEN	CRMSEN	CINSSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-29 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
28 DMAESEN	DMA error status enable 0b - Masked 1b - Enabled
27 —	Reserved
26 TNESEN	Tuning error status enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12ESEN	Auto CMD12 error status enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBESEN	Data end bit error status enable 0b - Masked 1b - Enabled
21 DCESEN	Data CRC error status enable 0b - Masked 1b - Enabled
20 DTOESEN	Data timeout error status enable 0b - Masked 1b - Enabled
19 CIESEN	Command index error status enable 0b - Masked 1b - Enabled
18 CEBESEN	Command end bit error status enable 0b - Masked 1b - Enabled
17	Command CRC error status enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
CCESEN	0b - Masked 1b - Enabled
16 CTOESEN	Command timeout error status enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQISEN	Command queuing status enable 0b - Masked 1b - Enabled
13 TPSEN	Tuning pass status enable 0b - Masked 1b - Enabled
12 RTESEN	Re-tuning event status enable 0b - Masked 1b - Enabled
11 —	Reserved
10-9 —	Reserved
8 CINTSEN	Card interrupt status enable If this field is set to 0, uSDHC clears the interrupt request to the system. The Card Interrupt detection is stopped when this field is cleared and restarted when this field is set to 1. The host driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this field again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0b - Masked 1b - Enabled
7 CRMSEN	Card removal status enable 0b - Masked 1b - Enabled
6 CINSSEN	Card insertion status enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Masked 1b - Enabled
5 BRRSEN	Buffer read ready status enable 0b - Masked 1b - Enabled
4 BWRSEN	Buffer write ready status enable 0b - Masked 1b - Enabled
3 DINTSEN	DMA interrupt status enable 0b - Masked 1b - Enabled
2 BGESEN	Block gap event status enable 0b - Masked 1b - Enabled
1 TCSEN	Transfer complete status enable 0b - Masked 1b - Enabled
0 CCSEN	Command complete status enable 0b - Masked 1b - Enabled

40.6.1.16 Interrupt Signal Enable (INT_SIGNAL_EN)

Offset

Register	Offset
INT_SIGNAL_EN	38h

Function

This register is used to select which interrupt status is indicated to the host system as the interrupt. These status fields all share the same interrupt lines. Setting any of these fields to 1 enables interrupt generation. The corresponding Status register field generates an interrupt when the corresponding interrupt signal enable field is set.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAEIEN	0	TNEIEN	0	AC12EIEN	0	DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CQIIEN	TPIEN	RTEIEN	0	Reserved		CINTIEN	CRMIEN	CINSIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-29 —	Reserved
28 DMAEIEN	DMA error interrupt enable 0b - Masked 1b - Enable
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12EIEN	Auto CMD12 error interrupt enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable 0b - Masked

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enabled
21 DCEIEN	Data CRC error interrupt enable 0b - Masked 1b - Enabled
20 DTOEIEN	Data timeout error interrupt enable 0b - Masked 1b - Enabled
19 CIEIEN	Command index error interrupt enable 0b - Masked 1b - Enabled
18 CEBEIEN	Command end bit error interrupt enable 0b - Masked 1b - Enabled
17 CCEIEN	Command CRC error interrupt enable 0b - Masked 1b - Enabled
16 CTOEIEN	Command timeout error interrupt enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQIIEN	Command queuing signal enable 0b - Masked 1b - Enabled
13 TPIEN	Tuning pass interrupt enable 0b - Masked 1b - Enabled
12 RTEIEN	Re-tuning event interrupt enable 0b - Masked 1b - Enabled
11	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
10-9 —	Reserved
8 CINTIEN	Card interrupt enable 0b - Masked 1b - Enabled
7 CRMIEN	Card removal interrupt enable 0b - Masked 1b - Enabled
6 CINSIEN	Card insertion interrupt enable 0b - Masked 1b - Enabled
5 BRRIEN	Buffer read ready interrupt enable 0b - Masked 1b - Enabled
4 BWRIEN	Buffer write ready interrupt enable 0b - Masked 1b - Enabled
3 DINTIEN	DMA interrupt enable 0b - Masked 1b - Enabled
2 BGEIEN	Block gap event interrupt enable 0b - Masked 1b - Enabled
1 TCIEN	Transfer complete interrupt enable 0b - Masked 1b - Enabled
0 CCIEN	Command complete interrupt enable 0b - Masked 1b - Enabled

40.6.1.17 Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)

Offset

Register	Offset
AUTOCMD12_ERR_STATUS	3Ch

Function

When the Auto CMD12 Error Status field in the Status register is set, the host driver checks this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in field 04-01. This register is valid only when the Auto CMD12 Error status field is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 321. Relationship between command CRC error and command timeout error for auto CMD12

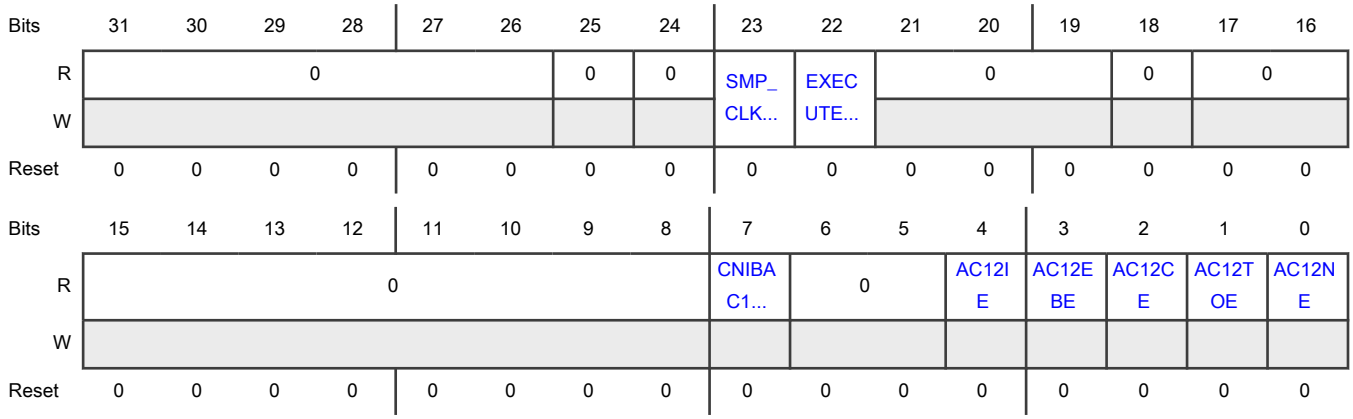
Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

- When uSDHC is going to issue an Auto CMD12
  - Set field 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
  - Set field 0 to 0 if the Auto CMD12 is issued
- At the end field of an Auto CMD12 response
  - Check errors correspond to fields 1-4
  - Set fields 1-4 corresponding to detected errors
  - Clear fields 1-4 corresponding to detected errors
- Before reading the Auto CMD12 Error Status field 7
  - Set field 7 to 1 if there is a command that can't be issued
  - Clear field 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, field 7 is sampled when the driver is not writing to the Command register. So, it is suggested to read this register only when the AC12E field in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error fields (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 —	Reserved
24 —	Reserved
23 SMP_CLK_SEL	Sample clock select This field is used to select sampling clock to receive CMD and DATA. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is controlled by re-tuning procedure and is valid after the tuning procedure is complete (AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=0). A value of '1' means that tuning has been completed successfully and a value of '0' implies the tuning has failed. Writing '1' to this field is meaningless and ignored when the tuning has failed. The tuning circuit is reset by writing '0' to this field. This field can be cleared by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=1. Once the tuning circuit is reset, it takes time to complete tuning sequence. Therefore, host driver should keep this field as 1 to perform the re-tuning sequence in a short time. This field cannot be changed while the Host controller is receiving response or a read data block. 0b - Fixed clock is used to sample data 1b - Tuned clock is used to sample data
22 EXECUTE_TUNING	Execute tuning This field is used to start the tuning procedure. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is set to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]. 0b - Tuning procedure is stopped

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Start tuning procedure
21-19 —	Reserved
18 —	Reserved
17-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error Setting this field to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register. 0b - No error 1b - Not issued
6-5 —	Reserved
4 AC12IE	Auto CMD12 / 23 index error Occurs if the command index error occurs in response to a command. 0b - No error 1b - Error, the CMD index in response is not CMD12/23
3 AC12EBE	Auto CMD12 / 23 end bit error Occurs when detecting that the end field of command response is 0 which should be 1. 0b - No error 1b - End bit error generated
2 AC12CE	Auto CMD12 / 23 CRC error Occurs when detecting a CRC error in the command response. 0b - No CRC error 1b - CRC error met in Auto CMD12/23 response
1 AC12TOE	Auto CMD12 / 23 timeout error Occurs if no response is returned within 64 SDCLK cycles from the end field of the command. If this field is set to1, the other error status fields (2-4) have no meaning. 0b - No error 1b - Time out

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 AC12NE	Auto CMD12 not executed If memory multiple block data transfer is not started, due to a command error, this field is not set because it is not necessary to issue an Auto CMD12. Setting this field to 1 means uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this field is set to 1, other error status fields (1-4) have no meaning. 0b - Executed 1b - Not executed

40.6.1.18 Host Controller Capabilities (HOST_CTRL_CAP)

Offset

Register	Offset
HOST_CTRL_CAP	40h

Function

This register provides the host driver with information specific to uSDHC implementation. The value in this register is the power-on-reset value and does not change with a software reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0				VS18	VS30	VS33	SRS	DMAS	HSS	ADMAS	0	MBL		
W																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		USE_TUN...	Reserved	Reserved				Reserved					DDR50_S...	SDR104_...	SDR50_S...
W																
Reset	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	1

Fields

Field	Function
31 —	Reserved
30-27	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
26 VS18	Voltage support 1.8 V This field depends on the host system ability. 0b - 1.8 V not supported 1b - 1.8 V supported
25 VS30	Voltage support 3.0 V This field depends on the host system ability. 0b - 3.0 V not supported 1b - 3.0 V supported
24 VS33	Voltage support 3.3 V This field depends on the host system ability. 0b - 3.3 V not supported 1b - 3.3 V supported
23 SRS	Suspend / resume support This field indicates whether uSDHC supports Suspend / Resume functionality. If this field is 0, the Suspend and Resume mechanism, as well as the read wait, are not supported, and the host driver does not issue either Suspend or Resume commands. 0b - Not supported 1b - Supported
22 DMAS	DMA support This field indicates whether uSDHC can use the internal DMA to transfer data between system memory and the data buffer directly. 0b - DMA not supported 1b - DMA supported
21 HSS	High speed support This field indicates whether uSDHC supports High Speed mode and the host system can supply a SD clock frequency from 25 MHz to 50 MHz. 0b - High speed not supported 1b - High speed supported
20 ADMAS	ADMA support This field indicates whether uSDHC supports the ADMA feature. 0b - Advanced DMA not supported

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Advanced DMA supported
19 —	Reserved
18-16 MBL	Max block length This field indicates the maximum block size that the host driver can read and write to the buffer in uSDHC. The buffer transfers block size without wait cycles. 000b - 512 bytes 001b - 1024 bytes 010b - 2048 bytes 011b - 4096 bytes
15-14 —	Reserved
13 USE_TUNING_SDR50	Use Tuning for SDR50 This field is used to enable tuning for SDR50 mode 0b - SDR50 does not support tuning 1b - SDR50 supports tuning
12 —	Reserved
11-8 —	Reserved
7-3 —	Reserved
2 DDR50_SUPPORT	DDR50 support This field indicates support of DDR50 mode.
1 SDR104_SUPPORT	SDR104 support This field indicates support of SDR104 mode.
0 SDR50_SUPPORT	SDR50 support This field indicates support of SDR50 mode.

40.6.1.19 Watermark Level (WTMK_LVL)

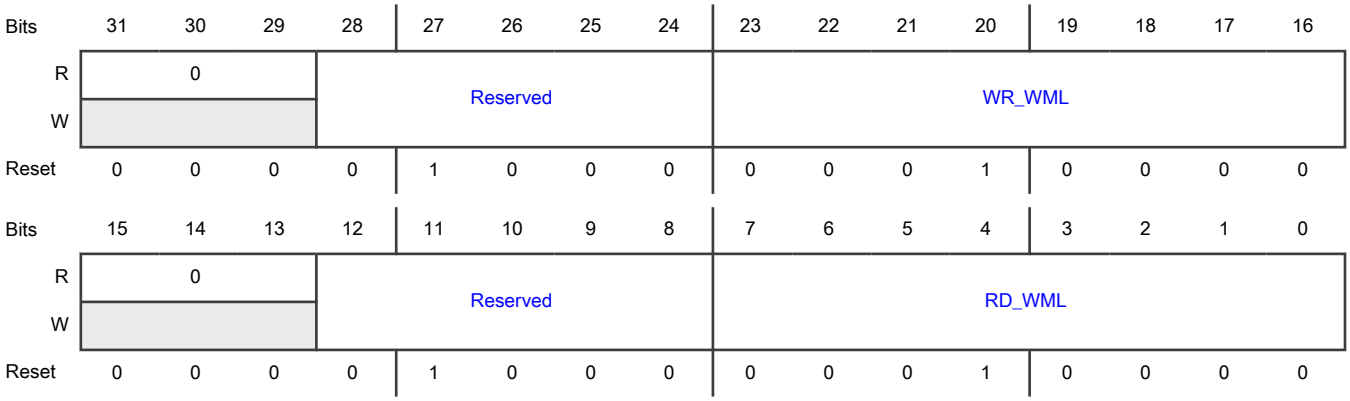
Offset

Register	Offset
WTMK_LVL	44h

Function

This register indicates configurability of write and read watermark levels (FIFO threshold). Their value can range from 1 to 128 words.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-24 —	Reserved
23-16 WR_WML	Write watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also, the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.When using internal DMA, WR_WML can be set any value between 1 and 128. To get max AXI throughput, we recommend to use integer multiple of 16.
15-13 —	Reserved
12-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 RD_WML	Read watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also, the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.When using internal DMA mode, RD_WML must be set to the value less than 16. To get max AXI throughput, recommend to use 16.

40.6.1.20 Mixer Control (MIX_CTRL)

Offset

Register	Offset
MIX_CTRL	48h

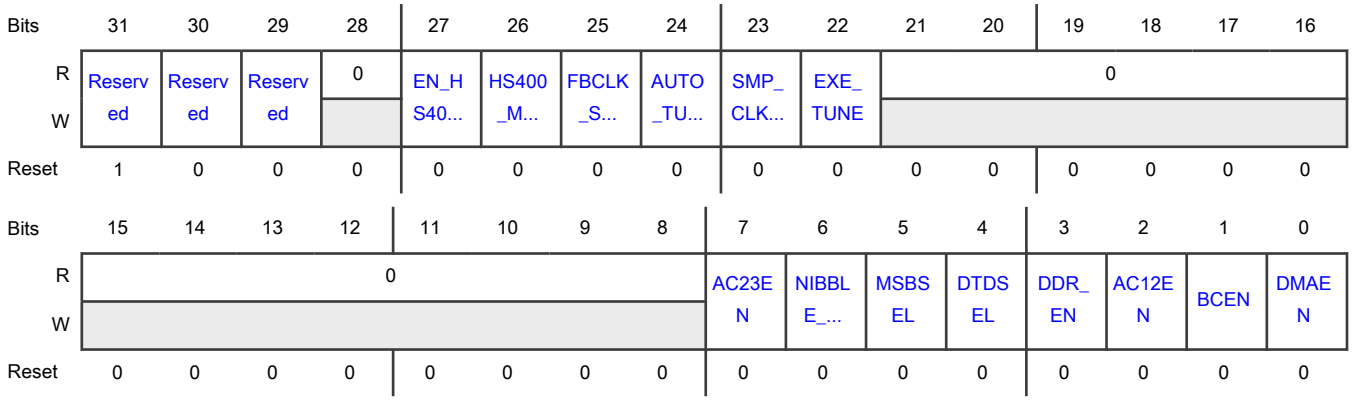
Function

This register is used to DMA and data transfer. To prevent data loss, the software should check if data transfer is active before writing this register. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

Table 322. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

Diagram



Fields

Field	Function
31 —	Reserved Always write as 1.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 0.
28 —	Reserved
27 EN_HS400_MODE	Enable enhance HS400 mode Enhance HS400 enable
26 HS400_MODE	Enable HS400 mode HS400 Enable
25 FBCLK_SEL	Feedback clock source selection (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) 0b - Feedback clock comes from the loopback CLK 1b - Feedback clock comes from the ipp_card_clk_out
24 AUTO_TUNE_EN	Auto tuning enable (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) 0b - Disable auto tuning 1b - Enable auto tuning
23 SMP_CLK_SEL	Clock selection When TUNING_CTRL[STD_TUNING_EN] is 0, this field is used to select Tuned clock or Fixed clock to sample data / cmd (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) 0b - Fixed clock is used to sample data / cmd 1b - Tuned clock is used to sample data / cmd
22 EXE_TUNE	Execute tuning: (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) When TUNING_CTRL[STD_TUNING_EN] is 0, this field is set to 1 to indicate the host driver is starting tuning procedure. Tuning procedure is stopped by writing 0. 0b - Not tuned or tuning completed 1b - Execute tuning
21-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 AC23EN	Auto CMD23 enable This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	Nibble position indication This field indicates the nibble position in the DDR 4-bit mode. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single block select This field enables multiple block DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0b - Single block 1b - Multiple blocks
4 DTDSEL	Data transfer direction select This field defines the direction of DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. 0b - Write (Host to card) 1b - Read (Card to host)
3 DDR_EN	Dual data rate mode selection This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only.
2 AC12EN	Auto CMD12 enable Multiple block transfers for memory require a CMD12 to stop the transaction. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0b - Disable 1b - Enable
1 BCEN	Block count enable This field is used to enable the Block Count register, which is only relevant for multiple block transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
0 DMAEN	DMA enable This field enables DMA functionality. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0b - Disable 1b - Enable

40.6.1.21 Force Event (FORCE_EVENT)

Offset

Register	Offset
FORCE_EVENT	50h

Function

This register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding field of the Interrupt Status Enable Register is set. Writing 1 to this register sets the corresponding field of Interrupt Status register, while writing 0 to it has no effect. A read from this register always results in 0's. To change the corresponding status fields in the Interrupt Status register, make sure to set IPGEN field in System Control Register so that peripheral clock is always active.

Forcing a card interrupt generates a short pulse on the DATA1 line, and the driver might treat this interrupt as a normal interrupt. The interrupt service routine might skip polling the card interrupt factor as the interrupt is self cleared.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
W	FEVTCINT			FEVTDMAE		FEVTTNE		FEVTAC1...		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCTOE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0		0	0	0	0	0
W									FEVTCNI...			FEVTAC1...	FEVTAC1...	FEVTAC1...	FEVTAC1...	FEVTAC1...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 FEVTCINT	Force event card interrupt Writing 1 to this field generates a short low-level pulse on the internal DATA1 line, as if a self-clearing interrupt was received from the external card. If enabled, the CINT field is set and the interrupt service routine might treat this interrupt as a normal interrupt from the external card.
30-29 —	Reserved
28 FEVTDMAE	Force event DMA error Forces the DMAE field of Interrupt Status register to be set
27 —	Reserved
26 FEVTTNE	Force tuning error Forces the TNE field of Interrupt Status register to be set
25 —	Reserved
24 FEVTAC12E	Force event Auto Command 12 error Forces the AC12E field of Interrupt Status register to be set
23 —	Reserved
22 FEVTDEBE	Force event data end bit error Forces the DEBE field of Interrupt Status register to be set
21 FEVTDCE	Force event data CRC error Forces the DCE field of Interrupt Status register to be set
20 FEVTDTOE	Force event data time out error Force the DTOE field of Interrupt Status register to be set
19 FEVTCIE	Force event command index error Forces the CCE field of Interrupt Status register to be set
18 FEVTCEBE	Force event command end bit error Forces the CEBE field of Interrupt Status register to be set
17 FEVTCCE	Force event command CRC error Forces the CCE field of Interrupt Status register to be set

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 FEVTCTOE	Force event command time out error Forces the CTOE field of Interrupt Status register to be set
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto Command 12 error Forces the CNIBAC12E field in the Auto Command12 Error Status register to be set
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto Command 12 index error Forces the AC12IE field in the Auto Command12 Error Status register to be set
3 FEVTAC12EBE	Force event Auto Command 12 end bit error Forces the AC12EBE field in the Auto Command12 Error Status register to be set
2 FEVTAC12CE	Force event auto command 12 CRC error Forces the AC12CE field in the Auto Command12 Error Status register to be set
1 FEVTAC12TOE	Force event auto command 12 time out error Forces the AC12TOE field in the Auto Command12 Error Status register to be set
0 FEVTAC12NE	Force event auto command 12 not executed Forces the AC12NE field in the Auto Command12 Error Status register to be set

40.6.1.22 ADMA Error Status (ADMA_ERR_STATUS)

Offset

Register	Offset
ADMA_ERR_STATUS	54h

Function

When an ADMA Error Interrupt has occurred, the ADMA error sates field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.

- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

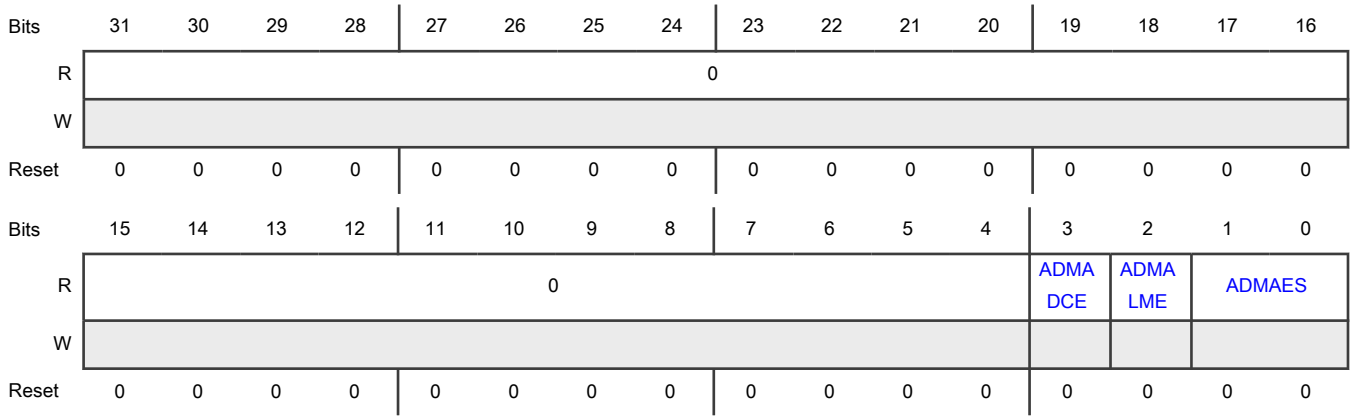
In case of a write operation, the host driver should use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data might exist in the host controller.

The host controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The host driver can distinguish this error by reading the Valid field of the error descriptor.

Table 323. ADMA error state coding

D01-D00	ADMA error state (when error has occurred)	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (Fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (Change address)	No ADMA error is generated
11	ST_TFR (Transfer data)	Holds the address of the next executable descriptor command

Diagram



Fields

Field	Function
31-4 —	Reserved
3 ADMADCE	ADMA descriptor error This error occurs when invalid descriptor fetched by ADMA. 0b - No error 1b - Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 ADMALME	ADMA length mismatch error This error occurs in the following two cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length cannot be divided by the block length. 0b - No error 1b - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred) This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. See ADMA Error Status (ADMA_ERR_STATUS) for more details.

40.6.1.23 ADMA System Address (ADMA_SYS_ADDR)

Offset

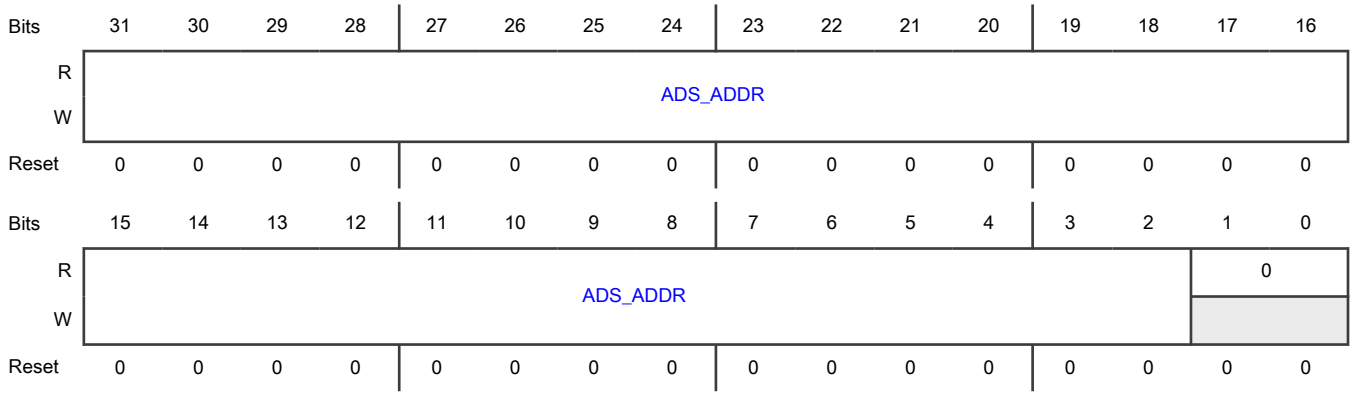
Register	Offset
ADMA_SYS_ADDR	58h

Function

This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the host driver sets the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned.

Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions.

Diagram



Fields

Field	Function
31-2 ADS_ADDR	ADMA system address This field contains the physical system memory address used for ADMA transfers.
1-0 —	Reserved

40.6.1.24 DLL (Delay Line) Control (DLL_CTRL)

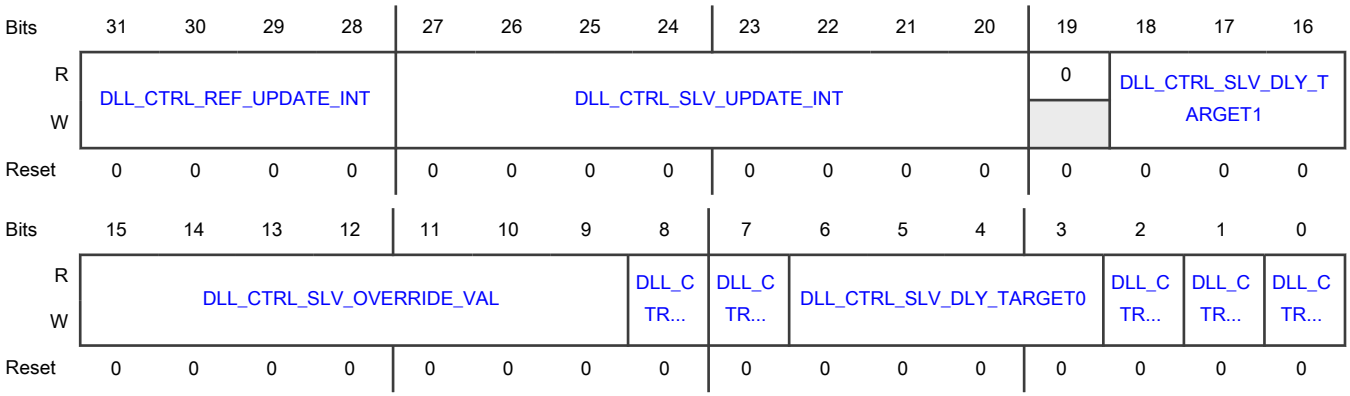
Offset

Register	Offset
DLL_CTRL	60h

Function

This register contains control fields for DLL.

Diagram



Fields

Field	Function
31-28 DLL_CTRL_REF_UPDATE_INT	DLL control loop update interval The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop updates every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20	Target delay line update interval

Table continues on the next page...

Table continued from the previous page...

Field	Function
DLL_CTRL_SLV_UPDATE_INT	If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the target-delay line using the SLV_FORCE_UPDATE register. Note that the target delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19 —	Reserved
18-16 DLL_CTRL_SLV_DLY_TARGET1	DLL target delay target1 See DLL_CTRL_SLV_DLY_TARGET0 below.
15-9 DLL_CTRL_SLV_OVERRIDE_VAL	DLL target override val When SLV_OVERRIDE = 1, this field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_SLV_OVERRIDE	DLL target override Set this field to 1 to Enable manual override for target delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_GATE_UPDATE	DLL gate update Set this field to 1 to prevent the DLL from updating (because when clock_in exists, glitches might appear during DLL updates). This field might be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6-3 DLL_CTRL_SLV_DLY_TARGET0	DLL target delay target0 The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an REF_CLOCK half-period. The delay is (DLL_CTRL_SLV_DLY_TARGET + 1)* REF_CLOCK_PERIOD[1] / 2) / 16, where DLL_CTRL_SLV_DLY_TARGET is a 7-bit value obtained by concatenating {DLL_CTRL_SLV_DLY_TARGET1, DLL_CTRL_SLV_DLY_TARGET0}. Therefore, the input read-clock can be delayed relative to the input data by (REF_CLOCK_PERIOD / 2) / 16 up to REF_CLOCK_PERIOD * 4. **NOTE** For the restrictions of delay cell implementation, the delay target must be set between REF_CLOCK_PERIOD/16 and REF_CLOCK_PERIOD*2 when REF_CLOCK is running at 200 MHz. When REF_CLOCK frequency is slower than 100 MHz, the maximum delay target might not reach REF_CLOCK_PERIOD*2.
2 DLL_CTRL_SLV_FORCE_UPD	DLL target delay line Setting this field to 1, forces the target delay line to update to the DLL calibrated value immediately. The target delay line updates automatically based on the SLV_UPDATE_INT interval or when a DLL lock

Table continues on the next page...

Table continued from the previous page...

Field	Function
	condition is sensed. Subsequent forcing of the target-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 DLL_CTRL_RESET	DLL reset Setting this field to 1 force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half-period phase shift. This signal is used by the DLL as edge-sensitive, so to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ENABLE	DLL and delay chain Set this field to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the target delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

1. REF_CLOCK_PERIOD is 1/REF_CLOCK.

40.6.1.25 DLL Status (DLL_STATUS)

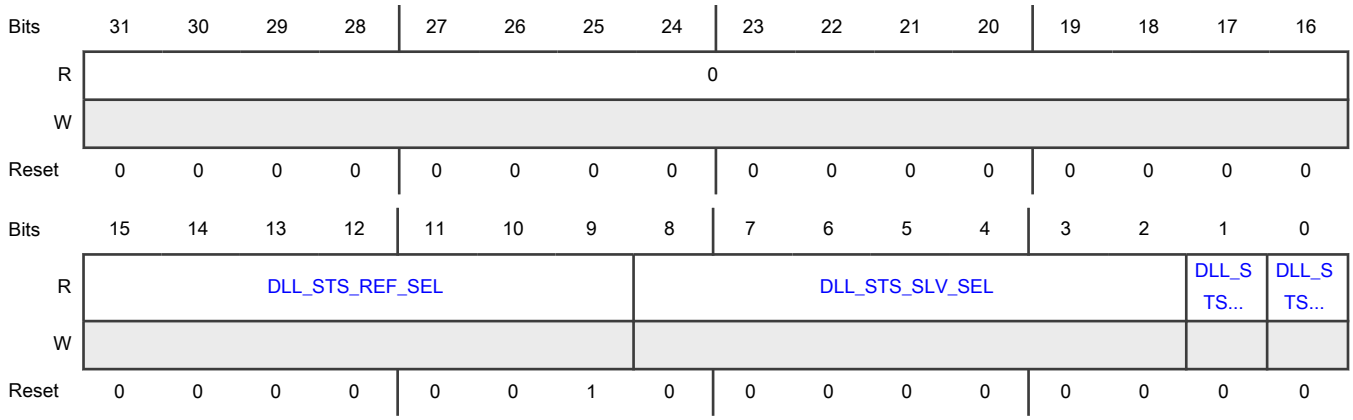
Offset

Register	Offset
DLL_STATUS	64h

Function

This register contains the DLL status information. All fields are read only and reads the same as the power-reset value.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-9 DLL_STS_REF_SEL	Reference delay line select taps This is encoded by 7 fields for 127 taps.
8-2 DLL_STS_SLV_SEL	Target delay line select status This is the instant value generated from reference chain. Because the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status This signifies that the DLL has detected and locked to an REF_CLOCK half-period phase shift , allowing the target delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Target delay-line lock status This signifies that a valid calibration has been set to the target-delay line and that the target-delay line is implementing the programmed delay value

40.6.1.26 CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)

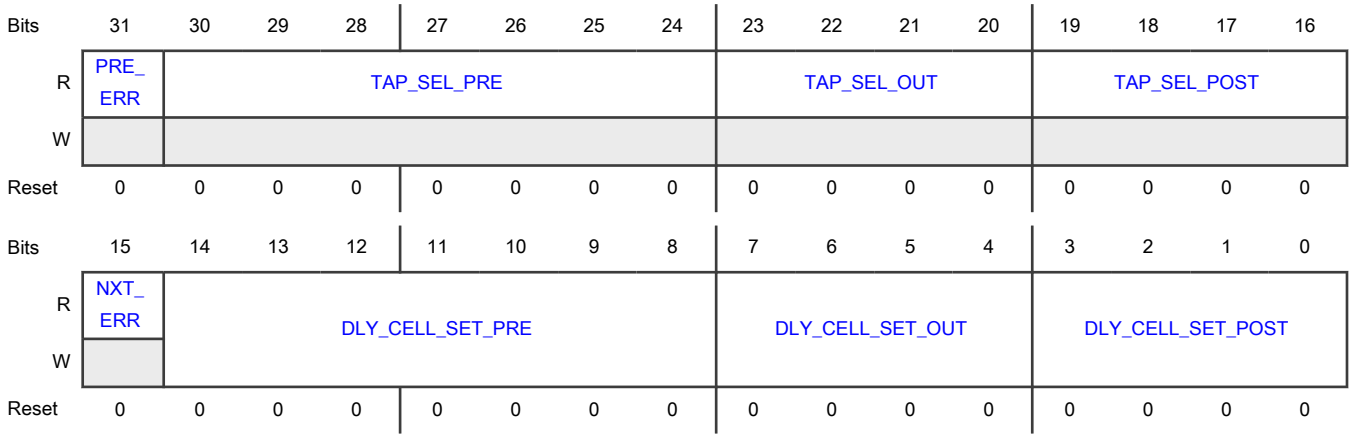
Offset

Register	Offset
CLK_TUNE_CTRL_STATUS	68h

Function

This register contains the Clock Tuning Control status information. This register is added to support SD3.0 UHS-I SDR104 mode and eMMC HS200 mode.

Diagram



Fields

Field	Function
31 PRE_ERR	PRE error PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
30-24 TAP_SEL_PRE	TAP_SEL_PRE Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE is updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23-20 TAP_SEL_OUT	Delay cells added on the feedback clock between CLK_PRE and CLK_OUT Reflects the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19-16 TAP_SEL_POST	Delay cells added on the feedback clock between CLK_OUT and CLK_POST Reflects the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
14-8 DLY_CELL_SET_PRE	delay cells on the feedback clock between the feedback clock and CLK_PRE Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7-4 DLY_CELL_SET_OUT	Delay cells on the feedback clock between CLK_PRE and CLK_OUT Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.

Table continues on the next page...

Table continued from the previous page...

Field	Function
3-0 DLY_CELL_SET_POST	Delay cells on the feedback clock between CLK_OUT and CLK_POST Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

40.6.1.27 Strobe DLL control (STROBE_DLL_CTRL)

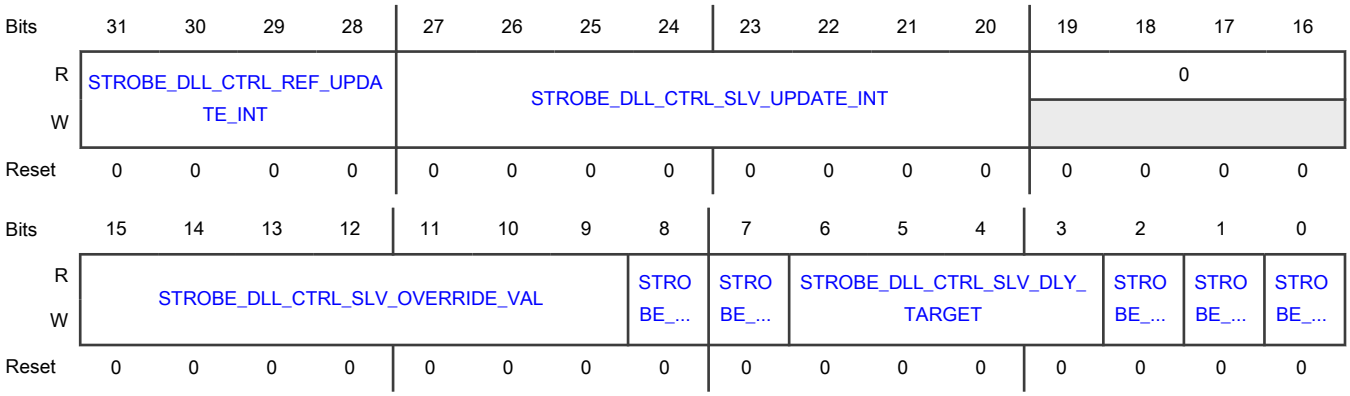
Offset

Register	Offset
STROBE_DLL_CTRL	70h

Function

This register contains the strobe DLL control information.

Diagram



Fields

Field	Function
31-28 STROBE_DLL_CTRL_REF_UPDATE_INT	Strobe DLL control reference update interval The interval cycle is (2 + STROBE_REF_UPDATE_INT) * STROBE_REF_CLOCK. By default, the DLL control loop updates every two STROBE_REF_CLOCK cycles. **NOTE** Increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20	Strobe DLL control target update interval

Table continues on the next page...

Table continued from the previous page...

Field	Function
STROBE_DLL_CTRL_SLV_UPDATE_INT	Target delay line update interval. If default 0 is used, it means 256 cycles of STROBE_REF_CLOCK. A value of 0x0F results in 15 cycles and so on. **NOTE** software can always cause an update of the target-delay line using the STROBE_SLV_FORCE_UPDATE register. The target delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_CTRL_SLV_OVERRIDE_VAL	Strobe DLL control target Override value When STROBE_SLV_OVERRIDE = 1, this field is used to manually select one of 128 physical taps. A value of 0 selects tap 1, and a value of 0x7F selects tap 128.
8 STROBE_DLL_CTRL_SLV_OVERRIDE	Strobe DLL control target override Set this field to 1 to Enable manual override for target delay chain using STROBE_SLV_OVERRIDE_VAL; set this field to 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if STROBE_SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0.
7 STROBE_DLL_CTRL_GATE_UPDATE	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
6-3 STROBE_DLL_CTRL_SLV_DLY_TARGET	Strobe DLL Control Target Delay Target The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an STROBE_REF_CLOCK half-period. The delay is ((STROBE_DLL_CTRL_SLV_DLY_TARGET + 1) * STROBE_REF_CLOCK_PERIOD[1] / 2) / 16, Therefore, the input read-clock can be delayed relative to the input data by (STROBE_REF_CLOCK_PERIOD / 2) / 16 up to (STROBE_REF_CLOCK_PERIOD *8)/16. **NOTE** For the restrictions of delay cell implementation, the minimum delay target that can be set is REF_CLOCK_PERIOD/16.
2 STROBE_DLL_CTRL_SLV_FORCE_UPD	Strobe DLL control target force updated Setting this field to 1, forces the target delay line to update to the DLL calibrated value immediately. The target delay line should automatically update the STROBE_SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the target-line update can only occur if STROBE_SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 STROBE_DLL_CTRL_RESET	Strobe DLL control reset Setting this field to 1 to force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half-period phase shift. This signal is used by the DLL as edge-sensitive, to create a subsequent reset, RESET must be taken low and then asserted again.
0 STROBE_DLL_CTRL_ENABLE	Strobe DLL control enable Set this field to 1 to enable the DLL and delay chain; otherwise, set to 0 to bypasses DLL. **NOTE** Using the target delay line override feature with STROBE_SLV_OVERRIDE and STROBE_SLV_OVERRIDE VAL, the DLL does not need to be enabled.

1. which is 1/STROBE_REF_CLOCK.

40.6.1.28 Strobe DLL status (STROBE_DLL_STATUS)

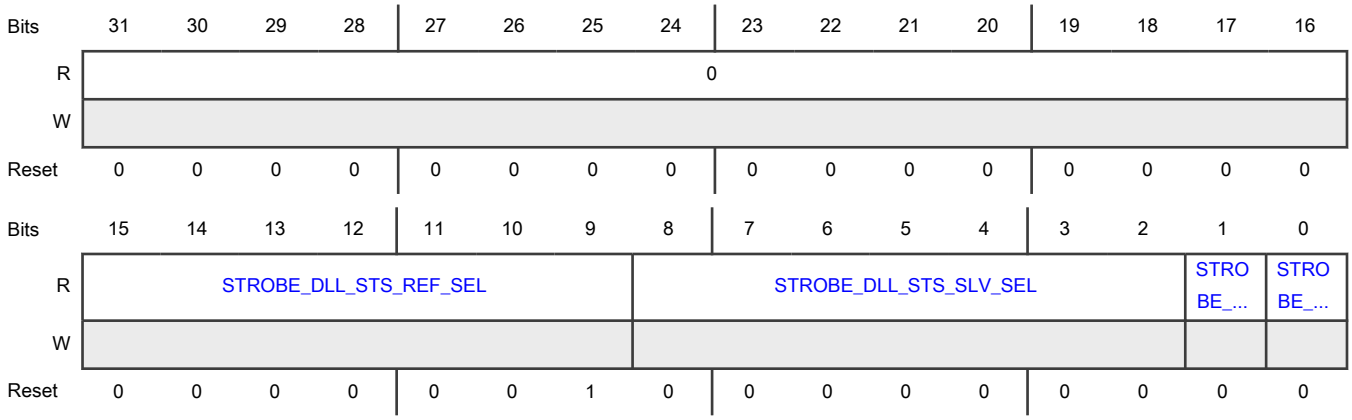
Offset

Register	Offset
STROBE_DLL_STATUS	74h

Function

This register contains the strobe DLL status information. All fields are read only and read the same as the power-reset value.

Diagram



Fields

Field	Function
31-16	This field is reserved.

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_STS_REF_SEL	Strobe DLL status reference select Reference delay line select taps. This is encoded by 7 fields for 127 taps.
8-2 STROBE_DLL_STS_SLV_SEL	Strobe DLL status target select Target delay line select status. This is the instant value generated from reference chain. Because the reference chain can only be updated when STROBE_REF_CLOCK is detected, this value can be updated with the right value when the reference is locked.
1 STROBE_DLL_STS_REF_LOCK	Strobe DLL status reference lock This signifies that the DLL has detected and locked to a half-phase REF_CLOCK shift, it allows the target delay-line to perform programmed clock delays.
0 STROBE_DLL_STS_SLV_LOCK	Strobe DLL status target lock Target delay-line lock status. This signifies that a valid calibration has been set to the target-delay line, and the target-delay line is implementing the programmed delay value.

40.6.1.29 Vendor Specific Register (VEND_SPEC)

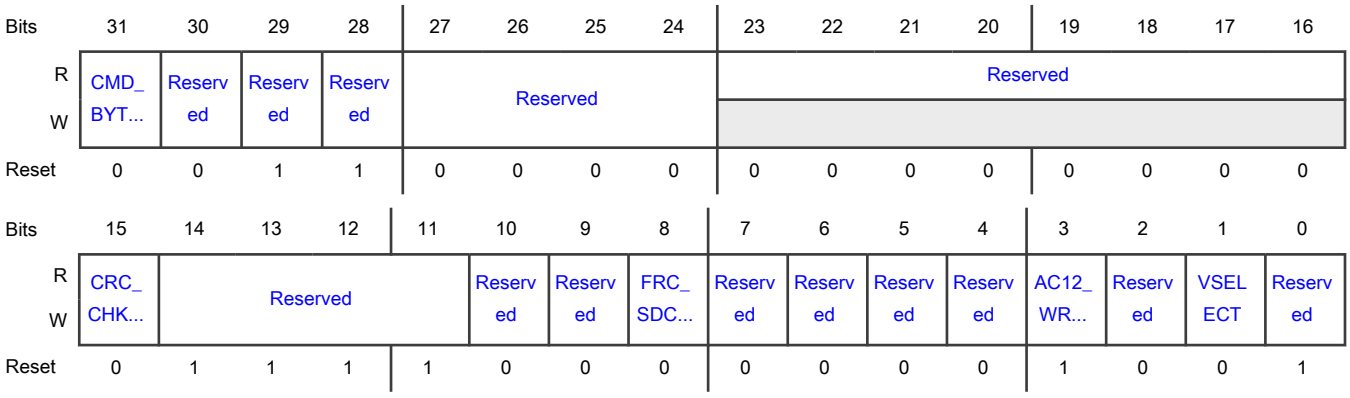
Offset

Register	Offset
VEND_SPEC	C0h

Function

This register contains the vendor specific control/status register.

Diagram



Fields

Field	Function
31 CMD_BYTE_EN	Register byte access for CMD_XFR_TYP This field controls the register byte access for Command Transfer Type (CMD_XFR_TYP). If this field is enabled, the register can be configured through byte write enable. IPS_bus can write only one byte once for one write operation. 0b - Disable. MIX_CTRL[7:0] is read/write and CMD_XFR_TYP[7:0] is read-only. 1b - Enable. MIX_CTRL[7:0] is read-only and CMD_XFR_TYP[7:0] is read/write.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 1
28 —	Reserved Always write as 0.
27-24 —	Reserved Always write as 4'b0000.
23-16 —	Reserved Always write as 8'h00.
15 CRC_CHK_DIS	CRC Check Disable 0b - Check CRC16 for every read data packet and check CRC fields for every write data packet 1b - Ignore CRC16 check for every read data packet and ignore CRC fields check for every write data packet
14-11 —	Reserved Reserved
10 —	Reserved Always write as 0.
9 —	Reserved Always write as 0.
8 FRC_SDCLK_ON	Force CLK Force CLK output active Do not set this bit to 1 unless it is necessary. Also, make sure that this bit is cleared when uSDHC's clock is about to be changed (frequency change, clock source change, or delay chain tuning). 0b - CLK active or inactive is fully controlled by the hardware. 1b - Force CLK active

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 AC12_WR_CHKBUSY_EN	Check busy enable Check busy enable after auto CMD12 for write data packet 0b - Do not check busy after auto CMD12 for write data packet 1b - Check busy after auto CMD12 for write data packet
2 —	Reserved
1 VSELECT	Voltage selection Change the value of output signal VSELECT to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads. 0b - Change the voltage to high voltage range, around 3.0 V 1b - Change the voltage to low voltage range , around 1.8 V
0 —	Reserved Always write as 0.

40.6.1.30 eMMC Boot (MMC_BOOT)

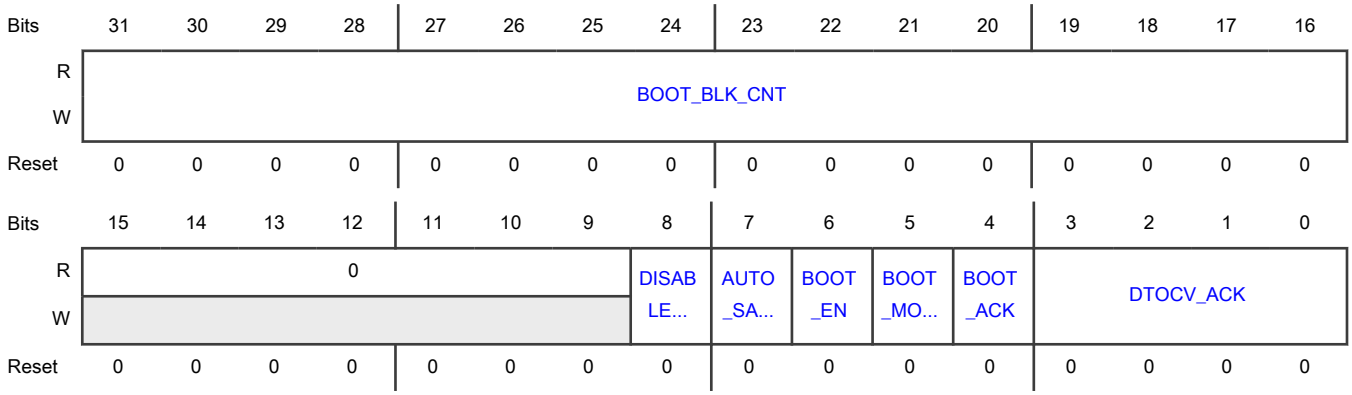
Offset

Register	Offset
MMC_BOOT	C4h

Function

This register contains the eMMC Fast Boot control register.

Diagram



Fields

Field	Function
31-16 BOOT_BLK_CNT	Stop At Block Gap value of automatic mode The value defines the Stop At Block Gap value of automatic mode. When received, card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Attributes Register, field 31 - 16 of 0x04.
15-9 —	Reserved
8 DISABLE_TIME_OUT	Time out **NOTE** When this field is set, there is no timeout check no matter whether BOOT_EN is set or not. 0b - Enable time out 1b - Disable time out
7 AUTO_SABG_EN	Auto stop at block gap During boot, enable auto stop at block gap function. This function is triggered, and host stops at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot enable Boot mode enable 0b - Fast boot disable 1b - Fast boot enable
5 BOOT_MODE	Boot mode Boot mode select 0b - Normal boot 1b - Alternative boot

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 BOOT_ACK	BOOT ACK Boot ACK mode select 0b - No ack 1b - Ack
3-0 DTOCV_ACK	DTOCV_ACK Boot ACK time out counter value 0000b - SDCLK x 2^32 0001b - SDCLK x 2^33 0010b - SDCLK x 2^18 0011b - SDCLK x 2^19 0100b - SDCLK x 2^20 0101b - SDCLK x 2^21 0110b - SDCLK x 2^22 0111b - SDCLK x 2^23 1110b - SDCLK x 2^30 1111b - SDCLK x 2^31

40.6.1.31 Vendor Specific 2 Register (VEND_SPEC2)

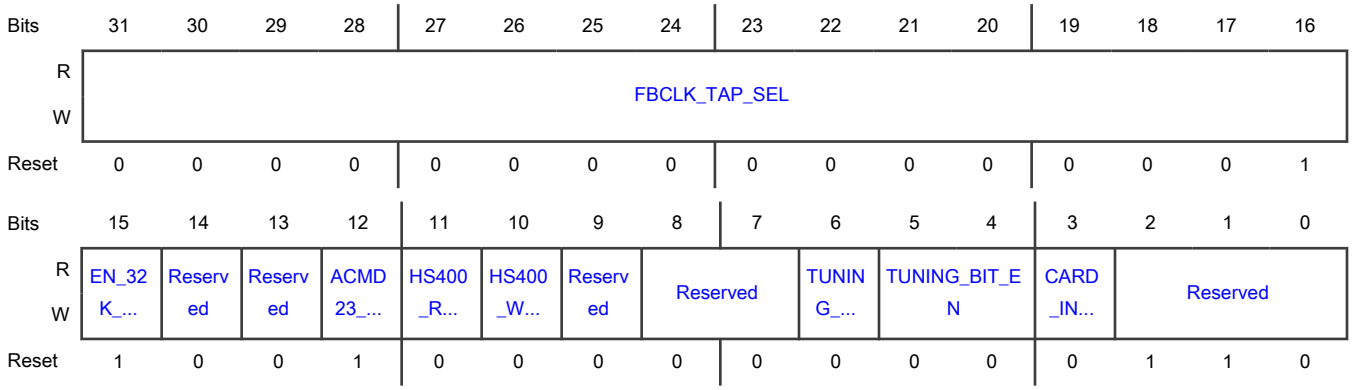
Offset

Register	Offset
VEND_SPEC2	C8h

Function

This register contains the vendor specific control 2 register.

Diagram



Fields

Field	Function
31-16 FBCLK_TAP_SEL	Enable extra delay on internal feedback clock It is a hot code. The default value is 16'h0001 that means the length of feedback delay chain is 1. The value of 16'h8000 means the length of feedback delay chain is 16.
15 EN_32K_CLK	Select the clock source for host card detection. It can use low power clock for card detection, by setting this bit to 1. 0b - Use the peripheral clock (ipg_clk) for card detection. 1b - Use the low power clock (ipg_clk_lp) for card detection.
14 —	Reserved
13 —	Reserved
12 ACMD23_ARGU2_EN	Argument2 register enable for ACMD23 0b - Disable 1b - Argument2 register enable for ACMD23 sharing with SDMA system address register. Default is enabled.
11 HS400_RD_CLK_STOP_EN	HS400 read clock stop enable Only stop clock at read block gap.
10 HS400_WR_CLK_STOP_EN	HS400 write clock stop enable Only stop clock at write block gap.
9	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
8-7 —	Reserved
6 TUNING_CMD_EN	Tuning command enable Enable the auto tuning circuit to check the CMD line. 0b - Auto tuning circuit does not check the CMD line. 1b - Auto tuning circuit checks the CMD line.
5-4 TUNING_BIT_EN	Tuning bit enable Enable the auto tuning circuit to check the DATA[7:0]. The value option is as the list below. 00b - Enable Tuning circuit for DATA[3:0] 01b - Enable Tuning circuit for DATA[7:0] 10b - Enable Tuning circuit for DATA[0] 11b - Invalid
3 CARD_INT_D3_TEST	Card interrupt detection test This field is used only for debugging. 0b - Check the card interrupt only when DATA3 is high. 1b - Check the card interrupt by ignoring the status of DATA3.
2-0 —	Reserved

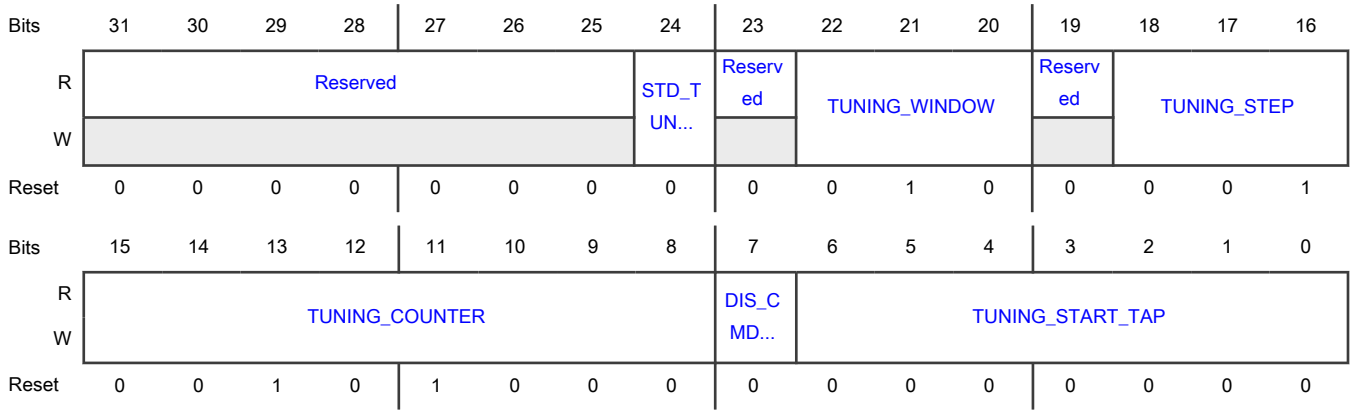
40.6.1.32 Tuning Control (TUNING_CTRL)

Offset

Register	Offset
TUNING_CTRL	CCh

Function
The register contains configuration of tuning circuit.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 STD_TUNING_EN	Standard tuning circuit and procedure enable This field is used to enable standard tuning circuit and procedure.
23 —	Reserved
22-20 TUNING_WINDOW	Data window Select data window value for auto tuning
19 —	Reserved
18-16 TUNING_STEP	TUNING_STEP The increasing delay cell steps in tuning procedure.
15-8 TUNING_COUNTER	Tuning counter The MAX repeat CMD19 times in tuning procedure.
7 DIS_CMD_CHK_FOR_STD_TUNING	Disable command check for standard tuning Writing 1 to this field disables command check (command CRC, CMD end bit error, and CMD index error or CMD timeout error) for standard tuning flow after each tuning command is sent.
6-0	Tuning start

Table continues on the next page...

Table continued from the previous page...

Field	Function
TUNING_START_TAP	The start delay cell point when send first CMD19 in tuning procedure.

40.6.1.33 Command Queuing Version (CQVER)

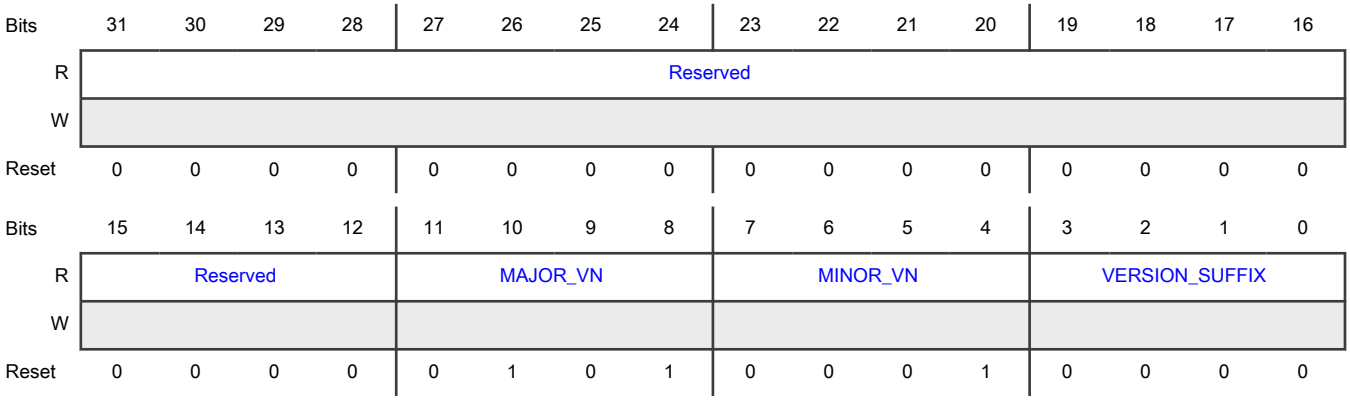
Offset

Register	Offset
CQVER	100h

Function

This register provides information about the version of the eMMC CQ standard that is implemented by CQE, in BCD format. The current version is 5.1.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-8 MAJOR_VN	eMMC major version number Digit left of decimal point, in BCD format
7-4 MINOR_VN	eMMC minor version number Digit right of decimal point, in BCD format
3-0	eMMC version suffix

Table continues on the next page...

Table continued from the previous page...

Field	Function
VERSION_SUFFIX	The second digit right of decimal point, in BCD format

40.6.1.34 Command Queuing Capabilities (CQCAP)

Offset

Register	Offset
CQCAP	104h

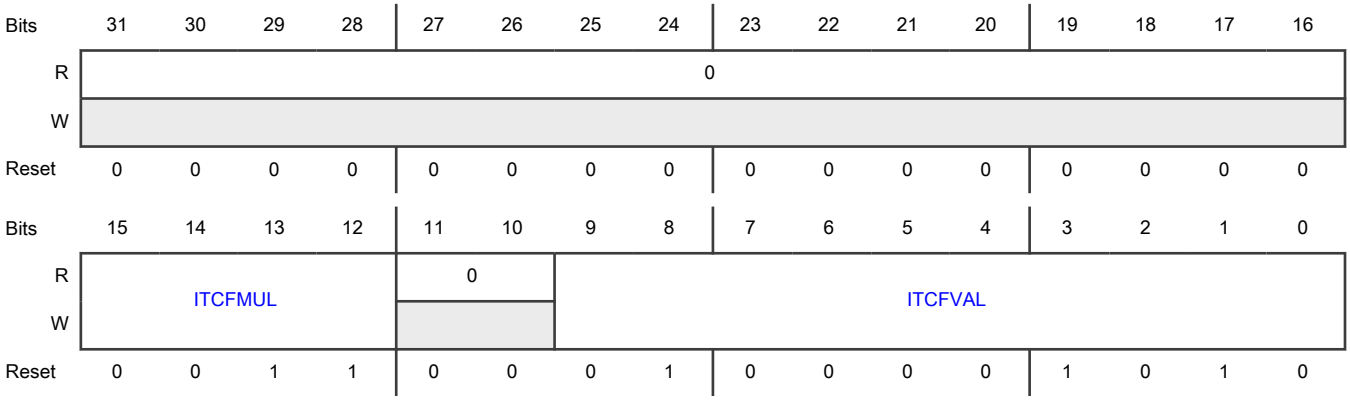
Function

This register indicates hardware capabilities.

NOTE

This register must not be configured and must be used as a read-only register. Writing to the read-write fields of this register does not affect any functionality. Reset values reflect the correct capability information.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-12 ITCFMUL	Internal timer clock frequency multiplier ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0001b - 0.001 MHz 0010b - 0.01 MHz 0011b - 0.1 MHz 0100b - 1 MHz 0101b - 10 MHz 0110b-1001b - Reserved
11-10 —	Reserved
9-0 ITCFVAL	Internal timer clock frequency value ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling. The clock frequency is calculated as ITCFVAL* ITCFMUL. For example, to encode 19.2 MHz, ITCFVAL is C0h (= 192 decimal) and ITCFMUL is 3h (0.1 MHz): 192 * 0.1 MHz = 19.2 MHz.

40.6.1.35 Command Queuing Configuration (CQCFG)

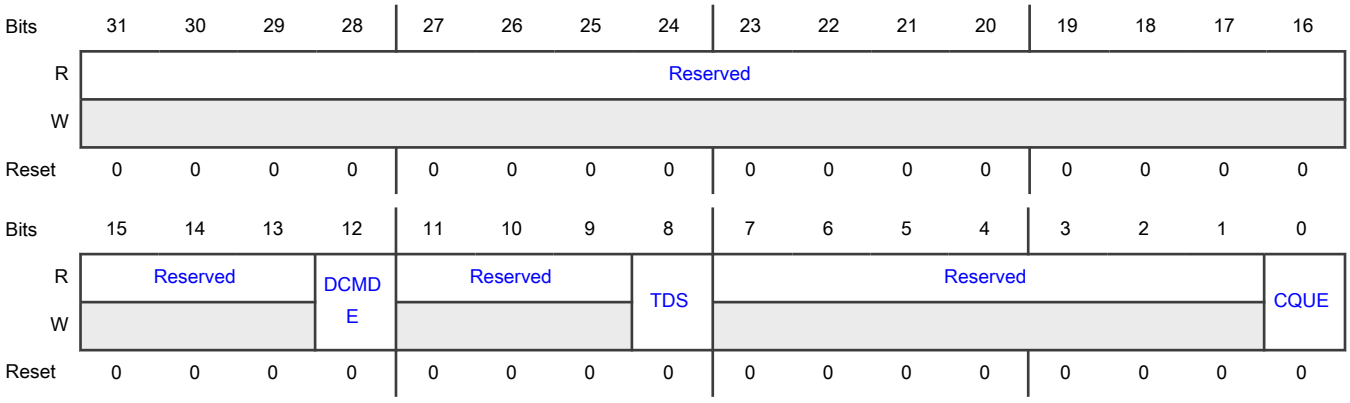
Offset

Register	Offset
CQCFG	108h

Function

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

Diagram



Fields

Field	Function
31-13 —	Reserved
12 DCMDE	Direct command (DCMD) enable This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor. CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor. 0b - Task descriptor in slot #31 is a Data Transfer Task Descriptor 1b - Task descriptor in slot #31 is a DCMD Task Descriptor
11-9 —	Reserved
8 TDS	Task descriptor size This bit indicates whether the task descriptor size is 128 bits or 64 bits as detailed in Data Structures section. This bit can only be configured when Command Queuing Enable bit is ‘0’ (command queuing is disabled) 0b - Task descriptor size is 64 bits 1b - Task descriptor size is 128 bits
7-1 —	Reserved
0 CQUE	Command queuing enable Software shall write '1' this bit when in order to enable command queuing mode (for example, enable CQE). When this bit is 0, CQE is disabled and software controls the eMMC bus using the legacy eMMC host controller. Before software writes '1' to this bit, software shall verify that the eMMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queuing mode, it shall clear all previous tasks if such exist before setting this bit to 0.

40.6.1.36 Command Queuing Control (CQCTL)

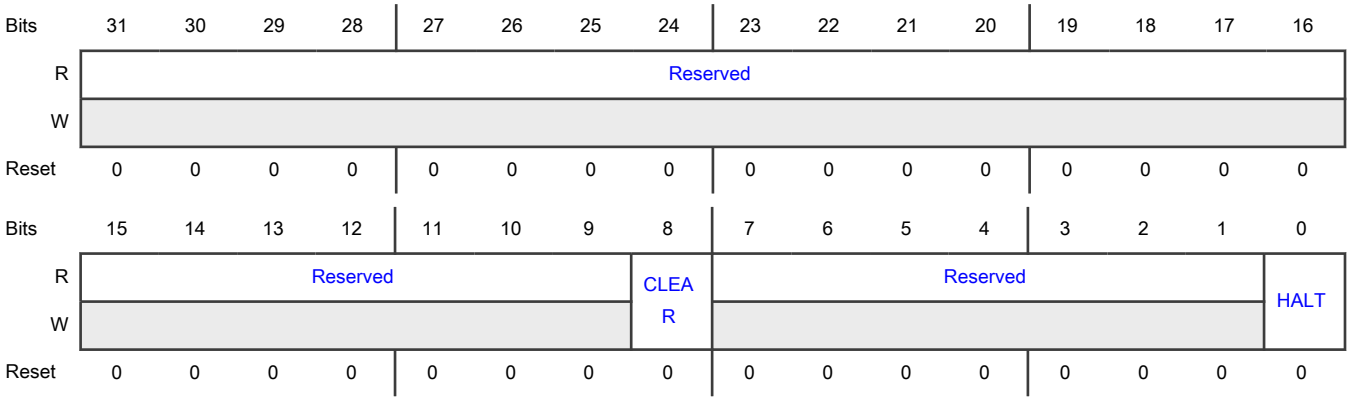
Offset

Register	Offset
CQCTL	10Ch

Function

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 CLEAR	Clear all tasks Software shall write '1' this bit when it wants to clear all the tasks sent to the chip. This bit can only be written when CQE is in halt state (for example, Halt bit is 1). When software writes 1, the value of the register is updated to '1', and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit. Software should poll on this bit until it is set to back 0 and may then resume normal operation, by clearing the Halt bit. CQE does not communicate to the chip that the tasks were cleared. It is software's responsibility to order the chip to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing '0' to this register shall have no effect.
7-1 —	Reserved
0 HALT	Halt Host software shall write '1' to the bit when it wants to acquire software control over the eMMC bus and disable CQE from issuing commands on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT) When software writes '1', CQE shall complete the ongoing task if such a task is in progress. Once the task is completed and CQE is in idle state, CQE shall not issue new commands and shall indicate so to software by setting this bit to 1. Software may poll on this bit until it is set to 1, and may only then send commands on the eMMC bus. In order to exit halt state (for example, resume CQE activity), software shall clear this bit (write '0'). Writing '0' when the value is already '0' shall have no effect.

40.6.1.37 Command Queuing Interrupt Status (CQIS)

Offset

Register	Offset
CQIS	110h

Function

This register indicates pending interrupts that require service. Each bit in this registers is asserted in response a specific event, only if the respective bit is set in CQISTE register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												TCL	RED	TCC	HAC
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3 TCL	Task cleared This status bit is asserted (if CQISTE[TCL_STE]=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).
2 RED	Response error detected interrupt This status bit is asserted (if CQISTE[RED_STE]=1) when a response is received with an error bit set in the device status field. The contents of the device status field are listed in 6.13. Software uses CQRMEM register to configure which device status bit fields may trigger an interrupt, and which are masked.
1 TCC	Task complete interrupt This status bit is asserted (if CQISTE[TCC_STE]=1) when at least one of the following two conditions are met:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• A task is completed and the INT bit is set in its Task Descriptor • Interrupt caused by Interrupt Coalescing logic (see 0)
0 HAC	Halt complete interrupt This status bit is asserted (if CQISTE[HAC_STE]=1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

40.6.1.38 Command Queuing Interrupt Status Enable (CQISTE)

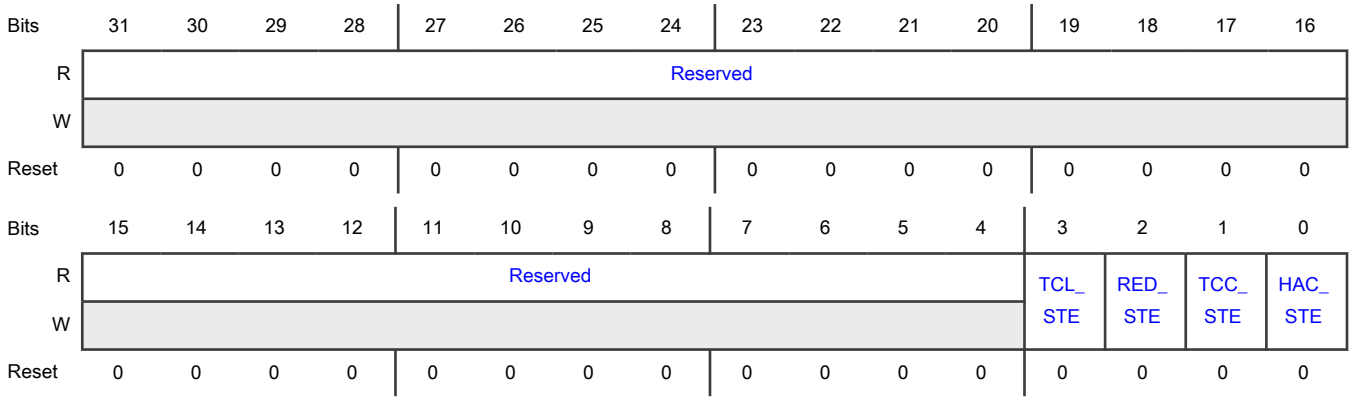
Offset

Register	Offset
CQISTE	114h

Function

This register enables and disables the reporting of the corresponding interrupt to host software in CQIS register. When a bit is set ('1') and the corresponding interrupt condition is active, then the bit is CQIS is asserted. Interrupt sources that are disabled (‘0’) are not indicated in the CQIS register. This register is bit-index matched to CQIS register.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 TCL_STE	Task cleared status enable Related to CQIS[TCL]

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - CQIS[TCL] is disabled 1b - CQIS[TCL] is set when its interrupt condition is active
2 RED_STE	Response error detected status enable Related to CQIS[RED] 0b - CQIS[RED]is disabled 1b - CQIS[RED] is set when its interrupt condition is active
1 TCC_STE	Task complete status enable Related to CQIS[TCC] 0b - CQIS[TCC] is disabled 1b - CQIS[TCC] is set when its interrupt condition is active
0 HAC_STE	Halt complete status enable Related to CQIS[HAC] 0b - CQIS[HAC] is disabled 1b - CQIS[HAC] is set when its interrupt condition is active

40.6.1.39 Command Queuing Interrupt Signal Enable (CQISGE)

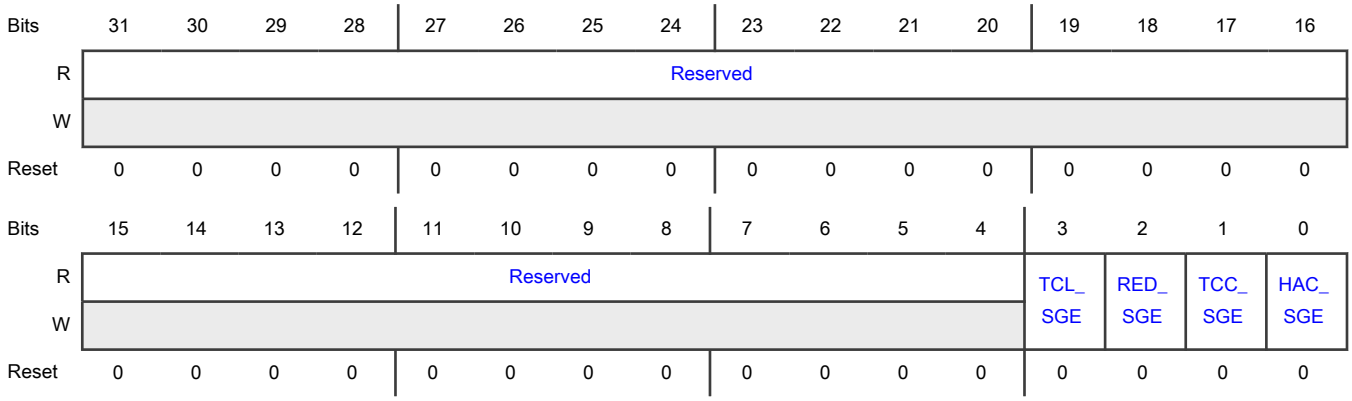
Offset

Register	Offset
CQISGE	118h

Function

This register enables and disables the generation of interrupts to host software. When a bit is set ('1') and the corresponding bit in CQIS is set, then an interrupt is generated. Interrupt sources that are disabled (‘0’) are still indicated in the CQIS register. This register is bit-index matched to CQIS register.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 TCL_SGE	Task cleared signal enable When set and CQIS[TCL] is asserted, the CQE generates an interrupt.
2 RED_SGE	Response error detected signal enable When set and CQIS[RED] is asserted, the CQE generates an interrupt.
1 TCC_SGE	Task complete signal enable When set and CQIS[TCC] is asserted, the CQE generates an interrupt.
0 HAC_SGE	Halt complete signal enable When set and CQIS[HAC] is asserted, the CQE generates an interrupt.

40.6.1.40 Command Queuing Interrupt Coalescing (CQIC)

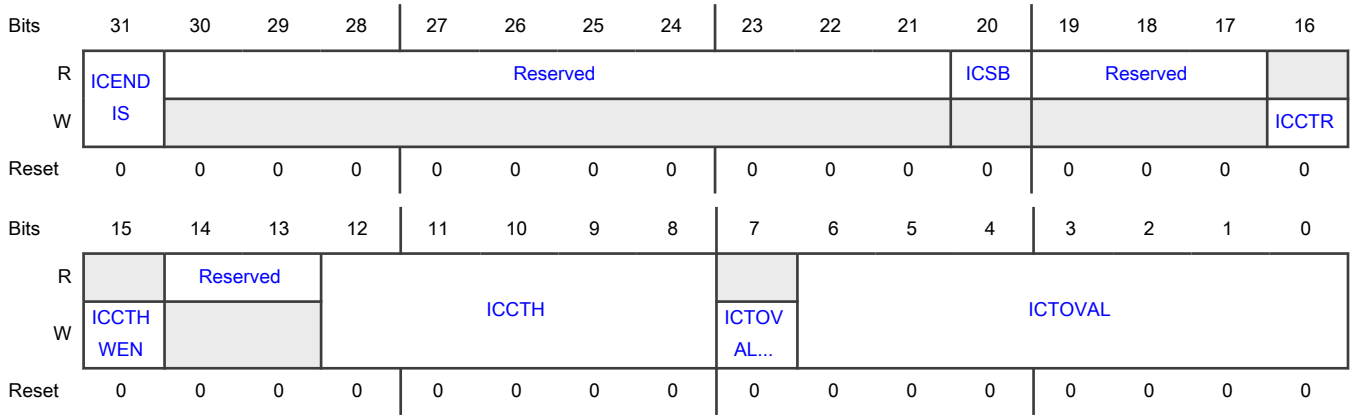
Offset

Register	Offset
CQIC	11Ch

Function

This register controls the interrupt coalescing feature.

Diagram



Fields

Field	Function
31 ICENDIS	Interrupt coalescing enable/disable When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
30-21 —	Reserved
20 ICSB	Interrupt coalescing status This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (for example, ICSB is set if and only if IC counter > 0). 0b - No task completions have occurred since last counter reset (IC counter =0) 1b - At least one task completion has been counted (IC counter >0)
19-17 —	Reserved
16 ICCTR	Counter and timer reset When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
15 ICCTHWEN	Interrupt coalescing counter threshold write enable When software writes ‘1’, the value ICCTH is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICCTH is not updated. **NOTE** Write operations to ICCTH are only allowed when the task queue is empty.
14-13 —	Reserved
12-8	Interrupt coalescing counter threshold

Table continues on the next page...

Table continued from the previous page...

Field	Function
ICCTH	Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor) which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in ICCTH. The maximum allowed value is 31. **NOTE** When ICCTH is 0, task completions are not counted, and counting-based interrupts are not generated. In order to write to this field, the ICCTHWEN field must be set at the same write operation.
7 ICTOVALWEN	Interrupt coalescing timeout value write enable When software writes ‘1’, the value ICTOVAL is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICTOVAL is not updated. **NOTE** Write operations to ICTOVAL are only allowed when the task queue is empty.
6-0 ICTOVAL	Interrupt coalescing timeout value Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field it generates an interrupt and stops. The timer’s unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 01h (1024 clock periods) and the maximum value is 7Fh (127*1024 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in ICTOVAL is 10h, the calculated polling period is 16*1024*52.08 ns= 853.33 us **NOTE** When ICTOVAL is 0, the timer is not running, and timer-based interrupts are not generated. In order to write to this field, the ICTOVALWEN bit must be set at the same write operation.

40.6.1.41 Command Queuing Task Descriptor List Base Address (CQTDLBA)

Offset

Register	Offset
CQTDLBA	120h

Function

This register is used for configuring the lower 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDLBA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDLBA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TDLBA	Task descriptor list base address This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on 1 KByte boundary: The lower 10 bits of this register shall be set to 0 by software and shall be ignored by CQE.

40.6.1.42 Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)

Offset

Register	Offset
CQTDLBAU	124h

Function

This register is used for configuring the upper 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TDLBAU	Task descriptor list base address This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode.

40.6.1.43 Command Queuing Task Doorbell (CQTDBR)

Offset

Register	Offset
CQTDBR	128h

Function

Using this register, software triggers CQE to process a new task.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TDBR	Task doorbell Software configures TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to '0' by CQE in one of the following events: 1. When a task execution is completed (with success or error) 2. The task is cleared using CQTCLR register 3. All tasks are cleared using CQCTL register 4. CQE is disabled using CQCFG register Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission: CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

40.6.1.44 Command Queuing Task Completion Notification (CQTCN)

Offset

Register	Offset
CQTCN	12Ch

Function

This register is used by CQE to notify software about completed tasks.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TCN	Task complete notification CQE shall set bit n of this register (at the same time it clears bit n of CQTDBR) when a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

40.6.1.45 Command Queuing Device Queue Status (CQDQS)

Offset

Register	Offset
CQDQS	130h

Function

This register stores the most recent value of the device’s queue status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DQS	Device queue status Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command (For example, the chip's queue status).

40.6.1.46 Command Queuing Device Pending Tasks (CQDPT)

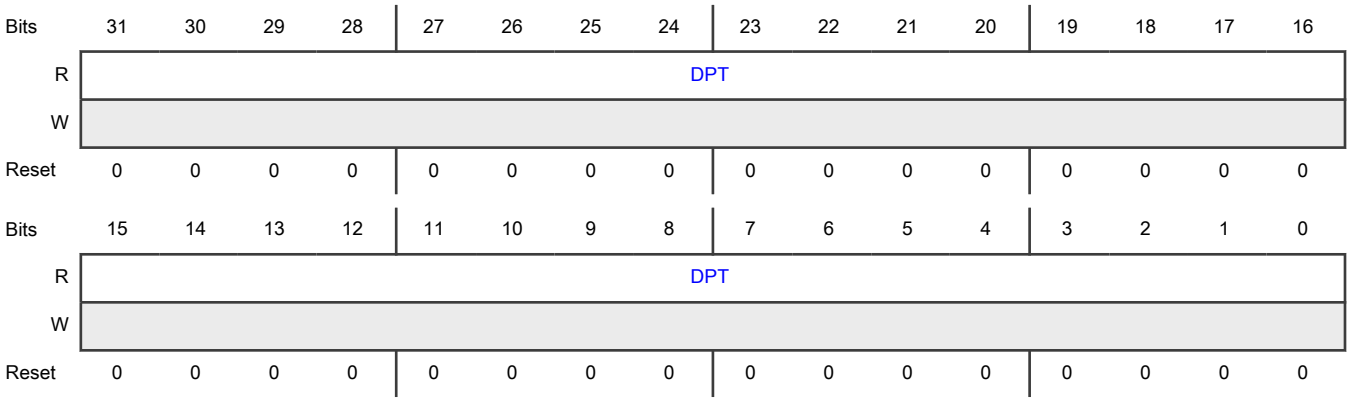
Offset

Register	Offset
CQDPT	134h

Function

This register indicates to software which tasks are queued in the device, awaiting execution.

Diagram



Fields

Field	Function
31-0 DPT	Device pending tasks Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task hasn't been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution. Software needs to read this register in the task-discard procedure, when the controller is halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the software orders CQE to resume its operation using CQCTL register.

40.6.1.47 Command Queuing Task Clear (CQTCLR)

Offset

Register	Offset
CQTCLR	138h

Function

This register is used for removing an outstanding task in the CQE. The register should be used only when CQE is in Halt state.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCLR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TCLR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TCLR	Task clear Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes '1' to a bit in this register, CQE updates the value to '1', and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete. Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host software shall send CMDQ_TASK _MGMT while CQE is still in Halt state. Host driver is not allowed to use this register to clear multiple tasks at the same time. Clearing multiple tasks can be done using CQCTL register. Writing 0 to a register bit shall have no impact.

40.6.1.48 Command Queuing Send Status Configuration 1 (CQSSC1)

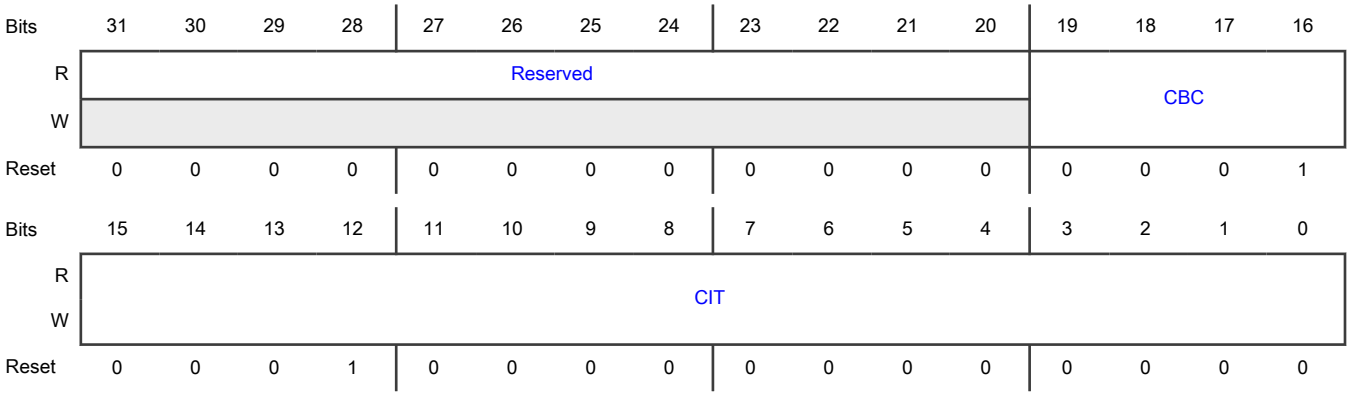
Offset

Register	Offset
CQSSC1	140h

Function

The register controls the when SEND_QUEUE_STATUS commands are sent.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-16 CBC	Send status command block counter This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue. A value of *n* means CQE sends status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command is not sent during the transaction. Instead it will be sent only when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.
15-0 CIT	Send status command idle timer This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). Default interval is: 4096 clock periods. For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSSC1.CIT is 1000h, the calculated polling period is 4096*52.08 ns= 213.33 us

40.6.1.49 Command Queuing Send Status Configuration 2 (CQSSC2)

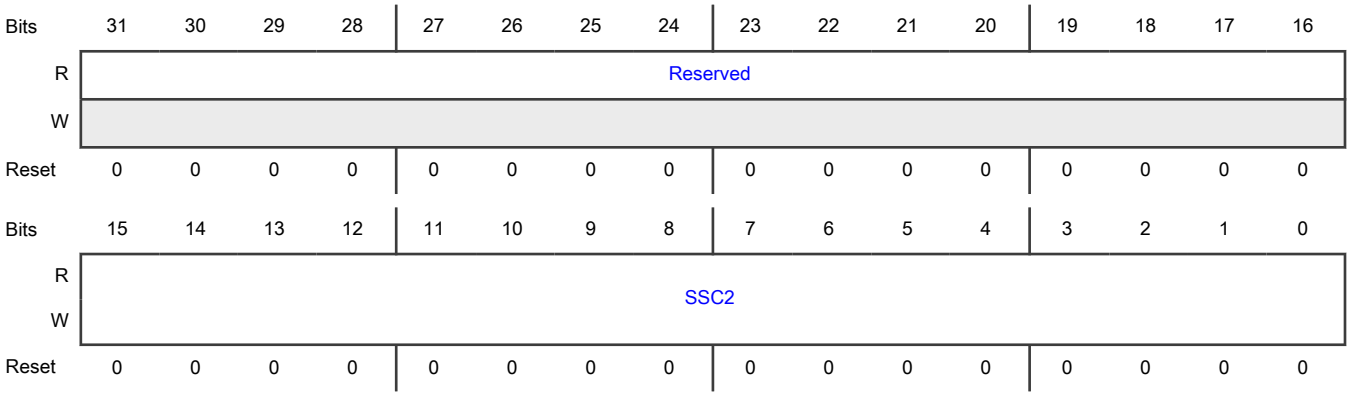
Offset

Register	Offset
CQSSC2	144h

Function

This register is used for configuring RCA field in SEND_QUEUE_STATUS command argument.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 SSC2	Send queue status RCA This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command argument. CQE copies this field to bits 31:16 of the argument when transmitting SEND_ QUEUE_STATUS (CMD13) command.

40.6.1.50 Command Queuing Command Response for Direct-Command Task (CQCRDCT)

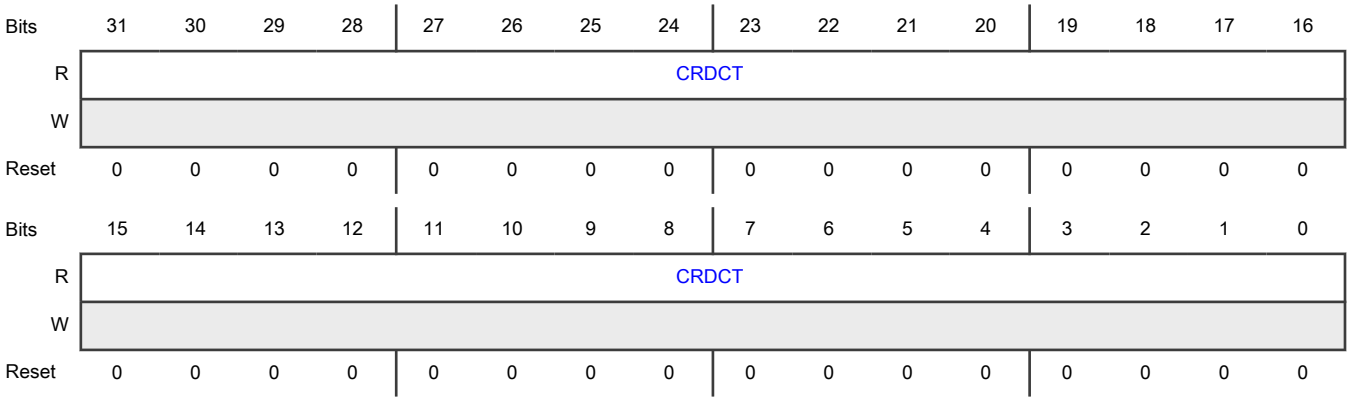
Offset

Register	Offset
CQCRDCT	148h

Function

This register is used for passing the response of a DCMD task to software.

Diagram



Fields

Field	Function
31-0 CRDCT	Direct command last response This register contains the response of the command generated by the last direct-command (DCMD) task which was sent. CQE updates this register when it receives the response for a DCMD task. This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE.

40.6.1.51 Command Queuing Response Mode Error Mask (CQRMEM)

Offset

Register	Offset
CQRMEM	150h

Function

This register controls the generation of Response Error Detection (RED) interrupt.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RMEM															
Reset	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RMEM															
Reset	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Function
31-0 RMEM	Response mode error mask This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses. The reset value of this register is set to trigger an interrupt on all “Error” type bits in the device status, as defined in 6.13. **NOTE** Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic. 0000_0000_0000_0000_0000_0000_0000_0000b - When a R1/R1b response is received, bit i in the device status is ignored 0000_0000_0000_0000_0000_0000_0000_0001b - When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated

40.6.1.52 Command Queuing Task Error Information (CQTERRI)

Offset

Register	Offset
CQTERRI	154h

Function

This register is updated by CQE when an error occurs on data or command related to a task activity.

When such error is detected by CQE or indicated by the eMMC controller CQE stores in CQTERRI the task IDs and the command indices of the commands which were executed on the command line and data lines when the error occurred.

Software is expected to use this information in the error recovery procedure. See B.2.8 for more details.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DTEFV	Reserved			DTETID				Reserved				DTECI			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RMEFV	Reserved			RMETID				Reserved				RMECI			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DTEFV	Data transfer error fields valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a data transfer is in progress when the error is detected/indicated, the bit is set to 1. If a no data transfer is in progress when the error is detected/indicated, the bit is cleared to 0.
30-29 —	Reserved
28-24 DTETID	Data transfer error task ID This field indicates the ID of the task which was executed on the data lines when an error occurred. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
23-22 —	Reserved
21-16 DTECI	Data transfer error command index This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
15 RMEFV	Response mode error fields valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a command transaction is in progress when the error is detected/indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
14-13 —	Reserved
12-8 RMETID	Response mode error task ID This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.
7-6 —	Reserved
5-0 RMECI	Response mode error command index This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

40.6.1.53 Command Queuing Command Response Index (CQCRI)

Offset

Register	Offset
CQCRI	158h

Function

This register stores the index of the last received command response.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										LCMDRI					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-6 —	Reserved
5-0 LCMDRI	Last command response index This field stores the index of the last received command response. CQE shall update the value every time a command response is received..

40.6.1.54 Command Queuing Command Response Argument (CQCRA)

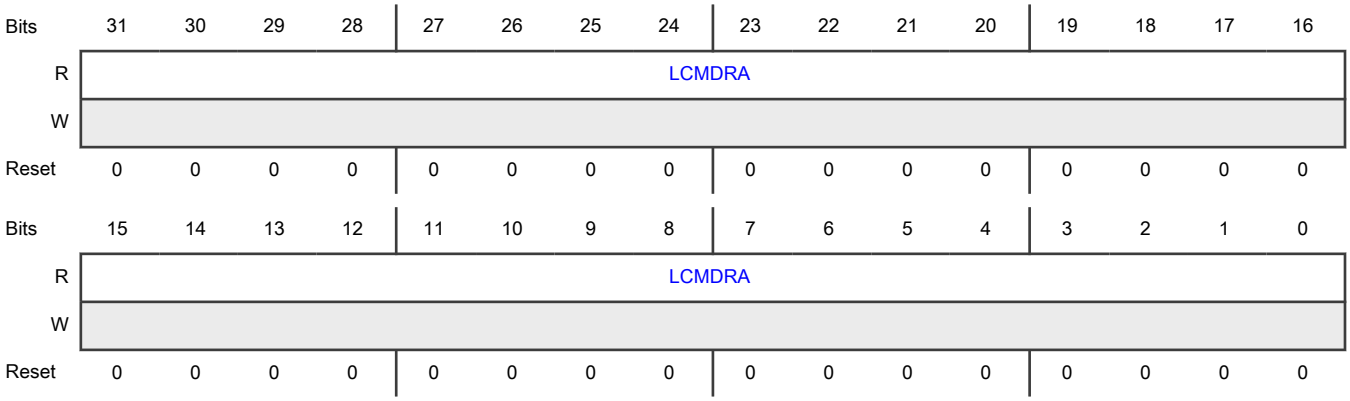
Offset

Register	Offset
CQCRA	15Ch

Function

This register stores the argument of the last received command response.

Diagram



Fields

Field	Function
31-0 LCMDRA	Last command response argument This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

40.7 Glossary

eMMC	Embedded Multi-Media Card
SD Card	Secure Digital Card

eSD	embedded SD Memory Device
SDIO	SD Input/Output
SD Combo	Memory and I/O SD Card
SDSC	SD memory card with Standard Capacity, up to and including 2 GB
SDHC	SD memory card with High Capacity, more than 2 GB, up to and including 32 GB
SDXC	SD memory card with Extended Capacity, more than 32 GB, up to and including 2 TB
Default Speed mode	up to 12.5 MB/s (3.3 V signaling, frequency up to 25 MHz)
High Speed mode	up to 25 MB/s (3.3 V signaling, frequency up to 50 MHz)
SDR12	up to 12.5 MB/s (1.8 V signaling, frequency up to 25 MHz)
SDR25	up to 25 MB/s (1.8 V signaling, frequency up to 50 MHz)
SDR50	up to 50 MB/s (1.8 V signaling, frequency up to 100 MHz)
SDR104	up to 104 MB/s (1.8 V signaling, frequency up to 208 MHz)
DDR50	up to 50 MB/s (1.8 V signaling, frequency up to 50 MHz, sampled on both clock edges)
DLL	Delay Line, which is used to tune SDCLK
UHS-I	Ultra High Speed - I
CMD	Command line
ACMD	Application CMD
DATA	Data line
SDCLK	Clock line
DMA	Direct Memory Access
ADMA	Advanced DMA
ADMA1	Advanced DMA1, only supports 4 KB aligned data transfer
ADMA2	Advanced DMA2, no 4 KB aligned limitation
Rsv	Reserved
Nop	None operation
CLK	Clock
CQE	Command Queuing Engine
TDL	Task Descriptor List
SDR	Single Data Rate signaling
DDR	Double Data Rate signaling
ASx	Abort Select
CCCR	SDIO or Combo Card Common Control Registers
CID	Card Identification number register
CIS	Card Information Structure register
CSD	Card Specific Data register

Chapter 41
External Serial Peripheral Interface (XSPI)

41.1 Chip-specific XSPI information

Table 324. Reference links to related information

Topic	Related module	Reference
Full description	XSPI	XSPI
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Memory map unit	MMU	MMU
Cache memory controller	XCACHE	XCACHE
Cache performance monitor	CMX_PERFMON	CMX_PERFMON
Prime boot from XSPI NOR		Boot ROM

41.1.1 Module instances

This chip contains three instances of the XSPI module, XSPI0, XSPI1, and XSPI2.

Table 325. Module instance features

Features	XSPI0	XSPI1	XSPI2
Located in	Controller in VDD2_COMP Memory interface in VDDN_COM	Controller in VDD2_COMP Memory interface in VDDN_COM	Controller in VDD2_MEDIA Memory interface in VDDN_COM
Controlled memories[1]	128 MB	128 MB	256 MB
Related caches	32 KB	32 KB	No
Related cache controller	CACHE64_CTRL0	CACHE64_CTRL1	No
Initiators access XSPI via M-bus[2]	CPU0 system bus, eDMA0, eDMA1, ELS, PKC, ETR	CPU0 code bus, eDMA0, eDMA1, ELS, PKC, ETR	No

Table continues on the next page...

Table 325. Module instance features (continued)

Features	XSPI0	XSPI1	XSPI2
Initiators access XSPI via NIC_NEDIA1	JPEGDEC, PNGDEC, VGPU, LCDIF, EZH-V and DAP via NIC_MEDIA0: USB0-1, uSDHC0-1	JPEGDEC, PNGDEC, VGPU, LCDIF, EZH-V and DAP via NIC_MEDIA0: USB0-1, uSDHC0-1	JPEGDEC, PNGDEC, VGPU, LCDIF, EZH-V and DAP via NIC_MEDIA0: USB0-1, uSDHC0-1 via NIC_DSP: HiFi4 via M-bus: CPU0 system bus, eDMA0, eDMA1, ELS, PKC, ETR via S-bus: CPU1 system bus, eDMA2, eDMA3
Initiators access XSPI via NIC_DSP	HiFi4	HiFi4	N/A
Prince CTR/GCM	Yes	Yes	No
Prince XEX	No	Yes	No
x16 mode	No (8 bidirectional data lines)	Yes (16 bidirectional data lines)	Yes (16 bidirectional data lines)
Dummy pad loopback	No	No	No

1. See System memory map for more details.
2. The initiators access XSPI via M-bus also via cache.

41.1.2 IPED XEX write limitation

When IPED XEX is used in this chip, CPU0, eDMA0 and eDMA1 are the only managers that should write the memory, and the CACHE64_CTRL must be used with writeback mode enabled for the IPED XEX region(s). Other managers can be used to read the IPED XEX region, but they cannot write.

41.1.3 Stop request

In this chip, SLEEPCON modules can be used to send Stop request to the XSPI modules.

Table 326. Stop request to XSPI instances

Send stop request to	Control register in SLEEPCON
XSPI0	SYSCON0.PRIVATE_TCTRL2[XSPI0_STOP]
XSPI1	SYSCON0.PRIVATE_TCTRL2[XSPI1_STOP]
XSPI2	SYSCON0.SHA_MED_TCTRL0[XSPI2_STOP] or SYSCON1.SHA_MED_TCTRL0[XSPI2_STOP]

41.1.4 Configure XSPI to boot from XSPI NOR

In order to boot from XSPI NOR, configure XSPI depending on XSPI boot configuration.

41.1.5 Hard reset

In this chip, you can use the following register bits to reset XSPI modules:

Table 327. Reset XSPI modules

XSPI modules	Reset by
XSPI0	Write 1b to RSTCTL0.PRSTCTL2[XSPI0] or RSTCTL0.PRSTCTL2_SET[XSPI0]
XSPI1	Write 1b to RSTCTL0.PRSTCTL2[XSPI1] or RSTCTL0.PRSTCTL2_SET[XSPI1]
XSPI2	Write 1b to RSTCTL4.PRSTCTL0[XSPI2] or RSTCTL4.PRSTCTL0_SET[XSPI2]

41.1.6 SOC configuration register (SOCCR)

In this chip, the SOCCR[23:0] is not connected.

41.1.7 Manager ID allocation

In this chip, the AHB manager ID (MSTRID) is allocated as below.

Table 328. Manager ID allocations

Manager ID	Module	Manager ID	Module
xxxx_xxxx_xxx0_0000b	CPU0 system	xxxx_xxxx_xxx1_0000b	CPU0 code
xxxx_xxxx_xxx0_0001b	eDMA0	xxxx_xxxx_xxx1_0001b	Reserved
xxxx_xxxx_xxx0_0010b	eDMA1	xxxx_xxxx_xxx1_0010b	Reserved
xxxx_xxxx_xxx0_0011b	ELS	xxxx_xxxx_xxx1_0011b	Reserved
xxxx_xxxx_xxx0_0100b	HiFi4	xxxx_xxxx_xxx1_0100b	Reserved
xxxx_xxxx_xxx0_0101b	CPU1 system	xxxx_xxxx_xxx1_0101b	CPU1 code
xxxx_xxxx_xxx0_0110b	eDMA2	xxxx_xxxx_xxx1_0110b	JPEGDEC
xxxx_xxxx_xxx0_0111b	eDMA3	xxxx_xxxx_xxx1_0111b	PNG
xxxx_xxxx_xxx0_1000b	HiFi1	xxxx_xxxx_xxx1_1000b	Reserved
xxxx_xxxx_xxx0_1001b	EZH-V	xxxx_xxxx_xxx1_1001b	Reserved
xxxx_xxxx_xxx0_1010b	PKC	xxxx_xxxx_xxx1_1010b	Reserved
xxxx_xxxx_xxx0_1011b	eMMC0	xxxx_xxxx_xxx1_1011b	eMMC1
xxxx_xxxx_xxx0_1100b	USB0	xxxx_xxxx_xxx1_1100b	USB1 (eUSB)
xxxx_xxxx_xxx0_1101b	VGPU	xxxx_xxxx_xxx1_1101b	LCDIF
xxxx_xxxx_xxx0_1110b	ETR	xxxx_xxxx_xxx1_1110b	Reserved
xxxx_xxxx_xxx0_1111b	DAP	xxxx_xxxx_xxx1_1111b	Reserved

41.1.8 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

41.2 Overview

XSPI is a memory controller IP, designed to support serial flash or RAM memories. The controller includes a programmable sequence engine providing flexibility to support existing and future memory devices. XSPI supports single, dual, quad modes of operation. Also this supports high performance x16 mode of operation in Octal mode.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 329. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

41.2.1 Features

XSPI supports the following features:

- Programmable sequence engine for memory device commands and operations to support all existing and future memory devices, with support for all types of addressing. For example sequences, see Serial flash memory devices.
- Single, dual, and quad modes of operation supported for Quad flash memories
- Serial NAND device support for NOR-compatible command through the IPS interface
- Octal and single IO operation supported for Octal flash memories
- Double Data Rate (DDR)/Double Transfer Rate (DTR) mode that generates data on every edge of the memory clock
- Flash memory data strobe signal to support data sampling in DDR and Single Data Rate (SDR) mode
- HyperRAM support
- TX buffer fill via IPS register space (32-bit access) or AHB (64-bit width interface)
- IPS initiator to read the RX data buffer through the IPS register space (32-bit access).
- Multi-initiator AHB accesses with priority
- Flexible and configurable buffer for each initiator, with total available buffer size of 4096 bytes.
- Direct memory mapping of all AHB memory accesses to the chip system memory space.
- x16 operation with EIGHT PAD support.
- Programmable bit skewing adjustment for dual DQS.
- Enable or disable encryption of data written to external memories.
- Encryption or decryption for random reads or writes based on the data block size or boundaries as expected by IPED PRINCE algorithm with certain limitations.

41.2.2 Block diagram

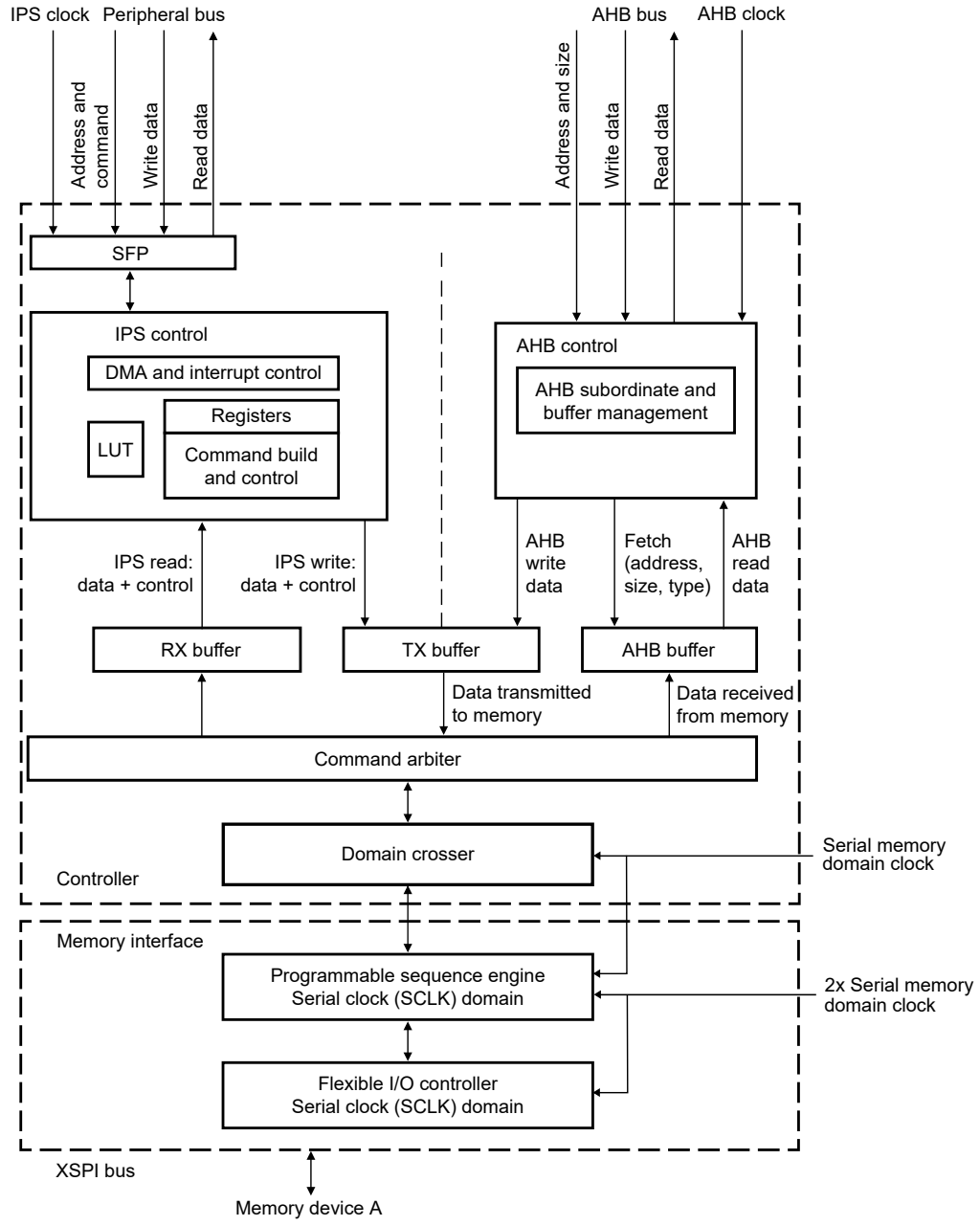


Figure 143. XSPI block diagram

41.3 Functional description

This section provides a functional description of the XSPI module.

41.3.1 Behavior of XSPI in chip modes

Chip mode	XSPI behavior
Normal	For write and read access to an external memory device. See Normal mode overview. • Serial flash memory write: Program data into memory through the IP interface or AHB interface. See Flash memory programming using IP command. • Serial flash memory read: Read the contents of memory through one of two separate read channels: the RX buffer and the AHB buffer. See Flash memory read using IP command for details.
Module disable	Use to disable the serial flash memory clock and AHB commands. Module disable mode stops the clock to XSPI main logic. XSPI enters module disable mode if an external controller asserts stop when Doze mode is enabled (MCR[DOZE] = 1) or when you write 1 to MCR[MDIS].

41.3.2 Serial flash memory access schemes

In the individual flash memory mode, all supported commands are available.

41.3.3 Normal mode overview

In normal mode, XSPI communicates with an external serial flash memory device through 16 high-data rate bidirectional data lines. It uses the programmable sequence engine to send sequences of instructions, along with optional address, mode, dummy, and data operands, to the memory device. You program these sequences in the lookup table (LUT). Together, the engine and the LUT provide the flexibility to support memory devices from various flash memory vendors by being able to accommodate a wide variety of command protocols.

For additional information, see:

- Programmable sequence engine
- Programmable sequence engine instructions
- Lookup table (LUT)

41.3.3.1 Programmable sequence engine

The programmable sequence engine uses sequences of instruction-operand pairs stored in the LUT. As its name implies, the engine executes the instructions in sequence.

The programmable sequence engine allows you to configure the XSPI module according to the serial flash memory connected on board. This flexible structure is compatible with new command or protocol changes from different vendors.

41.3.3.2 Programmable sequence engine instructions

This tables lists and describes the instructions supported by the programmable sequence engine and their operands.

Table 330. Programmable sequence engine instructions

Command	Instruction (INSTR*n*)	Pads (PAD*n*)	Operand (OPRND*n*)	Action
CMD	000001b (1h)	00b = 1 pad 01b = 2 pads 10b = 4 pads	8-bit command value	Provide the SFM command operand (encoded) on the specified number of pads in Single transfer rate (STR) mode.

Table continues on the next page...

Table 330. Programmable sequence engine instructions (continued)

Command	Instruction (INSTR*n*)	Pads (PAD*n*)	Operand (OPRND*n*)	Action
ADDR	000010b (2h)	11b = 8 pads[1]	Number of address bits	Provide the address of the transaction, using the necessary number of address cycles based on the operand value and the specified number of pads. For example, 24 bits on 4 pads takes 6 cycles in SDR mode or 3 cycles in DDR mode. For AHB-initiated transactions, the address is the actual address of the incoming transaction. For IPS-initiated transactions, the address comes from SFAR. If you specify a column address space value (SFACR[CAS] > 0), then XSPI takes CAS into consideration.
DUMMY	000011b (03h)		Number of dummy clock cycles (from 2 to 64)	Provide the specified number of dummy cycles. PAD*n* defines the number of pads in input mode. For example, 00b (1 pad) means that pad 1 is not driven, but all the rest are driven. See Dummy cycles generation. **NOTE** If DLL is enabled (DLLCR*x*[DLLEN] = 1) and you need *n* dummy cycles, you must program two back-to-back (split) DUMMY instructions: DUMMY with *n* – 2 cycles, followed by DUMMY with 2 cycles.
MODE	000100b (04h)		8-bit mode value	Provide an 8-bit operand on the specified number of pads.
MODE2	000101b (05h)	00b = 1 pad 01b = 2 pads	2-bit mode value	Provide a 2-bit operand on the specified number of pads.[2]
MODE4	000110b (06h)	00b = 1 pad 01b = 2 pads 10b = 4 pads	4-bit mode value	Provide a 4-bit operand on the specified number of pads. [3]

Table continues on the next page...

Table 330. Programmable sequence engine instructions (continued)

Command	Instruction (INSTR*n*)	Pads (PAD*n*)	Operand (OPRND*n*)	Action
READ	000111b (07h)	00b = 1 pad 01b = 2 pads 10b = 4 pads 11b = 8 pads	Number of data bytes to read. This value must be a multiple of 8.	Read the specified number of bytes on the specified number of pads. You can overwrite the data size: • AHB-initiated transactions: BUF*n*CR[ADATSZ] • IPS-initiated transactions: IPCR[IDATSZ]
WRITE	001000b (08h)		Number of data bytes to write	Write the specified number of bytes on the specified number of pads. You can overwrite the data size (IPCR[IDATSZ]).
JMP_ON_CS[4]	001001b (09h)	NA	Instruction number	Every time CS deasserts, jump to the sequence instruction pointed to by the operand. This instruction allows you to specify XSPI behavior on initiation of a new read transaction following a CS deassertion. **NOTE** If DLL is enabled (DLLCR*x*[DLLEN] = 1), XSPI does not support this instruction.
ADDR_DDR	001010b (0Ah)	00b = 1 pad 01b = 2 pads 10b = 4 pads 11b = 8 pads	Number of address bits to send	Provide the address of the transaction of the specified width using on the specified number of pads using each clock edge of the memory clock. For example: a 24-bit address on 4 pads requires 3 cycles. For AHB-initiated transactions, the address is the actual address of the incoming transaction. For IPS-initiated transactions, the address comes from SFAR. If you specify a column address space value (SFACR[CAS] > 0), then XSPI takes CAS into consideration.
MODE_DDR	001011b (0Bh)		8-bit mode value	Provide the 8-bit operand on the specified number of pads using each clock edge of the memory clock. For example: with 4 pads, an 8-bit operand requires 1 cycle; with 2 pads, an 8-bit operand takes 2 cycles.

Table continues on the next page...

Table 330. Programmable sequence engine instructions (continued)

Command	Instruction (INSTR*n*)	Pads (PAD*n*)	Operand (OPRND*n*)	Action
MODE2_DDR	001100b (0Ch)	00b = 1 pad	2-bit mode value	Provide the 2-bit operand on the specified number of pads using each clock edge of the memory clock.[5]
MODE4_DDR	001101b (0Dh)	00b = 1 pad 01b = 2 pads	4-bit mode value	Provide the 4-bit operand on the specified number of pads using each clock edge of the memory clock.[6].
READ_DDR	001110b (0Eh)	00b = 1 pad 01b = 2 pads 10b = 4 pads 11b = 8 pads	Number of data bytes to read. This value must be a multiple of 8.	Read the specified number of bytes on the specified number of pads using each clock edge of the memory clock. You can overwrite the data size: • AHB-initiated transactions: BUF*n*CR[ADATSZ] • IPS-initiated transactions: IPCR[IDATSZ][7]
WRITE_DDR	001111b (0Fh)		Number of data bytes to write. This value must be a multiple of 8.	Write the specified number of bytes on the specified number of pads using each clock edge of the memory clock. You can overwrite the data size (IPCR[IDATSZ][7]).
DATA_LEARN	010000b (10h)	00b = 1 pad 01b = 2 pads 10b = 4 pads 11b = 8 pads	8-bit data learning pattern Operand bits 6–0: • Number of bits to use for the data learning pattern. For example, if operand is 16 then a 16-bit data learning pattern is read and compared.	Find the correct sampling point with the data learning pattern. See Data learning This instruction causes XSPI to ignore the sampling configuration (SMPR) and find the correct sampling point by sampling the data learning pattern.[8]
CMD_DDR	010001b (11h)	00b = 1 pad 01b = 2 pads 10b = 4 pads 11b = 8 pads	8-bit command value	Provide the SFM command operand (encoded) to memory with the SFM command operand (ncoded) on the specified number of pads using each clock edge of the memory clock (DTR mode).
CADDR	010010b (12h)		Number of column address bits to send	Provide the specified number of column address bits on the specified number of pads. The actual address depends on SFACR[CAS]. For example, if SFACR[CAS] is 3, then the address to memory is bits 2–0 of the incoming address for AHB-initiated transactions and SFAR for IP-initiated transactions. XSPI appends 0

Table continues on the next page...

Table 330. Programmable sequence engine instructions (continued)

Command	Instruction (INSTR*n*)	Pads (PAD*n*)	Operand (OPRND*n*)	Action
				if SFACR[CAS] is less than the specified number of pads.
CADDR_DDR	010011b (13h)		Number of column address bits to be sent	Provide the specified number of column address bits on the specified number of pads using each clock edge of the memory clock. The actual address to be provided to memory depends on the value of SFACR[CAS]. For example, if CAS is 3, then the address to memory is [2:0] of the incoming address for AHB and the value of SFAR for IP. This is appended with 0 if CAS is less than the number of pads for memory.
JMP_TO_SEQ[4]	010100b (14h)	NA	Sequence number	After CS deassertion of initial sequence, jump to the LUT sequence pointed to by the operand. This instruction allows the programmer to join two sequences and initiates the second flash memory transaction automatically. It can be used to join write-enables with flash memory write or data learn with read. **NOTE** If DLL is enabled (DLLCR*x*[DLLEN] = 1), XSPI does not support this instruction.
STOP[4]	000000b (00h)	NA	NA	Stop execution; deassert CS. **NOTE** Unused LUT locations need to be filled with STOP instruction.

1. For x16 mode same 8 pads configuration is used.
2. For a one-pad instruction, MODE2 takes two cycles on the memory device interface.
3. For a one-pad instruction, MODE4 takes four cycles on the memory interface. For a four-pad instruction, MODE4 takes one cycle on the memory interface.
4. A sequence ending with this instruction must have all remaining bits as 0s after it.
5. For a one-pad instruction, MODE2_DDR takes one memory clock cycle on the memory interface.
6. For a one-pad instruction, MODE4_DDR takes two memory clock cycles on the memory interface. For a four-pad instruction, MODE4_DDR takes half a cycle on the memory interface.
7. In DDR mode, XSPI supports a minimum IDATSZ of 2 bytes. Specifying IDATSZ as 1 byte imposes a half-cycle sampling limitation, that causes assertion of the CS.
8. Do not use 0h or FFh as the data learning pattern.

41.3.3.3 Lookup table (LUT)

The LUT contains storage for 16 program sequences. Each sequence can have a maximum of 5 instruction-operand pairs (10 instruction-operands), which when executed sequentially by the programmable sequence engine, generate valid memory transactions. You program these instruction sequences into their corresponding LUT*n* registers. See LUT and sequence structure.

At reset, sequence 0 (LUT0–LUT4) contains the basic read sequence described in Basic read sequence. After reset, you can reprogram the complete LUT to support the on-board memory device or devices. To protect its contents in the event of a code run over, you can lock the LUT, which prevents writing to the LUT until it has been unlocked again.

For additional information, see:

- Locking the LUT
- Unlocking the LUT
- Example LUT sequences

41.3.3.4 LUT and sequence structure

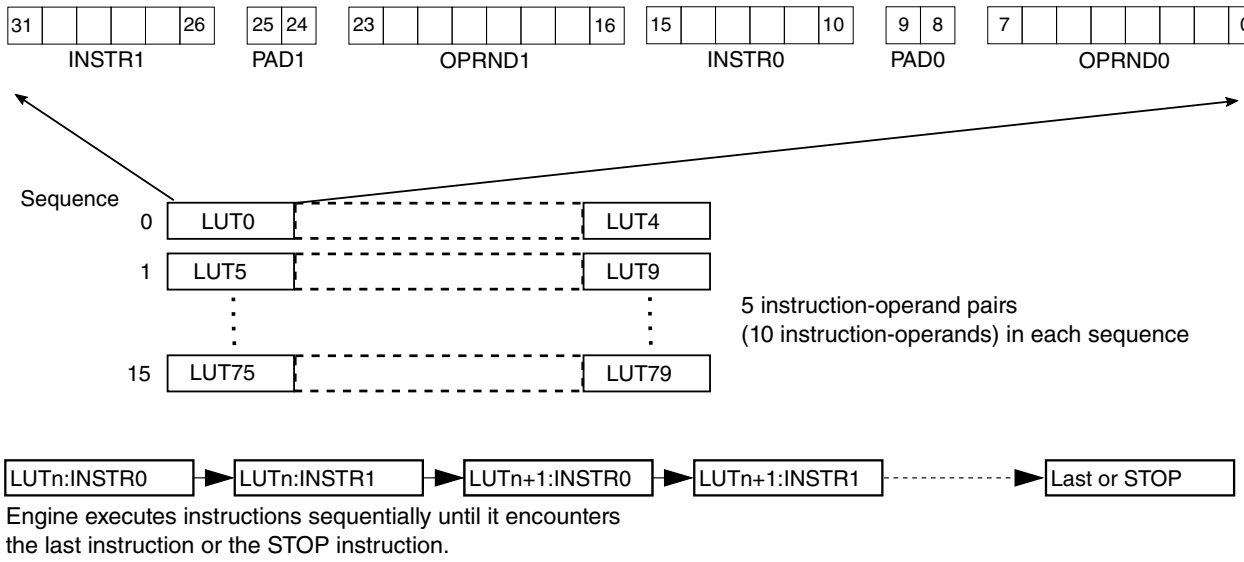


Figure 144. LUT and sequence structure

NOTE

For AHB and IPS transactions, the LUT OPRNDn value for WRITE, WRITE_DDR, READ, and READ_DDR instructions must be 0000_1000b (8 bytes).

41.3.3.5 Basic read sequence

After reset, instruction sequence 0 contains this default read sequence.

Table 331. Basic read sequence

Sequence	Instruction	Pads	Operand	Description
LUT0 0	CMD	00b	0000_0011 b (03h)	Read data byte command on one pad.

Table continues on the next page...

Table 331. Basic read sequence (continued)

Sequence	Instruction	Pads	Operand	Description
LUT0 1	ADDR	00b	0001_1000 b (18h)	Send 24 address bits on one pad.
LUT1 0	READ	00b	0000_1000 b (8h)	Read 64 bits.
LUT1 1	JMP_ON_CS	00b	0000_0000 b (0h)	Jump to instruction 0 (CMD).

41.3.3.6 Locking the LUT

The key for locking or unlocking the LUT is 5AF05AF0h. To lock the LUT:

Table 332. Locking the LUT

Step	Purpose	Programming	Notes
1	Insert the key.	Write 5AF05AF0h to LUTKEY[KEY].	
2	Lock the LUT. You must perform this step immediately after step 1. If any IPS transaction occurs between these steps, the lock action fails.	Write 01b to LCKCR.	A successful write to this register locks the LUT. Write access to LCKCR is disabled when the global configuration lock is enabled (MGC[GCLCK] != 00b).
3	Check the lock status.	Read LCKCR[LOCK].	If LCKCR[LOCK] is 1, LUT is locked.

41.3.3.7 Unlocking the LUT

The key for locking or unlocking the LUT is 5AF05AF0h. To unlock the LUT:

Table 333. Unlocking the LUT

Step	Purpose	Programming	Notes
1	Insert the key.	Write 5AF05AF0h to LUTKEY[KEY].	
2	Unlock the LUT. You must perform this step immediately after step 1. If any IPS transaction occurs between these steps, the lock action fails.	Write 10b to LCKCR.	A successful write to this register unlocks the LUT. Write access to LCKCR is disabled when the global configuration lock is enabled (MGC[GCLCK] != 00b).
3	Check the unlock status.	Read LCKCR[UNLOCK].	If LCKCR[UNLOCK] is 1, LUT is unlocked.

41.3.3.8 Issuing SFM commands

Each access to the external device follows this sequence:

1. You must pre-populate the LUT with the serial flash memory command sequences that are required for the flash memory device being used.
2. The module executes the instructions in this sequence one by one. The transaction starts and the module sets the value of SR[BUSY].

3. Communication with the external serial flash memory device starts and the transaction executes.
4. After the transaction is complete (all transmit and receive operations with the external serial flash memory device are complete), the module resets SR[BUSY]. In case of an IP command, FR[TFF] is asserted.

You can trigger the processing of SFM commands in the XSPI module in one of the following ways:

Using IP commands

For details, see Flash memory programming using IP command and Flash memory read using IP command.

Using AHB commands

For details, see AHB read and AHB write.

41.3.3.9 Flash memory programming using IP command

In all NOR Flash devices memory sector to be written needs to be erased first. The programming sequence is then initiated in the following way:

1. When SFP block is present, the IP command has to be targeted for a specific queue (TG) and based on this the registers have to be selected.
2. Check that TG queue that is selected for transfer is empty (TGSFARS[VLD] or TGSFARS_SUB_1[VLD] is 0).
3. Write the serial flash memory address to the Serial Flash Memory Address (SFP_TG_SFAR) or Serial Flash Memory Address (SFP_TG_SUB_1_SFAR) for that specific queue. For IP commands not related to specific addresses, the base address of the related flash memory needs to be programmed.
4. Write the sequence ID and data size details in the IP Configuration (SFP_TG_IPCR) or IP Configuration (SFP_TG_SUB_1_IPCR) depending on the queue selected. Maximum value of IPCR[IDATSZ(data size) is equal to the TX buffer size (1 KB) of XSPI.
5. After writing the SFP_TG_IPCR[SEQID] and SFP_TG_SUB_1_IPCR[SEQID], the queue is locked and if SFP checks are passed then transfer request is send to XSPI arbiter.
6. Wait for arbitration to be granted. In case of flash memory write transfers after the arbitration has been granted - clear the TX buffer, if it has older data present by writing 1 to MCR[CLR_TXF] field. Then program the correct TBCT[WMRK] and fill the TX data for the program command into the circular buffer through the TBDR.
7. After the TX data has been filled and TBCT[WMRK] is triggered, the SR[TXWA] flag goes low and transaction is triggered to XSPI which starts executing it.
8. The TBDR register access remains enabled for the queue which has the arbitration granted.
9. SR[TXFULL] and TBSR[TRBFL] can be used to check if the buffer is ready to receive more data. There will be few-cycle's latency before the status is reflected in TBSR[TRBFL] and SR[TXFULL]. TBSR[TRCTR] can be read to check how many words have been written into the TX buffer.
10. SR[BUSY] flag, SR[ARB_STATE] and TGSFARS[VLD] or TGSFARS_SUB_1[VLD] can be checked to know when the transfer is completed.

The software ensures that the correct sequence is programmed into the LUT in accordance with the flash memory connected to the module. The data is fetched from the TX buffer. It consists of 256 entries of 32-bit sizes and is organized as a circular FIFO, the read pointer for which is incremented after each fetch. When all the data is transmitted, the XSPI module returns from the busy state to the idle state. However, this is not true for the external device because the internal programming is still ongoing. You may monitor the relevant status information available from the serial flash memory device and ensure that the programming is done appropriately.

The application must define the IDATSZ value in alignment with the data programming requirements of the TBDR register, which is 4-byte aligned. Initiating a configuration write transaction on flash memory with a data size less than 32 bits results in the removal of one data entry from the TX buffer. In such cases, only the valid bits are utilized, while the remaining bits are discarded.

NOTE

See Secure flash protection for details on TG queues and related registers.

41.3.3.9.1 TX buffer operation

The TX buffer provides the data used for page programming through IP command or AHB interface. When TBCT[WMRK] is greater than or equal to the payload size, for proper operation it is required to write the entire payload data in TX buffer prior to starting execution of the page programming command. When TBCT[WMRK] is less than payload size, it is required to provide at least one entry in the TX buffer prior to starting the execution of the page programming command. The application must ensure that the required number of data bytes is written into the TX buffer fast enough as long as the command is executed without a TX buffer overflow or underrun.

The XSPI module sets the FR[TBFF] field as long as the TX buffer is not full and can accept more data. At the end of write through TX buffer, you must clear FR[TBFF] to avoid unnecessary last TX buffer fill interrupt. However, there would always be a pending request asserted from XSPI controller at the end of any DMA transfer. When external DMA finishes transfer iteration, this request from XSPI is kept asserted for the next iteration loop.

NOTE

Even if the generation of DMA requests for filling the TX buffer is disabled by using RSER[TBFDE], the TX buffer still accepts a DMA transfer because of the last asserted pending request.

Disabling of DMA transfer should be controlled by an external DMA initiator.

When the XSPI module tries to pull data out of an empty TX buffer, FR[TBUF] signals the TX buffer underrun. The current IP command leading to the underrun condition is aborted after the available data has been programmed and SW needs to re-execute the entire transaction. SR[TXFULL] and TBSR[TRBFL] can be used to check if the buffer is not full and can receive more data. When this sequence command is complete, FR[TBFF] is asserted, indicating that the TX buffer is ready to be written again. In case of underrun, application needs to execute the transaction again.

The TX buffer overflow is not signaled explicitly, but SR[TXFULL] and TBSR[TRBFL] can be used to monitor if the buffer is not full and can receive more data.

For more information, see TX Buffer Status (TBSR) and Flag (FR).

41.3.3.9.2 TX underrun handling by hardware

1. When underrun event is detected at flash side due to slow data rate feeding or incorrect watermark setting, hardware flushes the residue data towards flash by initiating another transaction for residue data.
2. Set AWRCR[PPW_WR_DIS] = 1b for flash devices that require program wait period to enable the auto RDSR mechanism. See Page program wait handling for configuration details.
3. When transaction detects underrun event, hardware stops the current transfer and initiates a new transfer to flush the residue data either immediately (PSRAM/HyperRAM devices) or after waiting for Tpp (NOR flash devices).
4. Step 4 is repeated until all the residue data is transferred until SFP timeout.
5. FR[TBUF] is generated in case the required data is not received from the application within SFP timeout period.

41.3.3.10 Flash memory read using IP command

Host access to the data stored in the external serial flash memory device is performed in two steps. First, the data must be read into the internal buffers and in the second step, these internal buffers can be read by the host.

Reading serial flash memory data into the XSPI module internal buffers

A read access to the external serial flash memory device can be triggered through IP command as given below:

1. For reading flash memory data into the RX buffer select the TG queue through which transfer has to be started.
2. Check that TG queue that is selected for transfer is empty (TGSFARS[VLD] or TGSFARS_SUB_1[VLD] field is 0).
3. Write the serial flash memory address to the Serial Flash Memory Address (SFP_TG_SFAR) or Serial Flash Memory Address (SFP_TG_SUB_1_SFAR) for that specific queue.

4. Write the sequence ID and data size details in the IP Configuration (SFP_TG_IPCR) or IP Configuration (SFP_TG_SUB_1_IPCR) depending on the queue selected. Read data size shall be limited to the RX Buffer size. You must provide the correct sequence ID in SFP_TG_IPCR[SEQID]. The sequence ID points to a sequence in the LUT. The software needs to ensure that a correct read sequence is programmed in the LUT in accordance with the serial flash memory device connected on board.
5. After writing the IPCR[SEQID], the queue is locked and transfer request is sent to the XSPI arbiter. Wait for arbitration to be granted.

All available read commands supported by the external serial flash memory are possible. Processing of AHB buffer — hit scenario is parallel to ongoing IP read command. AHB read request is served from buffered data based on buffer hit scenario.

Optionally, it is possible to clear the RX buffer pointer prior to triggering the IP command by writing a 1 to MCR[CLR_RXF]. This invalidates the data currently present in the RX buffer and any new read data overwrites the old data. The transaction is not triggered to XSPI if there is some older data pending in the RX buffer. Using these inputs, the complete transaction is built when arbitration is granted for this queue. The transaction related to the read access starts and the requested number of bytes is fetched from the external serial flash memory device into the RX buffer. As the read access is triggered by an IP command, the value of both SR[IP_ACC] and SR[BUSY] is set to 1. A count of the number of entries currently in the Rx buffer can be obtained from RBSR[RDBFL].

NOTE

See Secure flash protection for details on TG queues and related registers.

Communication with the external serial flash memory stops if the specified number of bytes are read (on successful completion of the transaction).

NOTE

In case of external DQS strobe based sampling, read data size (IDATSZ or LUT size) must be in multiples of 8 bytes.

41.3.3.10.1 RX buffer push event

RX buffer push event is generated automatically whenever there is data available in RX Async FIFO and space available in RX buffer.

By default, each buffer push event adds two entries to the RX buffer because the interface to the serial clock domain is 64 bits in width. Depending on the number of bytes read from the serial flash memory device, it is possible for the very last buffer push event that only one entry is added.

RBSR[RDBFL] is incremented by the number of entries added to the RX buffer.

41.3.3.10.2 RX buffer POP event

RX buffer pop event needs to be generated after reading RBCT[WMRK] + 1 entries through RBDR0 - RBDR63.

Each buffer POP event removes (RBCT[WMRK] + 1) valid entries from the buffer. RBSR[RDBFL] is decremented by the same number and RBSR[RDCTR] is incremented accordingly.

41.3.3.10.3 Data transfer from the XSPI module internal buffers

The data read out from the external serial flash memory device by the XSPI module is stored in the internal buffers. The means of accessing the data from the buffer differs depending on which buffer the data is loaded to. See Block diagram for details on the two available buffers, the RX buffer and the AHB buffer, in this module. The buffer appears transparent to you and is non-memory mapped. See the "Flexible AHB Buffer" section for details.

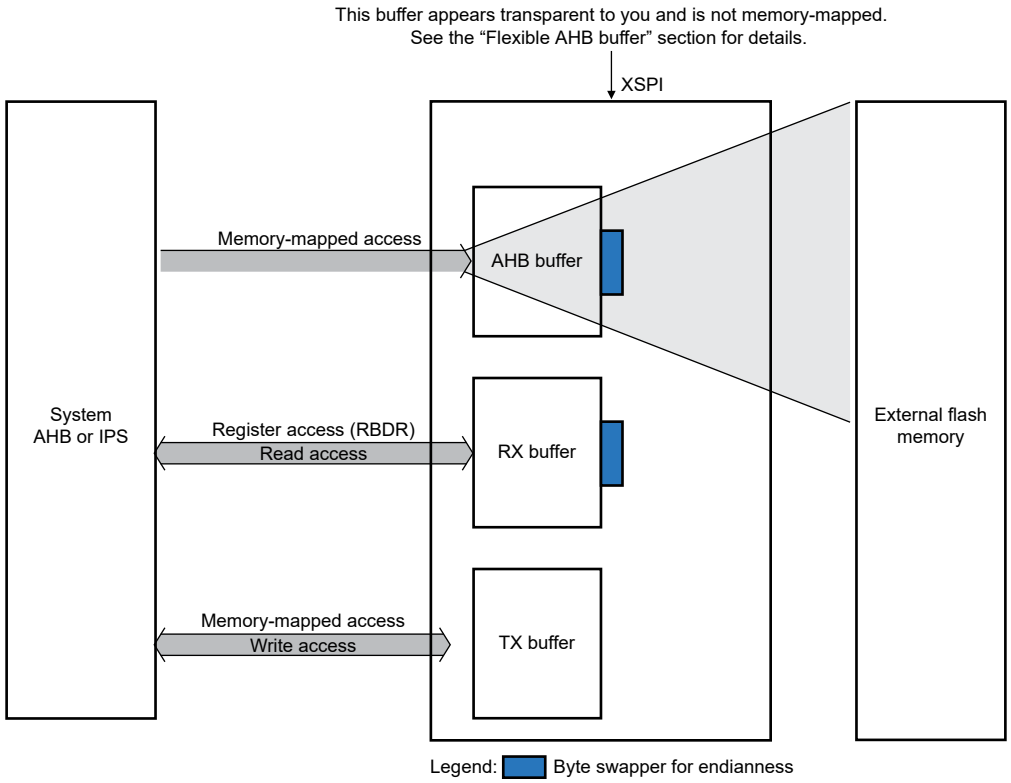


Figure 145. XSPI memory map

The RX buffer is implemented as FIFO of 32-bits width. Its content is accessible in the IPS address space in the area associated with RX Buffer Data (RBDR0 - RBDR63).

The RX buffer operation can be summarized as follows:

- RBCT[WMRK] determines at which fill level SR[RXWE] is asserted and how many entries are removed from the RX buffer on each buffer POP operation.
- SR[RXWE] indicates that the configured number of data entries is available in the RX buffer and RBSR[RDBFL] indicates how many valid entries are available in total.
- The first entry (RBDR0) always corresponds to the first valid entry in the RX buffer.

For details, see RX Buffer Data (RBDR0 - RBDR63).

- Flag-based data read of the RX buffer is performed by polling SR[RXWE]. When it is asserted, the valid entries can be read via the IPS address space (RBDRn). A buffer POP operation must be triggered by the application by writing a 1 to FR[RBDF]. This automatically updates the FIFO to point to the next entry as defined by RBCT[WMRK]. For example, if WMRK is set to 3, then the buffer discards 16 bytes of data.
- Internal DMA-controlled data read of the RX buffer is performed by using the DMA module. The application must ensure that the DMA controller of the related chip is programmed appropriately, as described in DMA.
- Internal DMA-controlled read out is triggered fully automatically by the assertion of SR[RXWE]. The related buffer POP operation is also handled completely inside the XSPI module. As in the case explained here, accessing the RX buffer content on RBDRn related addresses is equivalent.
- AHB buffer data read via memory-mapped access: This kind of access is performed by reading one of the addresses assigned to the external serial flash memory device(s) within the range specified in Table 406. If this is not the case, a memory-mapped AHB command read is triggered as described above. If the requested data is already available in the AHB buffer, it is provided directly to the host.

41.3.3.11 ARDB read

41.3.3.11.1 RX buffer read through ARDB registers

41.3.3.12 AHB read

For reading flash memory data into the AHB buffer, you must:

- Set up a read access by a initiator to the address range in the system memory map, which the external serial flash memory devices are mapped to.
- Write a desired value to SFACR[CAS], if required, or write 0 to the field.
- Program the buffer registers corresponding to the AHB initiator initiating the request.
- Provide the correct sequence ID in the BFGENCR. The software ensures that a correct read sequence is programmed in the LUT in accordance with the serial flash memory device connected on board. Flash memory device selection and access mode are determined by the address accessed in the AHB address space associated with the XSPI module (see Memory-mapped serial flash memory data—individual flash memory mode on flash memory A

On each AHB read access to the memory mapped area, the valid data in the AHB buffer is checked against the address requested in the actual read. When the AHB read request cannot be served (buffer-miss) from the content of the AHB buffer, the buffer is flushed and the controller executes the sequence pointed to by the sequence ID. The requested number of buffer entries defined in BUFxCR[ADATSZ] is then fetched from the external serial flash memory device into the internal AHB buffer. As the read access is triggered through the AHB miss condition, the value of SR[AHB_ACC] is set, driving SR[BUSY] in turn, until the transaction is complete. Communication with the external serial flash memory stops when the specified number of entries is filled.

When an AHB access is made to the flash memory mapped address, the data is fetched and returned to the AHB interface. The AHB interface is stalled (HREADY low) until the data is fetched. As soon as the data from the requested address is read by the XSPI module, the AHB read access is served. Therefore, it is possible to run sequential reads from the AHB buffer at arbitrary speed without the need to monitor any information about the availability of the data. Nevertheless, this access scheme stalls the AHB bus for the time required to read the data from the serial flash memory device. If you know that the access is sequential, a better way is to have a prefetch enabled by programming the value of BUFxCR[ADATSZ] so that the data is fetched into the buffer before the next sequential AHB access.

As long as the host restricts its accesses to the data present in the buffer and to the data currently fetched from the serial flash memory, it is possible to run the host read from the AHB buffer simultaneously with the serial flash memory read into the AHB buffer.

NOTE

AHB HRESP error is generated in case the AHB read address received is out of the valid flash address range programmed with XSPI. To avoid the timeout in this case of illegal access, software abort should be done by using the SPTRCLR[ABRT_CLR] bit.

41.3.3.12.1 Flexible read xAHB buffers

To reduce the latency of the reads for AHB initiators, the data read from serial flash memory is buffered in flexible AHB buffers. There are four such flexible buffers. The size of each of these buffers is configurable with the minimum size being 0 bytes and maximum size being the size of the complete buffer instantiated (4096 bytes). Each 1st–level buffer size can be configured in range of 0 KB, 512 B, 1 KB, 2 KB, 4 KB. Each of these buffers can be further sub-divided into 4 sub-buffers.

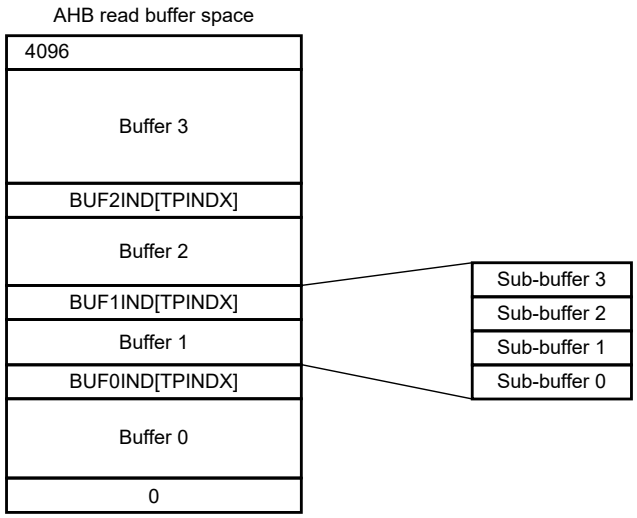


Figure 146. Flexible read AHB buffers

For any AHB access, the sequence pointed to by BFGENCR[SEQID] is used for the initiated flash memory transaction. BFGENCR[SEQID] points to an index of the LUT. See LUT for details.

Each 1st–level buffer size can be configured in range of 0 KB, 512 B, 1 KB, 2 KB, 4 KB. The size of buffer 0 is the difference between Buffer 0 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)) and 0. The size of buffer 1 is the difference between Buffer 1 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)) and Buffer 0 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)). The size of buffer2 is the difference between Buffer 2 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)) and Buffer 1 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)), buffer 3 is the difference between the size of the complete buffer (4096 bytes and Buffer 2 Top Index (Buffer n Top Index (BUF0IND - BUF2IND)).

NOTE

AHB initiator assignment is strictly prohibited for the AHB buffers those are allocated with a depth of zero, as this constitutes an invalid AHB buffer configuration.

The sub-buffer division for each buffer can be enabled by writing '1' to the respective BUFnCR[SUB_DIV_EN] field. Size of sub-buffer is decided based on the division factor from total buffer size. This division factor is programmed using respective BUFnCR[SUBBUF0_DIV], BUFnCR[SUBBUF1_DIV], and BUFnCR[SUBBUF2_DIV] fields. XSPI supports division factors of /2, /4, /8 and /16. Size of Sub-buffer 3 is decided based on the left over space from three sub-buffers (0-2).

For example: if TPINDX0 field in BUF0IND register is programmed as 8. Then the size of buffer 0 becomes 8*64 bits. Then size of sub buffer can be decided based on division factors programmed as given in table below:

Table 334. Sub-buffer division factors

Sub-buffer number	SUBBUF*n*_DIV	Sub-Buffer division factor	Sub-buffer size
Sub Buffer 0	001	/2	4*64 bits
Sub Buffer 1	010	/4	2*64 bits
Sub Buffer 2	011	/8	1*64 bits
Sub Buffer 3	-	Remaining size	1*64 bits

Ensure that buffer size is always programmed in multiples of 2 when sub-buffer division is enabled.

Each flexible AHB buffer is associated with the following:

- An AHB initiator: A single AHB initiator can be assigned to a buffer by programming the respective BUFnCR[MSTRID] field. Buffer3 may be configured as an "all initiator" buffer by writing 1 to BUF3CR[ALLMST]. When buffer3 is configured in such a way, any access from a initiator not associated with any other buffers (0-2) is routed to buffer3.
- A data size field representing the amount of data to be fetched from the flash memory on every missed access.

The initiator ID of every incoming AHB request is checked and the data is returned or fetched into the corresponding buffer. If the address being requested by AHB initiator is already present inside the buffer/sub-buffer it is called as buffer-hit. If the address is not already present or not same as the address currently being read/prefetched from flash it is called as buffer-miss. In case of buffer hit the data is returned from the buffer and no access is generated towards the serial flash memory. Every missed access to the buffer causes the controller to clear the buffer and fetch the BUFxCR[ADATSZ] amount of data from the serial flash memory. As such, you need not configure the buffer size to be greater than BUFxCR[ADATSZ] because the locations greater than BUFxCR[ADATSZ] are never used. The data is returned to the initiator as soon as the requested address is read from the serial flash memory. The controller, however, continues to prefetch the rest of the data in anticipation of a next consecutive request.

In case BUF3CR[ALLMST] field is set to '0' and incoming AHB read request initiator ID is not mapped to any of the programmed buffers, then controller will keep the AHB HREADY low and will provide pre-defined AHB error payload (AHB Error Payload High (AHB_ERR_PAYLOAD_HI)and AHB Error Payload Low (AHB_ERR_PAYLOAD_LO)) on the read data bus when the HREADY timeout counter expires and FR[AITEF] flag is set.

41.3.3.12.1.1 Buffer function with sub-buffer division enabled

When sub-buffer division is also enabled for a buffer it places additional constraints for incoming AHB transaction to be mapped to a sub-buffer. Each sub-buffer is associated with an address range which can be programmed in AHB Buffer n Sub Buffer m Start and End Address Range (AHB Buffer n Sub Buffer m Start and End Address Range (BUF0_0_ADDR_RANGE - BUF3_3_ADDR_RANGE)) registers. Address ranges will be aligned in unit of 64 KB. When an AHB read access comes it is assigned to a buffer based on its initiator ID. Then it is further assigned to a sub-buffer based on which sub-buffer address range this transaction lies into. When sub-buffer division is enabled for a buffer the ADATSZ field is ignored. In case of a buffer miss controller will fetch the data equal to the size of that sub-buffer. This address range mapping constraint is not applicable when sub-buffer division is disabled for a buffer. Buffer overflow will happen and FR[ABOF] flag is set, due to incorrect programming of the End address, and the next buffer contents are modified.

It should be ensured while programming that no two address range of sub-buffers within same buffer are overlapping. The required configuration registers for AHB buffer programming should be programmed before the transaction starts and should not be changed during runtime.

In case the incoming address from an AHB initiator does not lie within any of the sub-buffer's address range of that buffer or matches within multiple sub-buffer address ranges of that buffer, an error field (FR[RDADDR]) is set and interrupt is generated if enabled for that error field. The address and AHB initiator ID which generated this AHB transfer is latched in AHB Read Error Address (AHB Read Error Address (AHB_RD_ERR_ADDR)) and AHB Read Error Initiator ID (AHB Read Error Initiator ID (AHB_RD_ERR_MID)) fields respectively. On AHB interface no error response is generated for this transaction. And the AHB error payload (AHB Error Payload High (AHB_ERR_PAYLOAD_HI) , AHB Error Payload Low (AHB_ERR_PAYLOAD_LO)) is sent as the read data for that transaction. In case of multiple errors the error status registers will keep the first error information latched unless the error field (FR[RDADDR]) is cleared in Flag register by writing '1' to this field.

41.3.3.12.2 Abort mechanism during AHB read

When sub-buffer division is disabled the buffer hit/miss condition is checked at individual buffer level. When sub-buffer division is enabled for a buffer the hit/miss check is done at sub-buffer level.

Any ongoing read transaction (prefetch) on flash is aborted if a request from the same initiator (which started the pre-fetch) comes and results in a buffer miss. The abort can happen at any point of time. After the abort completes, the controller generates a new read transfer to flash with the address currently being requested by the AHB initiator. All the data prefetched in the buffer is discarded and new prefetch is started with the current AHB address.

In case sub-buffer division is enabled the prefetch takes place for individual sub-buffers. If the AHB initiators requests an address which is already present inside the sub-buffer, the controller gives the data from this sub-buffer. In case the data for requested address is not present inside the sub-buffer and it results in a buffer-miss, In this case if the prefetch was in progress on the same sub-buffer on which the requested address lies then the prefetch is aborted. Once the abort completes the data inside that

sub-buffer is discarded and new data starting from the current AHB address is prefetched. The content of other sub-buffers are not changed.

If the prefetch was in progress for some sub-buffer and AHB initiator requests an address which results in buffer-miss on some other sub-buffer of the same initiator then the controller will wait for prefetch to complete first. Once the prefetch for this sub-buffer completes a new read transaction will be generated on flash with current AHB address being requested and the data will be written into the other sub-buffer on which buffer-miss occurred.

An ongoing prefetch is also aborted in case the start address + data prefetched so far reaches the end-address boundary set for that sub-buffer. It ensures that two sub-buffers of a single buffer do not contain the data for same address.

These abort conditions are not treated as error and no flag/status is set for these aborts.

The ongoing prefetch transaction can be aborted in case software abort is requested by using the SPTRCLR[ABRT_CLR] bit. It will stop the ongoing read/prefetch from flash and invalidates all the data already present inside the buffers. The software abort during AHB read may result in FR[AITEF] flag generation, if the read is not initiated towards flash controller. This software abort may also result in FR[AAEF] flag being set in case any AHB read transfer was ongoing during software abort. Any ongoing prefetch is also aborted in case any AHB write request arrives while the prefetch is ongoing. But in this case the data already present inside the buffers is not invalidated when prefetch is enabled and no error flag is set. Any ongoing prefetch is also aborted in case any AHB write request arrives while the prefetch is ongoing.

41.3.3.12.3 Suspend-resume mechanism

AHB initiators are categorized into high priority or normal priority categories. Any low-priority AHB read access (prefetch) can be suspended by a high-priority AHB initiator request which results in a buffer miss. In case a higher priority initiator request results in a buffer-hit the data is provided from the buffer and it will not suspend any ongoing prefetch. The ongoing transaction is always suspended at 64-bit boundary. The suspended transaction is resumed after the high priority initiator is served and the high priority transaction including data prefetch is completed.

Buffer0 or Buffer 1 can be configured to be a high-priority initiator by writing 1 to BUF0CR[HP_EN] and BUF1CR[HP_EN] bits. There can be only one suspended transaction at a time. The status of suspended transaction can be read from Sequence Suspend Status (SPNDST). Suspended address is stored in Sequence Suspend Status (SPNDST). If the prefetch was in progress in any high priority buffer and AHB read request arrives for another high priority buffer with a buffer miss, it will wait for the prefetch to complete first. High priority initiator can not suspend the ongoing prefetch of another high priority initiator. After the high priority requests are served, suspended requests are given a priority over other AHB read requests from non-high priority initiators. Controller will first complete the suspended transfer and then service the prefetch request from non-high priority initiators. In case of buffer hit requests, the data is provided from the buffer and it does not affect any ongoing prefetch.

While a transaction is in suspended state, it might get aborted if a read request by the same suspended initiator arrives on same sub-buffer and results in a buffer-miss. In this case the SPNDST[STATE] will change from 01b to 00b without going to 11b. A suspended/resumed prefetch will also be aborted in case of software abort request or if any AHB write request arrives.

A prefetch by a non-high priority initiator can be suspended multiple times in case higher priority requests keep arriving in between with buffer miss. A high priority initiator request can’t suspend the prefetch of another high-priority initiator or prefetch of same initiator on any other sub-buffer.

SPTRCLR[PREFETCH_DIS] field should be set to 0 while using the suspend resume functionality else it may lead to read data mismatch.

41.3.3.12.4 HBURST support with AHB read

XSPI controller supports HBURST and HSIZE as part of AMBA AHB standard interface. HBURST indicates if the transfer forms part of a burst. Four, eight, and sixteen beat bursts are supported and the bursts might either be incrementing or wrapping. Undefined length incremental burst is also supported by XSPI. HSIZE indicates the size of the transfer, and supports 8-, 16-, 32-, and 64-bit data sizes. In case of WRAP accesses, XSPI generates aligned accesses to serial flash memory if there is no buffer hit for any incoming, non-sequential AHB read access. In case there is a buffer hit, the incoming AHB address is latched as it is. If the total burst size is more than the data prefetch size, an error response is generated on the bus and the value of FR[AIBSEF] is set as 1. The data prefetch size can be defined by BUFnCR[ADATSZ] or data size mentioned in the sequence pointed to by the SEQID field when ADATSZ is programmed as 0. In case sub-buffer division is enabled the data prefetch size is defined as

the size of sub-buffer. In case of wrap burst, data prefetch size must be greater than or equal to the wrap burst size + 32 bytes. A few examples are shown in the figure below:

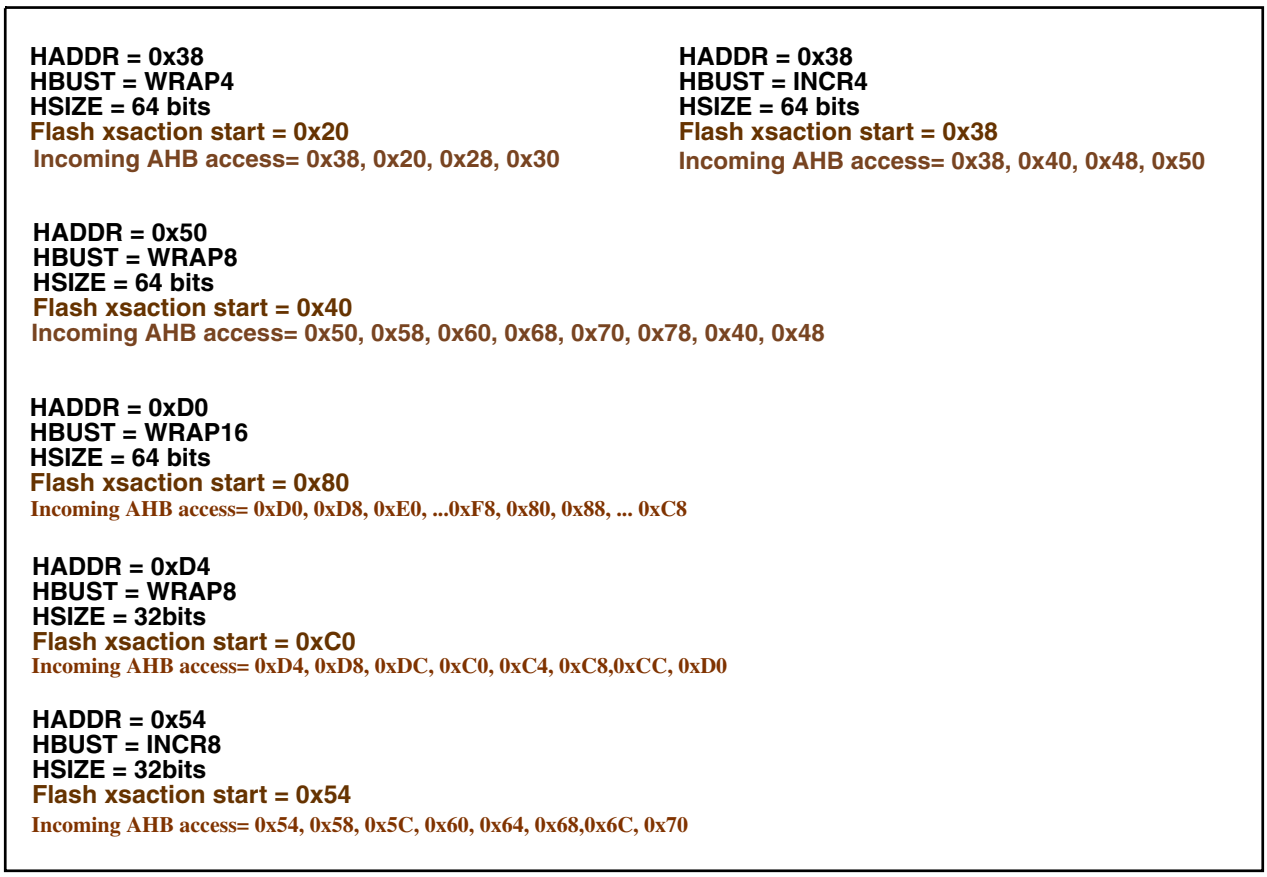


Figure 147. XSPI HBURST support

NOTE

The software must take care that the prefetch size should never be set less than the minimum data needed by any external interface to start processing.

NOTE

Whenever a core accesses XSPI memory with cache enabled, the prefetch size must be configured as equal or more than the cache line size; otherwise, FR[AIBSEF] error appears.

41.3.3.12.5 AHB read buffer status

The status of AHB sub-buffers is captured in AHB Buffer Status Register (AHB Buffer Status (AHB_BUF_STATUS)). Each of the sub-buffer status can take three values depending on which level the buffer is filled after the data prefetch. 00b – Sub-buffer is empty, 01b – Sub-buffer is full or 10b- Sub-buffer is partially full.

Sub-buffer can be empty in case no data prefetch has been started on that buffer since operation start or after clearing of buffers. It can be partially full in case prefetch is ongoing in that sub-buffer or prefetch was aborted in between or if the prefetch is suspended.

Status register fields SR[AHB0FUL, AHB1FUL, AHB2FUL, and AHB3FUL] also shows the individual buffers status as Full or not-empty. In case the sub-buffer division is disabled for that buffer these bits show the full or not-empty status for that buffer. In case sub-buffer division is enabled the buffer full status register bit get set to '1' if any of its sub-buffer is full.

41.3.3.12.6 AHB read performance monitors

XSPI provides a mechanism to track the hit and miss performance of AHB read transfers at every sub-buffer level. AHB Buffer Hit/Miss Performance Monitor Control Register (AHB Buffer Hit/Miss Performance Monitor Control (AHB_PERF_CTRL)) is used for controlling this performance monitoring. XSPI has four 16-bit Hit counters and four 16-bit miss counters which can be used for performance monitoring. Each set of counters is associated with a single AHB buffer partition and can be enabled by writing AHB_PERF_CTRL[BUFn_EN] fields. It can be further selected which sub-buffer the performance counters are restricted to, by using AHB_PERF_CTRL[SUB_BUF_SELn] fields. If the sub-buffer division is not enabled for that buffer irrespective of SUB_BUF_SEL programming it will monitor the accesses for complete buffer. After performing the required configuration in AHB_PERF_CTRL[CNTSTART] field can be written as '1' to start the performance monitoring. These counters work at system (AHB) clock.

Performance monitors have two 16-bit counters one for buffer hit (AHB read request address is found within the AHB buffer and no new read transfer is generated towards the flash or the incoming read request data is already part of prefetch ongoing on flash) and another for buffer miss (AHB read address is not present in the AHB buffers and is not equal to the address currently being prefetched from flash. It will cause a new read transaction to be generated towards flash).

After starting the performance counters the monitoring stops on any of the below conditions:

- If '1' is written to AHB_PERF_CTRL[CNTSTP] bit.
- If any of the buffer hit/miss counters overflow and corresponding overflow bit is set in AHB Buffer Hit/Miss Performance Monitor Control (AHB_PERF_CTRL).
- If the performance time counter overflows.

In case of overflow FR[PERFOVF] field is set in flag register and an interrupt can be generated if enabled in RSER register. The overflow bits in AHB Buffer Hit/Miss Performance Monitor Control (AHB_PERF_CTRL) remain latched unless this flag register bit is cleared by writing '1'. After the monitoring is stopped the values of the counters are updated in AHB Buffer n Performance Monitor (AHB Buffer n Performance Monitor (AHB_PERF_BUF0 - AHB_PERF_BUF3)) registers. In case of overflow the counter value may be shown as 0. The time period for which the performance monitoring is run is latched in AHB Performance Monitor Time Counter (AHB Performance Monitor Time Counter (AHB_PERF_TIME_CNT)). This is a 32-bit counter which is running on AHB clock frequency. These performance counters and time counter are cleared when new monitoring is started.

41.3.3.12.7 Stale data prevention inside AHB RX buffer

When AHB prefetch is enabled the AHB RX buffer is used to prefetch data as per size configured in BUFnCR[ADATSZ]. This can lead to stale data being present inside the AHB buffer in case any AHB or IP command is used to over write the address which is already prefetched inside the AHB buffer. To prevent stale data being read there are two options.

- Use SPTRCLR[PREFETCH_DIS] bit programming as '1' this will ensure that every time a new AHB read transaction starts it reads new data from external memory instead of reading the data already stored in AHB buffer. This method invalidates all the prefetched data in AHB buffers when a new AHB read transfer is started hence making sure that every new AHB transaction results in a buffer miss and new data is read in buffer.
- Use BFGENCR[WR_FLUSH_EN] bit. When this bit is programmed as '1' during IP command write or AHB write, XSPI checks for the address at which the data is being currently written. If the data for that corresponding address is already prefetched and present inside any of the AHB buffer/sub-buffer then it invalidates all data entries of that specific buffer/sub-buffer. The data present in rest of the buffers/sub-buffers is not changed.

41.3.3.13 AHB write

This module supports a single initiator AHB write with these burst types — INCR4, INCR8, INCR16, and INCR unspecified 4n bytes and single 4n bytes — and these burst types are supported with all HSIZE's. An AHB write transaction size can be expressed in multiples of 4 bytes, and a single transaction must have a minimum of 4 bytes. In case of HyperRAM ,this limitation of minimum 4 bytes is not present. Also unspecified INCR and single transactions are supported without the limitation of 4n bytes in case of HyperRAM. In case of RAM devices, this limitation of minimum 4 bytes is not present and also unspecified INCR and single transactions are supported without the limitation of 4 bytes. XSPI controller changes to the RAM mode in which, the write data mask can be done using DQS when MCR[DQS_OUT_EN] = 1. See OctalRAM support for more details.

The following restrictions apply to the XSPI module with respect to AHB write transactions:

- AHB transactions of WRAP* types are not supported. For these unsupported access types, the controller returns an AHB error response (HRESP) and does not initiate any flash memory transaction.
- Back to back, fixed address AHB write is not supported.
- Early burst termination is not supported for AHB transactions.

The XSPI controller queues back-to-back AHB writes with continuous address and sends them in a single flash memory transaction. However, in case the AHB address of a burst is not in continuous addressing sequence to previous write burst, a new flash memory transaction is initiated. Supports random read and Write operations for PSRAM devices.

Following is the programming sequence for an AHB write:

1. Check that SR[BUSY] is deasserted or is 0 and check that the TX buffer is empty. If there is any residue data present in the TX buffer (SR[TXNE]), the buffer must be cleared by writing '1' to MCR[CLR_TXF].
2. If the memory device used is a flash and based on the type of flash, AWRCR[PPW_WR_DIS] and AWRCR[PPW_RD_DIS] need to be programmed.

   The default reset value for both the fields is 0 and allows writing to flash memory device assuming a very low T_{pp} (page program wait time) period. This period is very high for most of the flash memory devices. After any write to a flash memory device, XSPI waits for the T_{pp} period to finish automatically before proceeding further when AWRCR[PPW_WR_DIS] = 1.

   This is controlled by the AWRCR[PPW_WR_DIS] and AWRCR[PPW_RD_DIS] field configurations.
3. Program the BFGENCR[SEQID] or BFGENCR[SEQID_WR] (if BFGENCR[SEQID_WR_EN] = 1) to point to the LUT where flash memory write sequence is stored.
4. AHB write initiator can now enable the transactions to be written to the flash memory.
5. When a flash memory write completes and flash memory CS is deasserted, the software must poll SR[BUSY] for the deasserted value, which is 0. This indicates that the AHB write is complete.
6. Whenever flash memory CS is deasserted after an AHB write sequence, FR[PPWF] is set if any of AWRCR[PPW_RD_DIS/PPW_WR_DIS] is enabled. FR[PPWF] disables any further flash memory read/write based on AWRCR[PPW_RD_DIS/PPW_WR_DIS]. After FR[PPWF] is set, any AHB transfer pending on bus remains in waiting state with HREADY low unless this flag is cleared.
7. On the expiry of wait period required for flash memory, FR[PPWF] is cleared by hardware or Software, depending on the BFGENCR[PPWF_CLR] bit. See Page program wait handling for details. After this flag is cleared, XSPI automatically generates another transfer for flushing the residue data in TX buffer, if any.
8. The software must ensure that the correct write LUT sequences are selected corresponding to the AHB write transactions. XSPI would malfunction if flash memory read sequence is selected during AHB write or vice versa.
9. SW should ensure that for NOR flash or other flashes where page program wait period is applicable after flash page programming (AWRCR[PPW_WR_DIS]=1), the AHB burst is broken at the page boundary crossing of flash. A new NON SEQ transaction should start after the page boundary crossing. XSPI generates AHB HRESP error if this condition is not met. For NOR flash devices this address should be 4 bytes aligned. Sequence-Abort scenario is generated to stop the ongoing TX in case of buffer underrun and No bus error will be generated. Another transaction will be generated with new calculated start address to send the residual TX data.

NOTE

- In case any IPS write transfer results in an under-run because of which some residue data is left in TX buffer and FR[TBUF] flag is set, the AHB write transfer will remain pending unless this flag is cleared by SW.
- In case of early abort (having residue data in TX FIFO) during an AHB write, all other requests are gated in arbitration till the residue data in TX buffer is flushed completely.

41.3.3.13.1 OctalRAM support

XSPI supports OctalRAM memories, and by virtue of this protocol, XSPI supports the following functionalities:

- Bidirectional data strobe/read write data strobe (RWDS)

- During command and address phase, to support normal/initial latency, the byte wise RWDS pad should be pulled down

- Variable refresh latency
  - If the value of MCR[VAR_LAT_EN] is 1, based on the status of RWDS from the memory device during the command/address phase, XSPI includes an additional initial access latency. If RWDS is high, XSPI includes twice the latency mentioned in the dummy phase. If RWDS is low, latency mentioned in the dummy phase is not included.
  - In XSPI, MCR is not part of LUT programming, so, if MCR[VAR_LAT_EN] = 0, then the RWDS status is not considered by the XSPI for dummy cycle latency.
  - If the value of MCR[VAR_LAT_EN] is not set, fixed latency mentioned in the dummy phase is included.
- Read data strobe to latch data from memory device
- Programming considerations:
  - Data masking during the AHB write is enabled when MCR[DQS_OUT_EN] is programmed as 1.

41.3.3.13.2 Page program size

In some of the external flash devices there is a restriction that data can be written only for a defined page size in a single flash program cycle. To adhere to this restriction for AHB write transfers, SFACR[PPWB] field can be programmed. This page program boundary size should be programmed in Log2 (size in bytes) format. If the AHB transfer is crossing this defined page boundary then XSPI breaks the transfer at page boundary and waits for flash programming to complete. Then it generates a new flash programming transfer from next page boundary.

Software should ensure that starting AHB address is 32 bit aligned so that at page boundary the limitation of AHB transaction in multiple of 4 bytes is maintained wherever required.

41.3.3.13.3 Timeout in case of AHB write

In case of AHB write if the AHB interface is not served and HREADY is kept low then timeout occurs and FR[AITEF] is set. If no AHB data has been accepted by XSPI (TX buffer is empty assuming no IPS transfer is currently ongoing) then after timeout remaining beats of the AHB write burst are given AHB HRESP error and AHB state machine recovers. But in case the data is accepted in the TX buffer (TX buffer not empty) for multiple beats but no transfer has been generated towards the flash. This situation may occur because of wrong SEQID programming or some other reason. If a timeout occurs in this situation then a SW reset (MCR[SWRSTHD]) will be required to clear the TX buffer and reset the AHB state machine.

41.3.3.14 Byte ordering of serial flash memory read data

The basic scheme is that the first byte read out of the serial flash memory device, which is addressed by SFAR[SFADR], corresponds to RBDR0[31–24] for IP command read. Similarly, to send a single byte it should be positioned in TBDR[7–0]. In contrast to that for AHB command read, the bytes are always positioned according to the byte ordering of the AHB bus.

- Byte ordering in individual flash memory mode

  The following table provides the byte ordering scheme of how the byte oriented data space of the serial flash memory device is mapped into one single 32-bit entry of the RX buffer or the AHB buffer. The table is valid within the following context:
  - Flash memory A in individual flash memory mode
  - All AHB data read commands with 32-bit access size

Table 335. Byte ordering in individual flash memory mode

Serial flash memory byte numbering	3	2	1	0
Buffer entry bit position [31–0] (32-bit data width)	[31–24]	[23–16]	[15–8]	[7–0]

NOTE

For IP commands, the read size can be specified as number of bytes. If this number is not a multiple of four, then the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

For AHB commands and reads, starting from an address not aligned to 32-bit boundaries, the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- Buffer entry ordering for 64-bit read access

  For read access via the AHB interface, a 64-bit access is possible. Each 64-bit access reads two 32-bit entries, simultaneously. The ordering of these 32-bit entries within the 64-bit word is provided in the following table.

Table 336. 64-bit read access buffer entry ordering

AHB read data bit position [63–0]	[63–32]	[31–0]
Buffer entry #	Even (0, 2, 4, ...)	Odd (1, 3, 5, ...)

41.3.3.15 Normal mode interrupt and DMA requests

XSPI module has different flags that can only generate interrupt requests and one flag that can generate an interrupt as well as DMA requests. The following table lists the eight conditions. Note that the flags mentioned in the table are associated with the Flag (FR).

Table 337. Interrupt and DMA request conditions

Condition	Flag (FR)	DMA
Data learn pattern failure	DLPFF	-
TX buffer fill	TBFF	-
TX buffer underrun	TBUF	X
Illegal instruction error	ILLINE	-
RX buffer drain	RBDF	X
RX buffer overflow	RBOF	-
AHB buffer overflow	ABOF	-
AHB sequence error	ABSEF	-
AHB illegal transaction error	AITEF	-
AHB illegal burst size error	AIBSEF	-
IP command trigger could not be executed error	IPIEF	-
IP command related transaction finished	TFF	-

Each condition has a corresponding field in Flag (FR) and a request enable field in Interrupt and DMA Request Enable (RSER). FR[RBDF] has separate enable fields for generating IRQ and DMA requests. Note that not all the fields have an individual IRQ line. See the chip's Interrupt Vector table for details.

- Transmit buffer fill interrupt request

  This indicates that the TX buffer can accept new data. The buffer fill interrupt is asserted if FR[TBFF] is asserted and if the value of the corresponding enable field, RSER[TBFIE], is 1. See TX buffer Operation for details on the assertion of FR[TBFF].

  Apart from IRQ, it is possible to handle the TX buffer fill by using the DMA. If the value of RSER[TBFDE] is 1, a DMA request is triggered when the number of available space in the TX buffer is more than the TBCT[WMRK] valid entries and value of

SR[TXWA] is set. The application must configure the environment appropriately (for example, the DMA controller) for the DMA transfer.

In SFP Enabled configurations, program RSER[TBFDE] bit to 1 only when arbitration is won and TBDR access is unlocked after checking STATE field of FSMSTAT register. After TBDR programming is completed by DMA, reset this bit otherwise, XSPI SFM state will remain busy and no further IPS accesses can be served by that Target queue.

- Receive buffer drain interrupt or DMA request

  This is derived from FR[RBDF], indicating that the RX buffer of the XSPI module has data available from the serial flash memory device to be read by the host. It remains set as long as SR[RXWE] is configured. Also, RSER[RBDIE] enables the related IRQ.

  Apart from the IRQ, it is possible to handle the RX buffer drain by using the DMA. If the value of RSER[RBDDE] is 1, a DMA request is triggered when the RX buffer contains more than RBCT[WMRK] valid entries. The application must set the environment appropriately (for example, the DMA controller) for the DMA transfers.

- Buffer overflow/underrun interrupt request

  This is a combination of the following fields (all located in the Flag (FR) with the related enable bits in the Interrupt and DMA Request Enable (RSER)):

  — TBUF - TX buffer underrun, enabled by TBUIE

  — RBOF - RX buffer overflow, enabled by RBOIE

  — ABOF - AHB buffer overflow, enabled by ABOIE

  — The transmit buffer underrun indicates that an underrun condition in the TX buffer has occurred. It is generated when a write instruction is triggered whilst the TX buffer is empty and the value of RSER[TBUIE] is 1.

  — The receive buffer overflow indicates that an overflow condition in the RX buffer has occurred. It is generated when the RX buffer is full, an additional read transfer attempts to write into the RX buffer, and the value of RSER[RBOIE] is 1.

  — The AHB buffer overflow indicates that an overflow condition in the AHB buffer has occurred. It is generated when the AHB buffer is full, an additional read transfer attempts to write into the AHB buffer and the value of RSER[ABOIE] is 1.

  — The data from the transfers that generated the individual overflow conditions is ignored.

- Serial flash memory command error interrupt request
- Transaction finished interrupt request

  The IP command transaction finished IRQ indicates the completion of the current IP command. It is triggered by FR[TFF] and is masked by RSER[TFIE].

IPED decryption error interrupt is generated in case the IPED module generates an error during data decryption. It can be enabled using RSER[IPEDIE].

AHB read address error interrupt is generated in case the read address given by AHB interface does not lie in any of the sub-buffer address ranges or lies within multiple sub-buffers address range of a single buffer. It is triggered by FR[RDADDR] and can be enabled using RSER[RDADDRIE].

Performance monitors overflow interrupts are triggered by FR[PERFOVF] and can be enabled by RSER[PERFOVIE] bit. these are generated when any of the performance monitor counter overflows.

Arbitration lock timeout interrupt is triggered by ERRSTAT[ARB_LOCK_TO] and is generated when arbitration lock requested by any SFP queue is not cleared and results in timeout counter expiring. It can be enabled by INT_EN[ARB_TO_IE].

Arbitration win interrupt is generated when any TG queue which has requested the transfer is granted. It can be enabled by INT_EN[ARB_WIN_IE] and is triggered by ERRSTAT[ARB_WIN].

41.3.3.16 Address schemes

Earlier, serial flash memories supported only a 24-bit address space, restricting the maximum memory size of the serial flash memory to 16 MB. The new memory specification supports two types of 32-bit addressing mode in addition to the legacy 24-bit address mode. It also supports segregation of address programmed into Row address and Column address of the flash memory.

Extended address mode

In this mode, the legacy 24-bit commands are converted to accept 32-bit address commands. The memory will default to the legacy 24-bit mode (only the first 16 MB of the memory can be accessed while in this mode), and the memory must be configured for 32-bit address mode to access the entire memory. Also, while programming the LUT sequence in XSPI for 32-bit mode, the ADDR and ADDR_DDR commands should be programmed with 32d as the operand value with SFACR[CAS] programmed to 0. If a flash memory needs some bits of the address as its column address, then it must be considered that a total of 12 bits are required by the flash memory; however, the number of bits should not exceed 32 because the maximum address supported by XSPI is 32 bits.. Each of the memory vendors have a different way of enabling this mode (see the memory specification from memory vendors). For example, the command B7h sent to the Macronix flash memory enables it for the 32-bit address mode.

Extended address register mode in external memory device

In this mode, the upper 8 bits of the 32-bit address are provided by the Extended address register in the memory. This effectively converts the legacy 24-bit address command into 32-bit address commands. The memories greater in size than 16 MB consist of banks of 16 MB each. The 8 bits occupied in the extended address register effectively enable a bank. For example, in Spansion memory, when the extended address register is updated with a value of 1h, with the help of the 17h command, it opens Bank1 of the memory. The consequent 24-bit address commands access Bank1. The extended address register needs to be updated with the respective value for access to other banks.

Separation of address into rows and columns

This mode has been introduced for flash memories that need addresses segregated into rows and columns. The value in SFACR[CAS] defines the width of the column address required by a flash memory. The actual address to be provided is derived from the incoming address in case of AHB-initiated transactions and the value of SFAR in case of IPS-initiated transactions, if the value of SFACR[CAS] is 0. Otherwise, the actual address takes CAS into consideration. If the value of SFACR[CAS] is 3, then bits 26-3 of the programmed address are sent to the flash memory as its page/row address and bits 2-0 are sent as its column address. If a flash memory requirement for column address is less than the number of pads in which address has to be sent, then XSPI appends the remaining bits by 0. You must program the operand value in CADDR and CADDR_DDR command accordingly. The total number of address bits in the column and row address must be 32-bits or less.

Word addressable mode for flash memory

This mode has been introduced for flash memories, which have a word-addressable memory. This means, each address of the flash memory contains one word (two bytes) of data. The value of SFACR[WA] is configured to 1 to enter this mode. If the address of flash memory contains two-words (4 bytes) of data, set the XSPI's SFACR[WA_4B_EN]. XSPI internally divides the incoming address in the AHB bus or the address in the SFAR to map it to a valid flash memory location. For example, if the incoming address is 2004h, the controller re-maps this address to access the flash memory location 1002h.

41.3.3.17 AHB alignment

XSPI module can define the AHB transaction (read/write) address boundary using BFGENCR[ALIGN]. Any continuous AHB write transfer is broken into smaller sub-accesses at the address boundary specified using the BFGENCR[ALIGN] field setting.

For AHB read transactions, XSPI checks the start address and end address (start address + ADATSZ programmed in BUFnCR registers) to see if it is crossing the address boundary specified by ALIGN. If the transaction crosses the address boundary, then it reduces the transaction size such that the data pre-fetch is stopped before the address boundary. Now, if the AHB initiator is reading the data sequentially it will get buffer hits up to the ALIGN boundary. When it reaches the next address boundary it will get a buffer miss and a new data pre-fetch will be launched towards the external memory device.

For AHB write transfers, XSPI keeps track of the start address and then compares it with the address of subsequent transaction received on the AHB bus. When the boundary programmed in BFGENCR[ALIGN] field is reached, XSPI negates AHB HREADY to block new accesses. This signal will be kept low until all the previous received write data is written to external memory and chip select is de-asserted. After that it allows next AHB write data to be received by making HREADY high thus ensuring that transaction is split at address boundary on external memory.

41.3.3.18 AHB transaction split

41.3.3.18.1 Write split

When this feature is enabled (BFGENCR[SPLITEN]=1), HW will split one single AHB write burst into smaller bursts on flash side, based on BFGENCR[AHBSSIZE] setting.

Assumptions:

When IPED GCM is enabled, Split size should be N*32B and the split happens on 40B/80B so that the Header/authentication is included with every block.

NOTE

When split is enabled, as the alignment logic will work before the split, both will work parallel and needed. When IPED is enabled alignment logic is used to handle the context boundary crossing(GCM and CTR) in a single transaction.

41.3.3.18.2 Read split

When this feature is enabled (BFGENCR[SPLITEN]=1), HW will realign the prefetch size to split size and in that way splits a single read burst into smaller bursts, based on BFGENCR[AHBSSIZE] setting.

Assumptions:

WRAP bursts will not work if the total burst size exceeds the split size.

When IPED GCM is enabled, the split size should be 32B/64B. HW will the split the bursts into 40/80 Bytes to include Header/authentication tag for each block.

41.3.4 Stop and module disable modes

41.3.4.1 Stop mode

XSPI supports the Stop mode protocol using the Stop request and Stop request acknowledgement handshake. By default, Stop request acknowledgement signal is de-asserted. When a request is made to enter Stop mode, the XSPI block acknowledges the request by asserting Stop request acknowledgement when it is ready to have its clocks shut off.

XSPI will return the Stop request acknowledgement handshake when all the conditions below are fulfilled:

- SR[BUSY] = 0
- SR[AHBTRN] = 0
- RBSR[RDBFL] = 0
- SR[RXDMA] = 0
- SR[TXDMA] = 0
- No pending interrupts (none of the flags in Flag (FR) that are enabled as interrupts is set).

The conditions mentioned above ensures the following:

- There is no SFM command currently being executed.
- All the data read into the RX buffer from the serial flash memory have been fetched by the application.
- There is no current AHB access.
- There is no active DMA request.
- There is no enabled interrupt that is pending.

While the clocks are shut off, the XSPI memory-mapped logic is not accessible. In Stop mode, write to XSPI registers is disabled and does not have any effect.

NOTE

It is illegal to issue a new SFM command starting two clock cycles prior to raising the request of entering Stop mode until the XSPI stays in this mode. Software must ensure that there are no pending transactions in SFP queues before requesting Stop or Module Disable mode.

41.3.4.2 Module Disable mode

Module Disable mode is a block-specific mode that the XSPI can enter to disable serial flash memory clock and AHB command. This mode can be entered by:

- The host software: by asserting MCR[MDIS].
- The hardware: by asserting the ipg_doze signal while MCR[DOZE] is asserted.

Below are the conditions that are checked by hardware before asserting acknowledgment as part of stop request/acknowledgment handshake to enter Module Disable mode:

- SR[BUSY] = 0
- SR[AHBTRN] = 0
- RBSR[RDBFL] = 0
- SR[RXDMA] = 0
- SR[TXDMA] = 0
- None of the flags in Flag (FR) are enabled as interrupts is set.

The conditions mentioned above ensures the following:

- There is no SFM command currently being executed.
- All the data read into the RX buffer from the serial flash memory have been fetched by the application.
- There is no current AHB access.
- There is no active DMA request.
- There is no enabled interrupt that is pending. SFP interrupts are not checked here so software needs to handle those.

In Module Disable mode, few configuration fields (mentioned in MCR and SMPR register function) of XSPI can only be written.

NOTE

It is illegal to issue a new SFM command starting two clock cycles prior to raising the request of entering the Module Disable mode until the XSPI stays in this mode. Software should follow stop acknowledgment handshake to ensure that there are no pending transactions in SFP queues while requesting stop or module disable mode.

41.3.4.3 Leaving Stop mode and Module Disable mode

After XSPI has returned to Normal mode, the execution of the first SFM command is deferred until the clock to drive that part of the module related to the serial flash memory device is available.

41.3.5 Dummy cycles generation

The operand value for a DUMMY instruction (LUT*n*[OPRND*n*]) controls the dummy cycle operation as described in the following table.

- Bit 7 of the operand value controls split DUMMY operation. That is, if bit 7 is 1 (OPRND*n* > 80h), XPSI performs a split DUMMY operation.
- Bits 6–0 specify the number of cycles for the DUMMY operation, with modification resulting from the split DUMMY, RWDS, and variable latency settings.

For example sequences that use the dummy cycle capability, see Example LUT sequences.

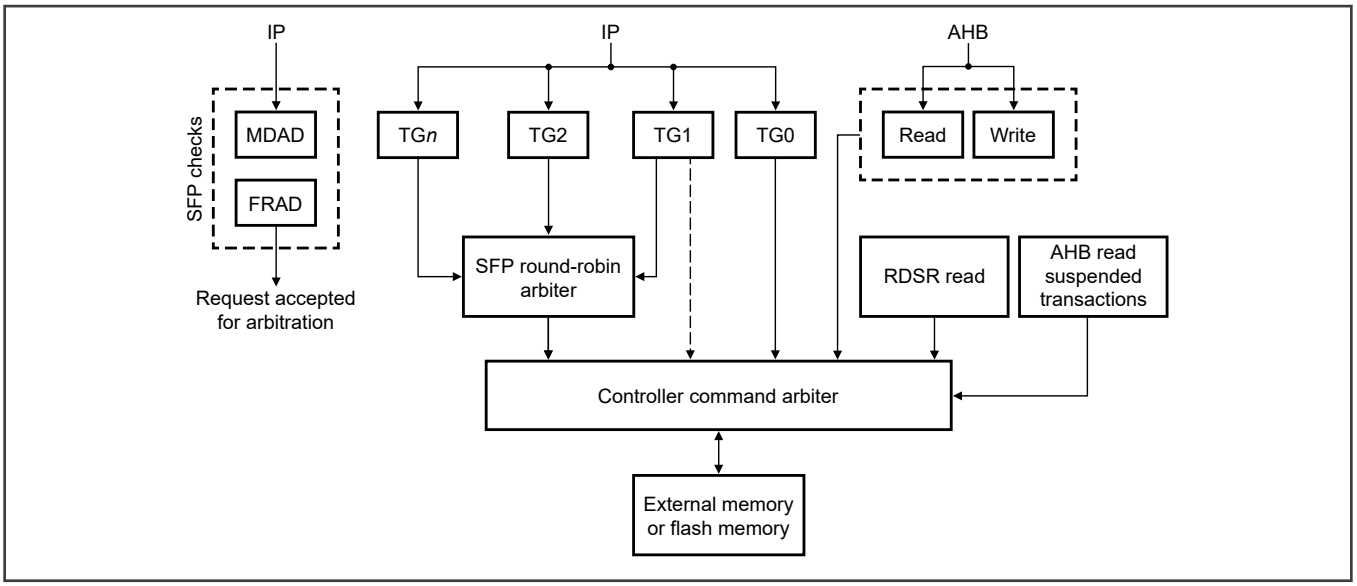
Table 338. Dummy cycles generation

Operand		RWDS[1]	Variable latency [2]	Dummy cycles	Notes
Bit 7	Bits 6–0 (*m*)				
1	> 1[3]	Asserted	1	2*m* + 1	This scenario requires one extra cycle to achieve 2x cycle latency. You accomplish this by inserting a MODE instruction (which adds 1 cycle) before the DUMMY instruction. With RAM devices, for *m* cycles, specify *m* – 1 in the operand.
0	> 1[3]	Asserted	1	2*m* + 2	—
1 or 0	> 1[3]	—	0	*m* + 1	—
1	0[4]	Asserted	1	3	—
0	0[4]	Asserted	1	4	—
0	0[4]	—	0	2	Two is the minimum number of dummy cycles.

1. Read Write Data Strobe
2. MCR[VAR_LAT_EN]
3. All pads act as inputs.
4. All pads act as outputs.

41.3.6 IP and AHB command arbitration

XSPI controller supports arbitration between the flash accesses coming from IP interface and AHB interfaces. The arbitration scheme is shown in figure below:



NOTE

Above figures shows 'n' number of TG queues but only 2 TG queues exist. Also SFP round-robin arbiter is not present and TG1 request directly goes to Controller command arbiter.

Arbitration priority between different requests coming to controller command arbiter block is given in table below with 1 being highest priority order.

Transaction request	Priority order
TG0 (IP command)	1
AHB write	2
AHB Read	3
Suspended AHB Read transaction	4
TG 1 (IP command)	5
Reserved	6
RDSR read	7

On IP command interface the incoming access requests can be filled in multiple TG queues based on the access criteria being set in SFP block. They are presented for arbitration only after they have passed through the MDAD and FRAD checks of SFP.

On AHB interface only one transaction from read and write can come at a time. AHB Read transaction is presented for arbitration only in case of buffer miss (when data is not present inside AHB buffer and has to be read from the external memory device). The suspended AHB Read prefetch requests are generated as a part of AHB Read requests when no high priority AHB Read initiators are pending on the bus. In case any AHB Write transfer comes, it may abort any ongoing read-prefetch or any suspended read transfers. For AHB Write, the requests generated after an early abort (data left in TX Buffer and CS de-asserted) are taken at highest priority level (even higher than TG0) as the data is already present inside the TX Buffer or is part of ongoing AHB Write transfers. Early abort in AHB Write is treated as pending AHB Write transfer which is in continuation of last AHB Write request. The RDSR read requests are generated when Page program wait period is ongoing after an AHB Write transfer has completed. This is not a software generated request; it is automatically generated by XSPI as part of page program period handling. See AHB command page program wait handling for details.

The arbitration state machine remains in IDLE state and then moves on to ARB state after the requests have arrived for arbitration. It remains in ARB state for one clock cycle then moves on to the state which has been granted arbitration, for example, AHB write, AHB Read, or SFP IPS TG state. After the flash transfer is completed the state machine goes to PROC_FIN state and waits for other blocks to finish the processing before going for next arbitration. The current state can be checked by reading Status Register (SR[ARB_STATE]).

Please refer the below tables for some of the examples of arbitration scenarios:

Table 339. Arbitration scenarios for handling AHB and IPS requests

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
AHB-AHB	AHB Read	AHB Read	**Scenario 1: Normal Read** AHB Read from Environment-2 will wait till AHB Read from Environment-1 (ongoing) completes. **Scenario 2: High priority based suspend transaction** AHB Read prefetch is ongoing for low priority AHB initiator from Environment-1. Higher priority AHB Read request from Environment-2 will suspend the ongoing prefetch and arbitration will be granted to Environment-2.
	AHB Read	AHB Write	**Scenario 1: Contention for grant**

Table continues on the next page...

Table 339. Arbitration scenarios for handling AHB and IPS requests (continued)

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
			AHB Read request (Environment-1)and AHB Write request (Environment-2) are present at arbiter for contention. As per Arbitration Priority (AHB Write is having greater priority than AHB Read) - Environment-2 will be given grant. **Scenario 2: Write terminate enabled (BFGENCR[WRTER_EN] = 1b)** AHB Read prefetch for Environment-1 is ongoing. AHB Write comes for Environment-2 and aborts the ongoing read prefetch and takes the grant. **Scenario 3: Write terminate disabled (BFGENCR[WRTER_EN] = 0b)** AHB Read prefetch for Environment-1 is ongoing. AHB Write comes for Environment-2. Grant to AHB Write is given after completion of AHB Read prefetch.
	AHB Write	AHB Read	**Scenario 1: Normal write before Read** AHB Read request (Environment-2) is granted after the completion of AHB Write (Environment-1). **Scenario 2: Wait period with Read While Write (RWW) NOR Flash** AHB Write from Environment-1 is completed (SR[BUSY] = 0b), FR[PPWF] is set, AWRCR[PPW_WR_DIS] is set, AWRCR[PPW_RD_DIS] is not set. AHB Read from Environment-2 comes on top. RWW flash, AHB Read is allowed during wait period, hence the arbitration will be given to the AHB Read from Environment-2. **Scenario 3: Wait period with non-RWW NOR Flash** AHB Write is completed (SR[BUSY] = 0b), FR[PPWF] is set, AWRCR[PPW_WR_DIS] is set, and AWRCR[PPW_RD_DIS] is set. Non-RWW flash, AHB Read is not allowed during wait period, hence the arbitration will not be given to the AHB Read (Environment-2) till PPW period is over. **Scenario 4: No wait period with PSRAM** There is no PPW (wait period) for PSRAM. AHB Read request (Environment-2) is granted after the completion of AHB Write (Environment-1).
	AHB Write	AHB Write	**Scenario 1: Continuous address - no PPW**

Table continues on the next page...

Table 339. Arbitration scenarios for handling AHB and IPS requests (continued)

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
			Due to continuous address both AHB Write will be executed in the same grant. First AHB Write from Environment-1 followed by AHB Write from Environment-2. **Scenario 2: Random address - PPW** Due to random address, HREADY is pulled low to not allow AHB Write from Environment-2. Flushing mechanism is used to flush the data for AHB Write from Environment-1. Once the AHB Write from Environment-1 is completed, FR[PPWF] is set. During FR[PPWF], AHB Write request from Environment-2 is stalled till FR[PPWF] is over.
AHB-IPS	AHB Read	IPS Read	**Scenario 1: Normal operation** AHB transaction is ongoing, IPS transaction request is put on stall till the AHB transaction is completed (SR[BUSY] = 0b). **Scenario 2: AHB Read with HREADY timeout** User environment went dormant at Environment-1 causing FR[AITEF] flag to be set (HREADY timeout). IPS Read will be stalled. **Recovery mechanism:** On applying soft reset simultaneously on AHB and flash side, will release the AHB grant. Arbitration goes to IDLE as SR[BUSY] = 0b (due to chip select deassertion). Arbitration is given according to priority for the pending requests. In this case IPS is given grant **Cost:** Software needs to re-initiate the AHB transaction for Environment-1 from the address reported by XSPI during soft reset. Due to this pending IPS request from Environment-2 gets the grant and AHB transaction will be delayed.
	AHB Read	IPS Write/Erase	
	AHB Write	IPS Write/Read	**Scenario 1: Underrun - Residue data in TX path** No issue as early abort flag is set and auto flushing mechanism will be executed for flushing the residue data. The arbitration is blocked for IPS transaction (Environment-2). **Scenario 2: PPW for Non-RWW flash** AHB Write is completed (SR[BUSY] = 0b), FR[PPWF] is set, AWRCR[PPW_WR_DIS] is set, and AWRCR[PPW_RD_DIS] is set. IPS comes on top: Due to FR[PPWF] set, IPS transaction is stalled. Due to AWRCR[PPW_WR_DIS] and AWRCR[PPW_RD_DIS] is set, AHB Write and AHB Read are also blocked.

Table continues on the next page...

Table 339. Arbitration scenarios for handling AHB and IPS requests (continued)

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
			Scenario 3: PPWF with HREADY timeout AHB Write is completed, FR[PPWF] flag is set, HREADY timeout happens. Due to FR[PPWF] flag Arbitration is blocked for IPS and arbitration lock timer will start. To clear the hang scenario software reset is used. Software reset does not clear the FR[PPWF] flag. After software deassert, as PPW is there, IPS will be blocked,only RDSR request will be given grant. **Scenario 4: PPWF with HREADY timeout and Arbitration Lock timeout** AHB Write is completed, FR[PPWF] flag is set, HREADY timeout happens, Arbitration lock timeout also occurred. Due to FR[PPWF] flag Arbitration is blocked for IPS and arbitration lock timer will start and arbitration lock timeout is generated on reaching arbitration counter max value. To clear the hang scenario software reset is used. Software reset does not clear the FR[PPWF] flag. After software deassertion, as PPW is there, IPS will be blocked, only RDSR request will be given grant.
IPS-AHB	IPS Read	AHB Read/Write	**Scenario 1:** IPS Read is ongoing, AHB will wait till the IPS is completed (SR[BUSY] = 0b). **Scenario 2:** While reading data through IPS, User became dormant, MTO came. IPS grant is release. AHB Grant is gated until MTO is cleared by Main environment.
IPS-IPS	IPS Read	IPS Write/Erase	**Scenario 1: If any ongoing queue becomes dormant and it has not taken arbitration lock** SW: 1. Main environment needs to clear the TX Buffer, once MTO happens and SR[TXNE] flag is set. 2. After clearing the TX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR[TBUF]) by writing W1C (XSPI HW will clear the interrupt for User environment, once the Main environment acknowledges them). 3. Main environment needs to clear it's own timeout flag.
	IPS Write/Erase	IPS Write/Erase	

Table continues on the next page...

Table 339. Arbitration scenarios for handling AHB and IPS requests (continued)

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
			New environment which got grant has to wait for these steps and FSMSTAT[STATE] will remain 00b until then. **Scenario 2: If any ongoing queue becomes dormant and it has taken arbitration lock** SW: • Main environment needs to clear the TX Buffer, once MTO happens and SR[TXNE] flag is set. • After clearing the TX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR[TBUF]) by writing W1C (XSPI HW will clear the interrupt for User environment, once the Main environment acknowledges them). • Main environment needs to clear it's own timeout flag. • Main environment to release arbitration lock through W1C field of ERRSTAT register after arbitration timeout error happens. **Scenario 3: SFP transaction has been arbitrated with arbitration lock and TBDR is being filled and User environment became dormant** • Main environment needs to clear TX Buffer/RX Buffer once MTO happens. • Main environment needs to clear it's own timeout flag. Arbitration stays with that TG queue until Arbitration lock timeout error comes and is acknowledged by the Main environment. **Scenario 4: SFP transaction has been arbitrated with arbitration lock and User environment was reading data and became dormant** 1. Main environment needs to clear the TX Buffer/RX Buffer once MTO happens. 2. After clearing the RX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR fields: RFDF if set) by writing W1C (XSPI HW will clear the interrupt for User environment, once the Main environment acknowledges them). 3. The Main environment needs to clear it's own timeout flag. Arbitration stays with that TG queue until Arbitration lock timeout error comes and is acknowledged by the Main environment.

Table continues on the next page...

Table 339. Arbitration scenarios for handling AHB and IPS requests (continued)

Arbitration category	Environment-1 operation (Ongoing)	Environment-2 operation (New Request)	Scenarios
			Scenario 5: TBDR is filled above threshold. XSPI is triggered with SFAR and IPCR and user became dormant and underrun came (watermark is wrongly programmed) **Situation:**Flash will request data, since User environment became dormant, underrun will happen (FR[TBUF] = 1b). SW: 1. Main environment needs to clear the TX Buffer, once MTO happens and SR[TXNE] flag is set. 2. After clearing the TX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR[TBUF]) by writing W1C (XSPI HW will clear the interrupt for User environment, once the Main environment acknowledges them). 3. Main environment needs to clear it's own timeout flag. New environment which got grant has to wait for these steps and FSMSTAT[STATE] will remain 00b until then. **Scenario 6: Arbitration lock is requested and TBDR is filled above threshold. XSPI is triggered with SFAR and IPCR and MTO happened (too much time due to wrong MTO programming)** **Situation:**Flash will request data, since User environment became dormant, underrun will happen (FR[TBUF] = 1b) 1. Main environment needs to clear the TX Buffer, once MTO happens and SR[TXNE] flag is set. 2. After clearing the TX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR[TBUF]) by writing W1C (XSPI HW will clear the interrupt for User environment, once the Main environment acknowledges them). 3. Main environment needs to clear it's own timeout flag. 4. Main environment to release arbitration lock through W1C field of ERRSTAT register after arbitration timeout error happens.
Semaphore	IPS Write/Erase	AHB Read	**Scenario:** Initiator (with domain ID) is given semaphore lock, IPS Transaction is ongoing, AHB Read request from the same initiator (domain ID) comes, then it will be given grant only after IPS is finished. Any AHB request from any other AHB initiator is given error.

Table 340. SFP round-robin arbitration scenarios for handling multiple IPS requests

Scenario	FSM State	MTO error	Arbitration lock requested	Arbitration lock counter	SW workaround
TG Queue has been granted but XSPI is busy	0	No	Yes	Not started (Since FSM state is 00)	TG will wait for XSPI to be idle. See XSPI IDLE conditions section
SFP transaction has been arbitrated and TBDR is being filled and User environment became dormant.	1	Yes (User environment became dormant)	Yes	Started (As soon as MTO error happens)	1. Main environment needs to clear the TX Buffer/RX Buffer pointers once MTO happens. 2. Main environment needs to clear its own timeout (MTO) flag ERRSTAT[TO_ERR] by W1C. 3. Status of TX Buffer register (TBSR) and RX Buffer register (RBSR) for that User environment queue is not updated unless the queue wins the arbitration again (keeping the last value). Arbitration stays with that TG queue until Arbitration lock timeout error comes and is acknowledged by the Main environment.
SFP transaction has been arbitrated and TBDR is filled above threshold. XSPI is triggered with SFAR and IPCR and MTO error happened	10	Yes	Yes	Started (As soon as MTO error happens)	
SFP transaction has been arbitrated and User environment was reading data and became dormant	11	Yes (User environment became dormant)	Yes	Started (As soon as MTO error happens)	1. Main environment needs to clear the TX Buffer/RX Buffer pointers once MTO happens. 2. Main environment needs to clear its own timeout (MTO) flag ERRSTAT[TO_ERR] by W1C. 3. Status of TX Buffer register (TBSR) and RX Buffer register (RBSR) for that User environment queue is not updated unless the queue wins the arbitration again (keeping the last value). 4. After clearing TX/RX Buffer, Main environment needs to clear the corresponding interrupt flags (for FR field: RBDF if set) by writing W1C. Arbitration stays with that TG queue until Arbitration lock timeout error comes and is acknowledged by the Main environment.

41.3.6.1 Page program wait handling

In many of the external flash memories there is a limitation of page program wait. After programming the flash no other access can be forwarded to the flash for some time duration.

41.3.6.1.1 IP command page program wait handling

In case of IP command when flash programming is done, arbitration can be locked by the TG queue which will not allow any other access request to go through unless this arbitration is unlocked by the same TG queue. This makes sure that the duration for which arbitration was locked no other request from AHB/IPS interface can go through except the transfer requests from TG queue which locked the arbitration. So effectively the arbitration remains granted for only the TG queue which has locked the arbitration. Software needs to ensure that after arbitration is locked it does not generate any transfers on that TG queue which can conflict with the page program requirements of flash. It can use this TG queue to read the status register of flash to check if page program wait period has finished, and then it can unlock the arbitration. If it is a RWW type of flash which allows read transactions even during page program wait period, then it can send flash read transfers also through during this arbitration lock period. These read transfers should be targeted on separate Flash block/FRAD region.

For details, see Arbitration lock.

In GCM mode, if a page boundary is crossed, the hardware automatically handles the page program wait period before initiating the next page program for the remaining data.

41.3.6.1.2 AHB command page program wait handling

In case of AHB transaction, based on the type of flash memory device that is used for page programming, software should configure AWRCR[PPW_WR_DIS] and AWRCR[PPW_RD_DIS] fields. The default reset value for both the fields is 0 and allows writing to flash memory device assuming a very low T_{pp}(page program wait time) period. This period is very high for most of the flash memory devices.

Whenever flash memory CS is de-asserted after an AHB write sequence completes, FR[PPWF] is set either of AWRCR[PPW_RD_DIS/PPW_WR_DIS] is set. If both the bits are set (AWRCR[PPW_RD_DIS] and AWRCR[PPW_WR_DIS]) then XSPI locks the arbitration and all accesses to the external memory are blocked. During this period if any AHB read access comes which results in buffer-hit the data is provided after reading from AHB buffer. If AHB read results in buffer miss, it will be kept in waiting state by the XSPI arbiter by keeping HREADY low. An AHB bus error (if SOCCFG[HRESP_ERR_MASK] is 0) is generated for all AHB write transfers. During this period all IP commands are blocked, so there is no mechanism by software to trigger a status register read towards the external memory device.

If only AWRCR[PPW_WR_DIS] is '1' but AWRCR[PPW_RD_DIS] is programmed as '0' then it means that flash is a RWW type and it can handle read transactions during this page program wait period. In this case all IP commands and any AHB write command is blocked from arbitration but any AHB read command will go through to external memory.

It should be ensured that correct programming is done for these bits based on the external memory/flash being used. Also ensure that in case of RWW flash, where AWRCR[PPW_RD_DIS] is '0', software ensures that any additional requirement on read transfer is met. For example, in some cases of RWW the read transaction is allowed during page program wait period but on different sector. It is the application's responsibility to ensure AHB read commands are launched to different sectors/pages from the AHB write that is completing. XSPI controller can’t check/handle these kind of limitations.

41.3.6.1.2.1 RDSR status read

When FR[PPWF] flag is set an internal counter starts within XSPI. After this counter reaches the value programmed in Page Program Wait Time Counter (PPWF_TCNT), a read is triggered by the XSPI module using the SEQID programmed in BFGENCR[SEQID_RDSR]. This counter is working on AHB clock frequency. It is required that this location in LUT is programmed with the required commands for reading the flash status register. After the time period programmed in this counter is reached and status register is read from the flash its value is updated in PPW_RDSR[RDSR] field and PPW_RDSR[VALID] field is set to '1'. The counter is reset to 0 and it again starts incrementing. This PPW_RDSR[31] field will remain set to '1' unless FR[PPWF] flag is cleared and PPW_RSDR[RSDR] bits will keep on updating its value as and when this counter reaches its programmed value. Software needs to ensure correct programming of the register fields in Page Program Wait Read Status (PPW_RDSR) register and correct LUT programming sequence for reading the flash status register as per BFGENCR[SEQID_RDSR]. Now there are two mechanisms to clear this FR[PPWF] flag once the page program wait period is over.

1. Automatic clear by XSPI – This mode is used when BFGENCR[PPWF_CLR]=1'. PPW_RDSR[LOC] needs to be programmed with the location of the field which is to be checked in PPW_RDSR[RDSR] field to know that page program wait period is over or not. When this field is read back as value programmed in PPW_RDSR[RDSR_VAL_CHK] field, XSPI automatically clears the FR[PPWF] flag and unlocks the arbitration for other requests.

2. Clearing by Software – This mode is used when BFGENCR[PPWF_CLR]=0. In this case XSPI depends on software to read PPW_RDSR[RDSR] and clear FR[PPWF] after the page program wait period is over. After the software clears FR[PPWF], it unlocks the arbitration for other requests.

The SFP arbitration lock timeout counter also starts when FR[PPWF] sets. If PPWF is not cleared within the time period specified by SFP_ARB_TIMEOUT[SFP_ARB_TOC] the arbitration lock timeout error (ERRSTAT[ARB_LOCK_TO]) is generated along with interrupt if enabled. Once this ERRSTAT[ARB_LOCK_TO] field is set, the FR[PPWF] flag can not be cleared by automatic mode or by software. Software must clear the arbitration lock timeout flag (ERRSTAT[ARB_LOCK_TO]) first and then needs to clear the FR[PPWF] flag manually to restart normal XSPI operation.

41.3.7 Byte ordering – endianness

XSPI provides support for swapping the flash memory read/write data based on the configuration of MCR[END_CFG]. By default, the data is always returned in a 64-bit LE format on the AHB bus and in a 32-bit LE format on the IPS interface when read through the RX buffer and written in a 32-bit LE format when written through the TX buffer.

The following table shows the complete bit ordering. BE specifies that the high-order bits of the associated data vectors are associated with low-order address positions, and LE specifies that the lower-order bits of the associated data vectors are associated with low-order address positions. See Figure 145.

Table 341. MCR[END_CFG]

00	64-bit BE
01	32-bit LE
10	32-bit BE
11	64-bit LE

The following tables show how this configuration is implemented in XSPI AHB and IPS interfaces, respectively. B in the table signifies byte and the index 1-8 refers to the byte position, which means 1 refers to bits[7:0], and 8 refers to bits[63:56].

Table 342. Byte ordering configuration in AHB

64-bit BE	B1	B2	B3	B4	B5	B6	B7	B8
64-bit LE	B8	B7	B6	B5	B4	B3	B2	B1
32-bit BE	B5	B6	B7	B8	B1	B2	B3	B4
32-bit LE	B4	B3	B2	B1	B8	B7	B6	B5

Table 343. Byte ordering configuration in IPS

32-bit BE	B1	B2	B3	B4
32-bit LE	B4	B3	B2	B1

The following topics show the byte ordering in 64-bit BE configuration for AHB buffer and 32-bit BE for TX/RX buffer.

41.3.7.1 Programming flash memory data

CPU writes configuration data to the TBDR register, such as:

- Write TBDR: 01_02_03_04h
- Write TBDR: 05_06_07_08h

The following table shows the content against each TX buffer entry.

Table 344. Example of XSPI TX buffer

TX buffer entry	Content
0	01_02_03_04h
1	05_06_07_08h

Programming the TX buffer into the external serial flash memory device results in the following byte order to be sent to the serial flash memory:

- 01...02...03...04...05...06...07...08

41.3.7.2 Reading flash memory data into the RX buffer

Reading the content from the same address provides the following sequence of bytes, identical to the write case:

- 01...02...03...04...05...06...07...08

The following table shows the content against each RX buffer entry.

Table 345. Resulting RX buffer content

RX buffer entry	Content
0	01_02_03_04h
1	05_06_07_08h

41.3.7.2.1 Readout of the RX buffer through RBDRn

The RX buffer content appears at CPU read access through the peripheral bus interface in the following order:

- Read RBDR0: 01_02_03_04h
- Read RBDR1: 05_06_07_08h

41.3.7.3 Reading flash memory data into the AHB buffer

Reading the content from the same address as it was written to provides the following sequence of bytes, identical to the write case:

- 01...02...03...04...05...06...07...08

The following table shows the content against each AHB buffer entry.

Table 346. Resulting AHB buffer content

AHB buffer entry	Content
0	01_02_03_04_05_06_07_08h

41.3.7.3.1 Readout of the AHB buffer through memory-mapped read

The AHB buffer content appears at read access on the AMBA AHB interface at the XSPI module boundary:

- 32-bit read access: 1_02_03_04h
- 32-bit read access: 5_06_07_08h
- 64-bit read access: 1_02_03_04_05_06_07_08h

41.3.8 Driving flash memory control signals in single and dual modes

In single and dual modes, the serial flash memory devices that can connect to the XSPI module expect additional control signals on the inputs, which are connected to IOFA[3]and IOFA[2] in the quad mode. For easy interfacing, the outputs IOFA[3:2] for flash memory A are driven to the logic state selected by configuration fields MCR[ISD3FA]and MCR[ISD2FA].

These outputs are driven all the time to the logic level programmed in the MCR except the time when quad commands of the serial flash memory are executed. See the specifications of the related serial flash memory device for details about the inactive level.

41.3.9 Serial flash memory devices

Several different vendors make flash memory devices with a XSPI interface. At present, there is no set standard for the XSPI instruction set. The most common commands currently have the same instruction code for all vendors; however, some commands are unique to specific vendors. For example sequences for selected memory devices, see Example LUT sequences.

41.3.9.1 Dual-die flash memories

Certain serial flash memory vendors provide dual-die packages that are essentially two devices (dies) stacked within the same package to increase the memory capacity of a single package. These two devices within a package share the same data and clock pins, but have individual chip selects. The following figure shows the dual-die package and the naming conventions used in this document. For simplicity, the data pins are shown to be unidirectional.

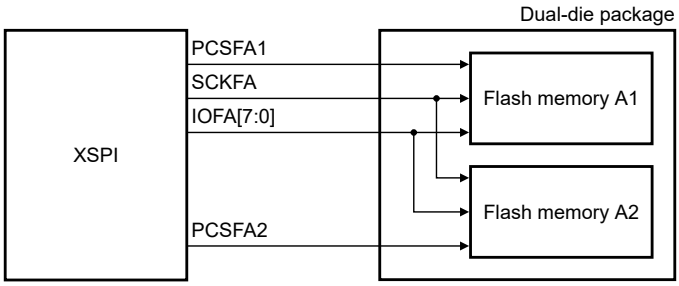


Figure 148. Generic configuration for dual-die support

Because the two devices within a package share the same I/O pads, they cannot function in the parallel mode.

41.3.10 Sampling of serial flash memory input data

41.3.10.1 Basic description

XSPI is used to read data from the serial flash memory device. Depending on the actual implementation, there is a delay between the internal clocking in the XSPI module and the external serial flash memory device. See the following figure for an overview of this scheme.

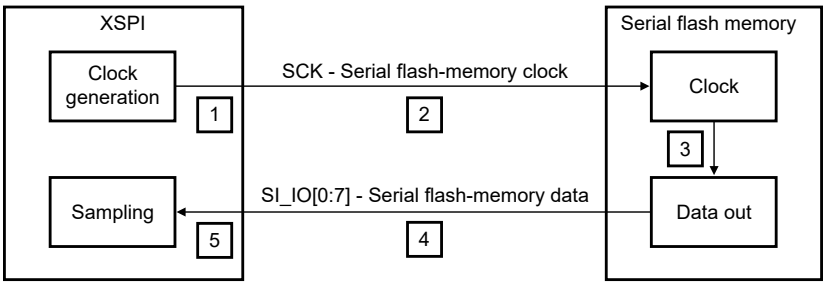


Figure 149. Serial flash memory sampling clock overview

The rising edge of the internal reference clock is taken as timing reference for the data output of the serial flash memory. After a time of t_{total_delay} the data arrives at the internal sampling stage of the XSPI module. Considering the figure provided here, the following parts of the delay chain contribute to t_{total_delay}:

1. Output delay of the serial flash memory clock for the device containing the XSPI module
2. Wire delay of PCB from the device containing the XSPI module to the external serial flash memory device
3. Clock to data out delay of the external serial flash memory device, including input and output delays
4. Wire delay of PCB from the external serial flash memory device to the device containing the XSPI module
5. Device delay corresponding to the input data

NOTE

The t_{total_delay} is specific to the characteristics of the actual implementation.

41.3.10.2 DQS sampling method

41.3.10.2.1 Basic description

In the DQS mode, the data strobe signal (DQS/RWDS) is used to sample the read data. Here, both DQS and the data sent by the flash memory move in the same direction; therefore, it is relatively easier to achieve at higher frequencies.

When using DQS for SDR reads, XSPI internally samples the incoming data on the rising edge of the strobe signal. FSPHS=0 only.

The next figure shows the sampling read data in the SDR mode using the DQS.

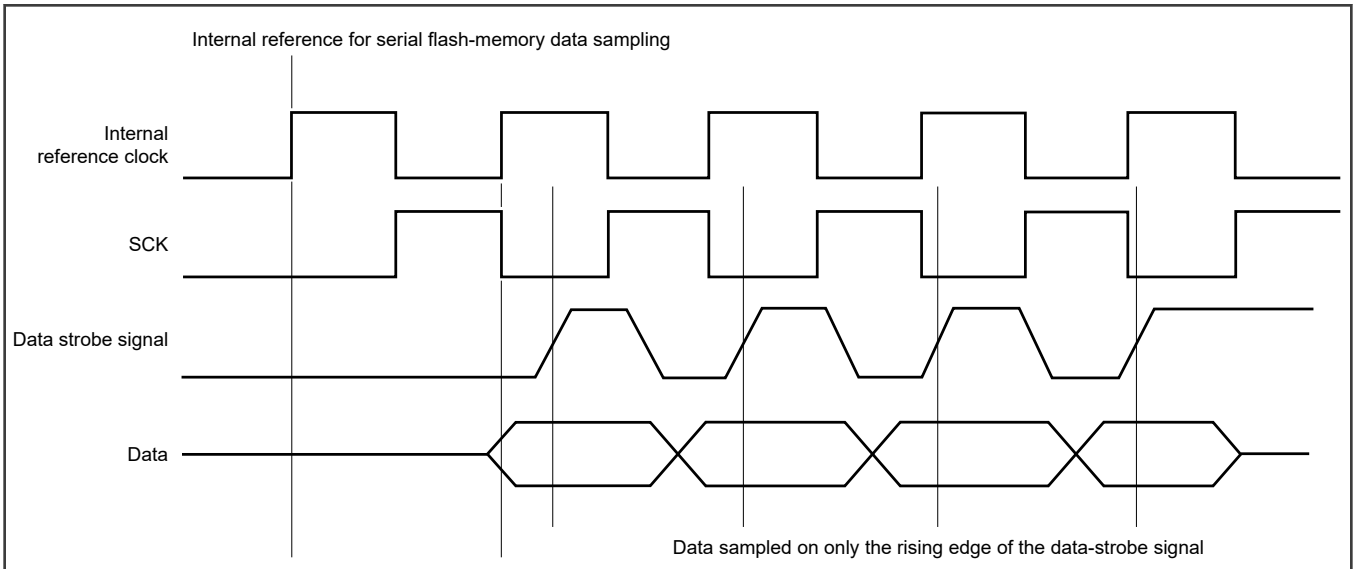


Figure 150. Data strobe functionality in SDR mode for read operation

NOTE

Consider "Data Strobe Signal" as "Data Strobe Signal driven by memory" and "Data" as "Data from memory".

NOTE

Refer to the Datasheet for specific timing waveforms of XSPI

When using the DQS for DDR reads, XSPI internally samples the incoming data on both the edges of the strobe signal. See the next figure for details.

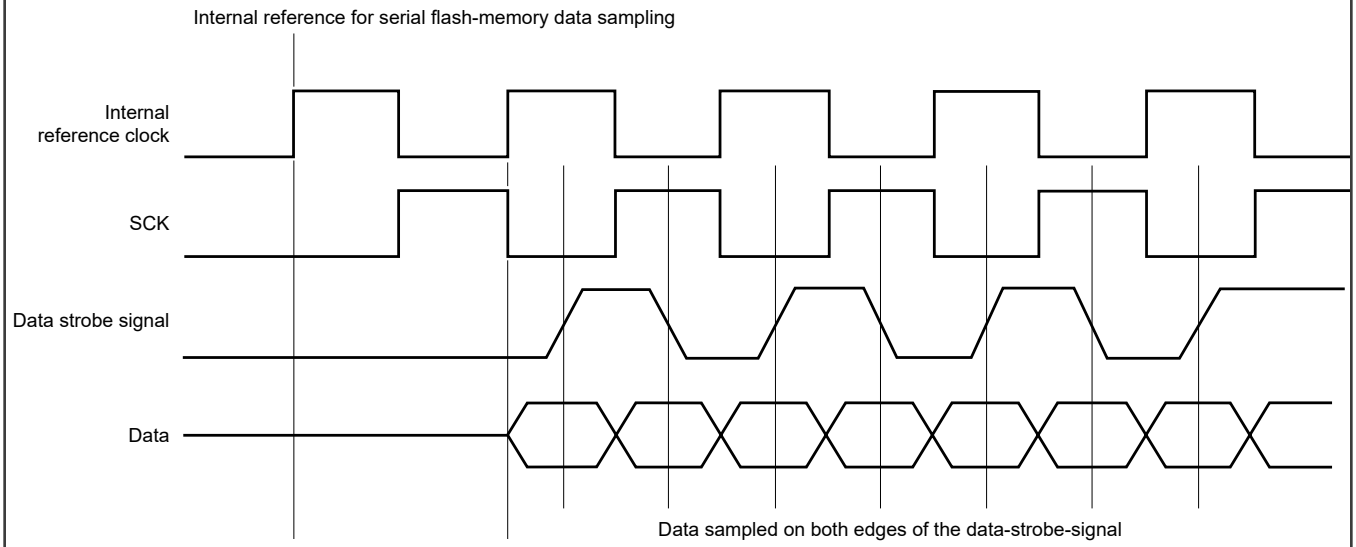


Figure 151. Data strobe functionality in DDR mode for read operation

NOTE

Consider "Data Strobe Signal" as "Data Strobe Signal driven by memory" and "Data" as "Data from memory".

NOTE

For Specific details - Refer to the Data Sheet specification of XSPI module.

41.3.10.2.2 External DQS

In serial flash memories supporting DQS, the data strobe signal is an output from the flash memory device that indicates when data is being transferred from the flash memory to the host controller. The data is then captured by the controller on:

- Only one edge (either rising or falling edge) of DQS signal in SDR mode
- Both rising/falling edges of the DQS signal in DDR mode

Some serial flash memories (for example, HyperFlash) provide the data strobe output (RWDS) with latency cycles included in between the ongoing read transactions. The data is sampled on both the edges of this signal taking into consideration that when no signal is provided by the flash memory between ongoing transactions, data is not sampled then, and at the end of the transaction, the required number of data to be fetched remains the same. See the next figure for details.

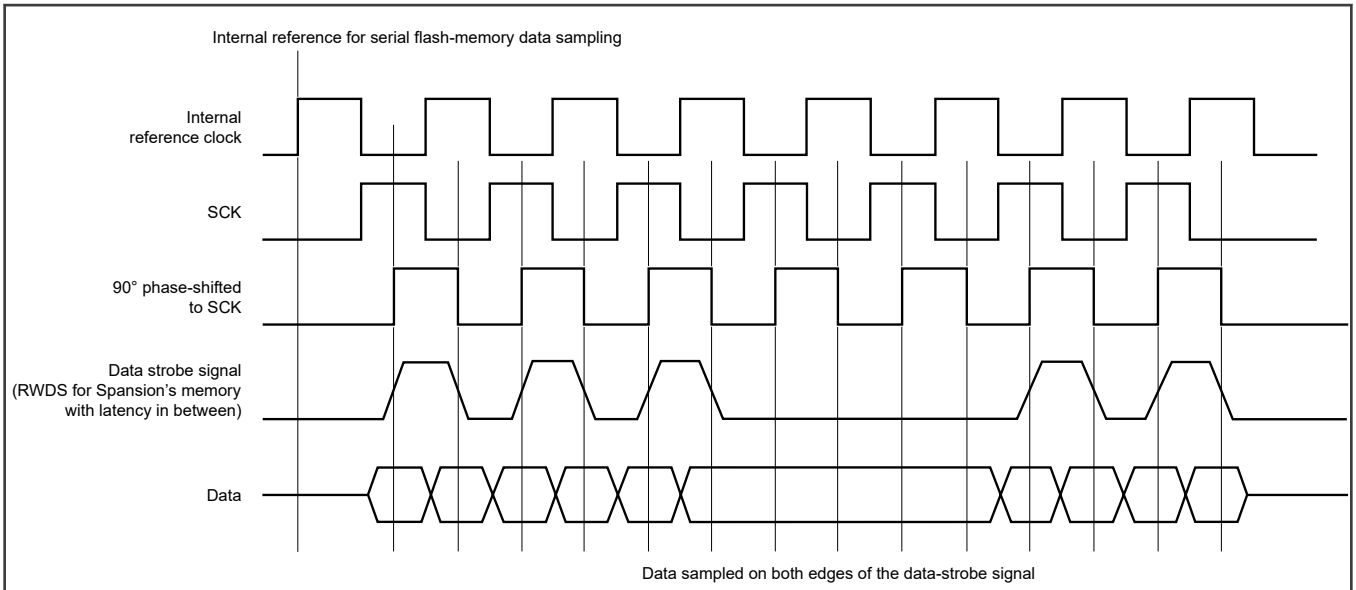


Figure 152. Data strobe functionality with latency included

External DQS sampling mode can be enabled by programming MCR[DQS_FA_SEL] = "11" for flash memory A. This mode supports the following configurations:

- Both high/low frequency delay chain manual programming using DLLCRA[SLV_DLY_COARSE] and DLLCRA[SLV_FLINE_OFFSET]
- DLL-assisted sampling only using DLLCRA[DLLEN]
- DLL-assisted auto data learning using DLLCRA[DLLEN] along with MCR[DLPEN]

41.3.10.2.3 Dummy Pad loopback

In this mode, the internal clock is loop-backed from a dummy internal pad to compensate data pad delays. This can be enabled by configuring the value of MCR[DQS_FA_SEL] as "01" for flash memory A. This mode can be used with the following configuration:

- High/low frequency delay chain manual programming in bypass mode using DLLCRA[SLV_DLY_COARSE] and DLLCR[FREQEN].
- DLL-assisted sampling only using DLLCRA[DLLEN]

NOTE

Refer to DS or "Chip-specific section" of XSPI for Fixed sampling Tap specific settings/details with or without DLL (in bypass mode)

- DLL-assisted auto data learning using DLLCRA[DLLEN] and along with MCR[DLPEN]

NOTE

Refer to Data learning with DLL for further details.

41.3.10.2.4 DQS pad loopback

This mode is similar to internal DQS but the internal clock is loop-backed through the DQS pad to compensate data pad delays. This can be enabled by configuring the value of MCR[DQS_FA_SEL] as "10" for flash memory A. This mode can be used with the following configuration:

- High/low frequency delay chain manual programming using DLLCRA[SLV_DLY_COARSE] and DLLCR[FREQEN].
- DLL-assisted sampling only using DLLCRA[DLLEN]

- DLL-assisted auto data learning using DLLCRA[DLLEN] and along with MCR[DLPEN]

NOTE

- To support DQS pad loopback scenario, if any External DQS pad is used instead of Dummy pad, then that External DQS pad should not be connected to the board (left floating) to avoid any electrical interference.

41.3.11 DLL and delay chain usage

The DLL is a general-purpose, dynamically adaptive clock delay module. It provides the ability to select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT). The DLL is suitable for applications where accurate delay adjustment is required, such as in case of DDR/SDR (with Data-Learning) and support for "DLL based fixed Tap sampling" etc.

41.3.11.1 DLL working concept

The DLL control loop consists of a counter, reference delay line, and phase detector which operate on the reference clock (ref_clk) input. The reference clock (ref_clk) is fed into the reference delay line. After reset, a single delay tap is selected for individual delay line or reference delay chain. A phase detector is used to detect the condition where a half shift of reference clock has occurred. In addition to this, signals are generated internally to either increment or decrement the counter, which controls the delay of reference delay line to identify the half-phase detection of reference clock. Any changes in the delay of the individual elements of the delay chain (because of PVT) automatically cause the phase detector logic to determine if a change in the counter value is required. After the half-phase shift is detected, an internal lock signal is generated, status is captured in DLLSR[DLLA_LOCK] and after a lock event occurs, the target delay chain is ready to be updated based on a triggering event.

41.3.11.2 DLL configuration

NOTE

See AC specifications in chip data sheet for individual functional modes like DDR, SDR, QUAD, OCTAL, etc.

Programmability	Description	Values
MCR[DDR_EN]	Enables DDR mode.	See "XSPI configurations" table in chip data sheet.
MCR[DQS_FA_SEL]	Selects the DQS clock for sampling read data at the flash memory A XSPI port.	See "XSPI configurations" table in chip data sheet.
MCR[DLPEN]	Enables the data learning mechanism.	See "XSPI configurations" table in chip data sheet.
FLSHCR[TDH]	Specifies which clock to use for data alignment.	See "XSPI configurations" table in chip data sheet.
FLSHCR[TCSH]	Specifies the CS hold time in serial flash memory clock cycles.	See "XSPI configurations" table in chip data sheet.
FLSHCR[TCSS]	Specifies the CS setup time in serial flash memory clock cycles.	See "XSPI configurations" table in chip data sheet.
DLLCRA registers:		
DLLCRA[DLLEN]	IO timing closure with DLL or without DLL.	See "XSPI configurations" table in chip data sheet.
DLLCRA[FREQEN]	IO timing closure with low or high DLL.	See "XSPI configurations" table in chip data sheet.

Table continues on the next page...

Table continued from the previous page...

Programmability	Description	Values
DLLCRA[DLL_CDL8]	Support up to 2x/3x PVT variation.	See "XSPI configurations" table in chip data sheet.
DLLCRA[DLL_REFCNTR]	DLL Reference Counter	See "XSPI configurations" table in chip data sheet.
DLLCRA[DLLRES]	Resolution margin	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_FINE_OFFSET]	Fine Offset Delay Elements	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_DLY_OFFSET]	T/16 Offset Delay Elements	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_DLY_COARSE]	Delay Elements in Delay Tap. Usage: DLLCRA[SLV_DLL_BYPASS] = 1	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_DLY_FINE]	Fine Adjustment Delay Elements Usage: DLLCRA[SLV_DLL_BYPASS] = 1	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLAVE_AUTO_UPDT]	Subordinate Chain Auto-Update Enable	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_EN]	Subordinate Enable	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_DLL_BYPASS]	Subordinate DLL Bypass	See "XSPI configurations" table in chip data sheet.
DLLCRA[SLV_UPD]	Subordinate Update	See "XSPI configurations" table in chip data sheet.
SMPR[DLLFSMPFA]	DLL Nth Tap selection for Data sampling	See "XSPI configurations" table in chip data sheet.
SMPR[FSDLY]	Selects the delay for full-speed commands for loopback DQS.	See "XSPI configurations" table in chip data sheet.
SMPR[FSPHS]	Selects the edge of the sampling clock valid for full-speed commands.	See "XSPI configurations" table in chip data sheet.

41.3.11.3 Target delay chain programming sequence

Programming sequence of Target delay chain for DLL Bypass mode and DLL Auto Update mode is explain in sub-sections.

41.3.11.3.1 DLL Bypass mode

Following is the programming sequence for DLL Bypass mode:

NOTE

For DLL minimum frequency support value, see the AC specifications in chip data sheet. Any low frequency application with respect to DLL minimum frequency support condition need to be closed in DLL bypass mode (DLL disable).

1. Program DLLCRA[SLV_EN]=1, DLLCRA[SLV_DLL_BYPASS]=1, and DLLCRA[SLAVE_AUTO_UPDT]=0.
2. Program the following fields to provide the desired DQS delay for sampling: DLLCRA[SLV_FINE_OFFSET], DLLCRA[SLV_DLY_COARSE], DLLCRA[SLV_DLY_FINE], and DLLCRA[FREQEN]. See the AC specifications (XSPI configurations) in chip data sheet for supported frequency/DLL functional mode.
3. Program DLLCRA[SLV_UPD]=1 to load these values in the target delay chain.
4. Check the target delay chain update status by polling DLLSR[SLVA_LOCK]=1.
5. Clear DLLCRA[SLV_UPD] after confirming the update state.

41.3.11.3.2 DLL Auto Update mode

Following is the programming sequence for DLL Auto Update mode:

NOTE

For DLL minimum frequency support value, see the AC specifications in chip data sheet. Any high frequency application with respect to DLL minimum frequency support condition need to be closed in DLL Auto Update mode (DLL enable).

1. Program DLLCRA[SLV_EN]=1, DLLCRA[SLV_DLL_BYPASS]=0, and DLLCRA[SLAVE_AUTO_UPDT]=1.
2. Program the DLL configuration by using DLLCRA[DLL_REFCNTR] and DLLCRA[DLLRES]. See the AC specifications (XSPI configurations) in chip data sheet for supported frequency/DLL functional mode.
3. Program the target settings to delay DQS by using these fields: DLLCRA[SLV_FINE_OFFSET], DLLCRA[SLV_DLY_OFFSET], and DLLCRA[FREQEN]. See the AC specifications (XSPI configurations) in chip data sheet for supported frequency/DLL functional mode.
4. If offset delay needs to be updated on the target chain, program DLLCRA[SLV_UPD]=1.
5. Enable DLL by programming DLLCRA[DLLEN]=1 and reset DLLCRA[SLV_UPD]=0. Target delay chain is updated automatically and can be checked by polling DLLSR[SLVA_LOCK]=1

41.3.12 Data learning

Auto Data learning is used to manage minimum data valid windows (t_{DVW}) from the flash memory as well as skew variations in the chip based on PVT conditions for both SDR and DDR modes. XSPI provides this feature via the DATA_LEARN instruction for all flash memories, irrespective of whether the flash memory supports it. The DATA_LEARN instruction accepts an operand that defines the number of bits of the known pattern for which the data learning has to be done. The known pattern is provided in the DLPR register. Data learning feature is supported for both SDR and DDR modes with PAD loopback DQS (MCR[DQS_FA_SEL] = 01b or 10b).

NOTE

Data learning feature is not supported if sampling clock source is flash provided External DQS (MCR[DQS_FA_SEL] = 11b).

During DATA_LEARN instruction, XSPI will compare the sampled data bits with internal data learn pattern (DLPR) and determine the correct sampling clock phase. The phase selection is automatically updated after executing DATA_LEARN instruction and applied to following Read instructions or sequences.

XSPI controller generates both 08 posedge clock phases (Phase 0 ~ Phase 7) and 08 negedge clock phases (Phase 0 ~ Phase 7) by DLL Target delay line for the selected sample clock (MCR[DQS_FA_SEL]: PAD loopback DQS). Clock Phase 0 is actually the input sample clock (DQS_IN) with or without any static delay based on DLLCRx configuration.

There are 16 sampling flops implemented (8 for each clock edge) for each Data IO line in one sampling unit based on below skew assumptions.

- Skew on each Data IO line need to be balanced to each sampling flop[i]
- Skew on DQS tap[i] to be balanced with each other to sample flop[i]
- DQS tap[i] : i^{th} tap to be balanced for all 8/16 Data IO lines

To accommodate pessimistic data skew between the different IO lines, you must program a different learning pattern on each of the data pins IO1 and IO3 through DLPR[DLPV] and (DLCR[DLP_SEL_FA] = 1b or 11b).

If DLL-assisted auto data learning using DLLCRA[DLLEN] along with MCR[DLPEN] is enabled, DATA_LEARN instruction causes XSPI to ignore the sampling configuration (SMPR) and find the correct sampling point (clock phase) by comparing the data learning pattern. The sample clock phase selected could be polled by register field DLSR_FA[POS_EDGE] and DLSR_FA[NEG_EDGE]. If data learning failed, there will be a status flag (DLSR_FA[DLPFF]) and interrupt bit set (FR[DLPFF]).

Auto data learning is supported with PAD loopback DQS only through MCR[DQS_FA_SEL] register settings. To start data learning, follow the below sequence:

1. Configure a known pattern in the Data Learn register (if programmable) inside flash memories that support data learning Preamble pattern. The pattern should be selected to have multiple low and high transitions in the data bus.
2. Set up XSPI DDR data read sequence, inserting a DATA_LEARN instruction before the READ_DDR instruction. The flash memory returns the data learning pattern in between dummy cycles and subsequent read command.

   Example sequence:

   CMD -> ADDR_DDR -> DUMMY -> DATA_LEARN -> READ_DDR -> STOP

   NOTE

   Set DATA_LEARN instruction operand bit[7] =1 to select DDR mode

   - 1 - DDR mode
   - 0 - SDR mode

   NOTE

   For flash memories that does not support data learning Preamble pattern:

   a. Configure a known location[fixed address] in flash memory with a known pattern. The pattern should be selected to have multiple low and high transitions in the data bus.
   b. Configure the known pattern in DLPR[DLPV].
   c. Set up a DATA_LEARN sequence with fixed address (memory) where Preamble data pattern is stored.
   d. Set up a XSPI DDR data read sequence by utilizing JMP_TO_SEQ command after executing the DATA_LEARN sequence.

      Example sequence:

      CMD -> CMD[fixed Address] -> DUMMY -> DATA_LEARN -> JMP_TO_SEQ[Index]

      [index]: CMD -> ADDR_DDR -> DUMMY -> READ_DDR -> STOP

   e. See Programming the data learning pattern in flash memory for details on how the Preamble pattern has to be ordered in memory for correct operation.
3. Initiate a read operation either through AHB or IPS interface.
4. XSPI reads the data from the flash memory. It then encounters the DATA_LEARN instruction and samples the incoming data on both edges (rising and falling) of the DQS.

   NOTE

   If the data from the flash memory does not match the data learning pattern, the FR[DLPFF] flag is set.

5. Hardware algorithm will select the best-suited sampling point based on the incoming data learning pattern using (03) three-tap algorithm.

This feature might be used to auto-calibrate the sampling point for DDR reads, and auto-calibration might be triggered depending on a change in PVT conditions. In case the sampling point (clock phase) needs to be changed based on initiating options (such as DLL lock status or software forced interrupt), data learn instruction has to be re-initiated.

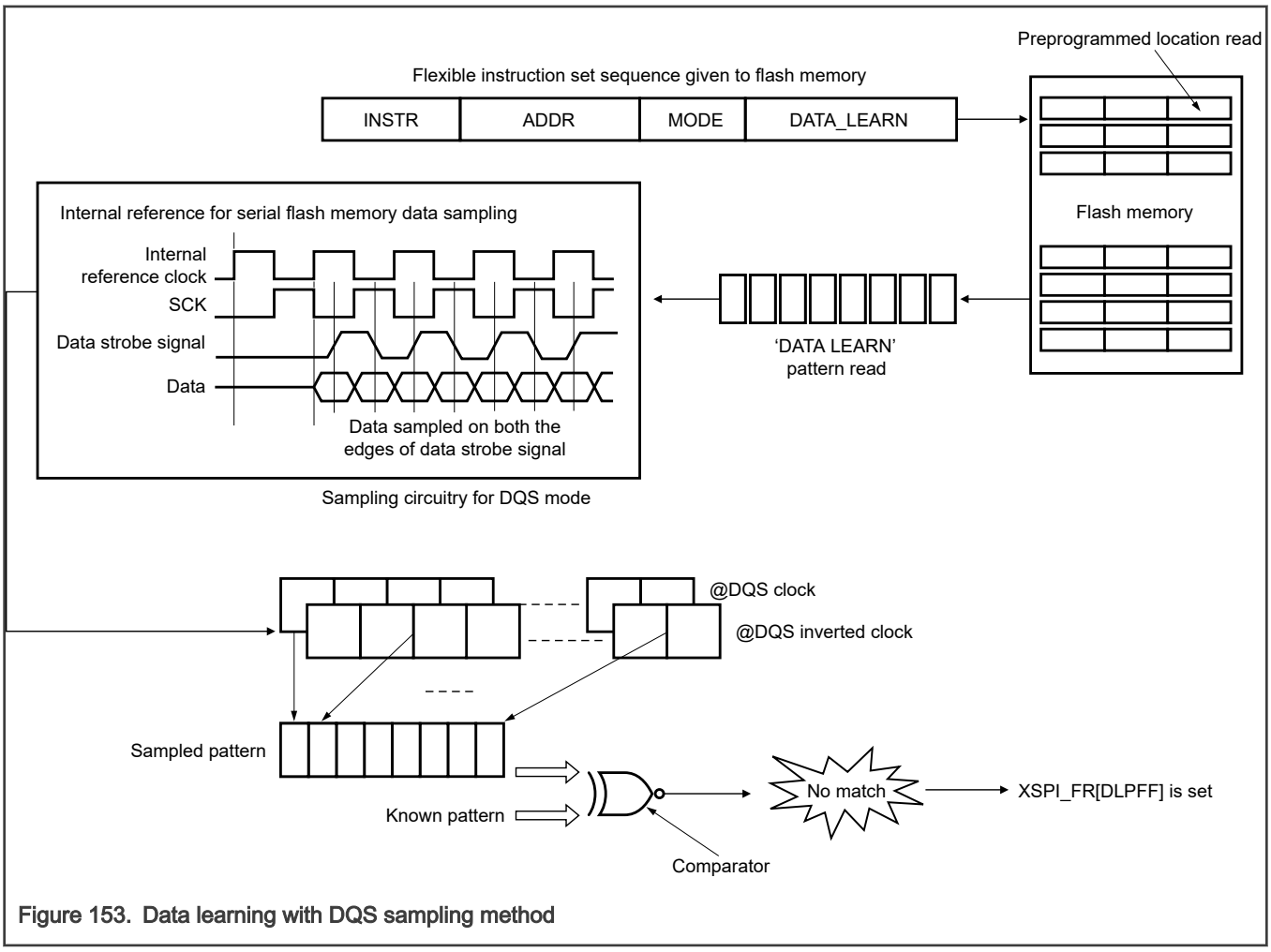


Figure 153. Data learning with DQS sampling method

41.3.12.1 Programming the data learning pattern in flash memory

Example scenario: If the value of DLPR[DLPV] is 43h, XSPI tries to match this sample on every incoming data line for single, dual, quad, or octal configurations. The following table shows the data programmed in flash memory with endianness taken into consideration.

Table 347. Programming the data learning pattern in flash memory

Endianness	Single IO	Dual IO	Quad IO	Octal IO
BE	43h	300Fh	0F0000FFh	00FF00000000FFFFh
LE	43h	0F30h	FF00000Fh	FFFF0000000000FFh

41.3.13 Data input hold requirement of flash memory

In DDR mode, the data is valid only for half clock cycle. It is difficult to meet the data input hold time requirement for flash memory. XSPI internally delays the data sent to flash memory so that it becomes easy to meet the hold requirement. The FLSHCR[TDH] is

used for this purpose. If FLSHCR[TDH] is configured to 0, then the data sent to flash memory is aligned with the internal reference clock of XSPI. If it is 1, then the data is aligned to 2x internal reference half clock.

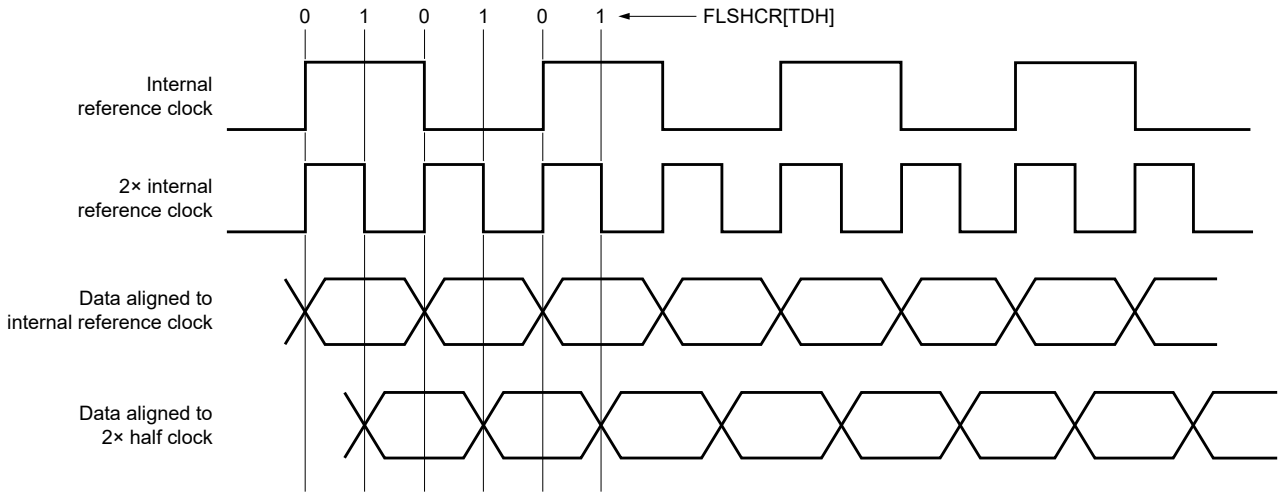


Figure 154. Data hold

41.3.14 Inline PRINCE encryption and decryption (IPED) CTR

IPED is a PRINCE-based encryption-decryption IP with a 64-bit block cipher and 128-bit key. XSPI supports automatic encryption on IP and AHB write commands, and automatic decryption on AHB read commands.

NOTE

- Read/write sends to IPED.i_data should always be 64 bits (8 byte).
- For AHB based read, start address will always be 8 bytes aligned by hardware and BUFnCR[ADATSZ] should be N*8 Byte aligned by application.
- For write, start address will always be 8 bytes aligned and should be handled by software.

Changing the register setting of IPEDCTRL changes the behavior of IPED.

Below register setting should be applied to use IPED feature:

- BFGENCR[ALIGN]: This configuration is needed to stop the AHB prefetch at the given boundary crossing 01-256 B, 10-512 B, and 11-1 KB

There are maximal sixteen contexts supported by XSPI to use different 64bits IPED.i_iv to encrypt/decrypt data. The sixteen ranges are controlled by IPEDCTX*START and IPEDCTX*END registers. There should be no overlap between these sixteen ranges; otherwise, the behavior of IP is undefined. Any single one read/write request to flash should not cross one context boundary. For AHB read/write it can be kept by hardware automatically by setting BFGENCR[ALIGN]. The IP command write boundary should be maintained by software. Any read/write across context boundary will lead to wrong data to/from external device.

Context 15 is reserved for OTA based region, and it is accessible by IPS transactions.

The write ability of IPEDCTX*n*START and IPEDCTX*n*END is controlled by IPEDCTXCTRL0 and IPEDCTXCTRL1. By default, after power on reset, IPEDCTX*n*START and IPEDCTX*n*END cannot be modified. For details see the truth table below:

Table 348. IPEDCTX*n*START and IPEDCTX*n*END write ability

IPEDCTXCTRL0[CTX*n*_FREEZE0]	IPEDCTXCTRL1[CTX*n*_FREEZE1]	CTX*n*_FREEZE0 and CTX*n*_FREEZE1 writable	IPEDCTX*n*START and IPEDCTX*n*END writable
10b	10b	Yes	Yes
01b (default)	01b (default)	Yes	No
don't care	not 10b?	No	No
00b or 11b?	don't care	No	No

41.3.15 IPED - Bypass mode specific data-size limitations

1. IPED - Bypass mode is enabled through static software programming by disabling the IPEDCTRL[IPED_EN], IPEDCTRL[IPED_XEX_EN], IPEDCTRL[IPWR_EN], and IPEDCTRL[AHBWR_EN]: 4 B data size restriction applied on non-encrypted regions based on IPED.
2. IPED - Bypass mode is targeted based on dynamic switching between GCM/CTR(XEX)/Bypass context regions: 8 B data size restriction applied on non-encrypted regions based on IPED.

41.3.16 PRINCE IP GCM function support

Now there are 8/16 region of PRINCE which have separated IV/AAD that are defined by start and end addresses. Each region's read write control is defined by IPEDCTXCTRL0 and IPEDCTXCTRL1 similar as before.

Ctx 15 is reserved for OTA based region and it is accessible by IPS transactions only when IP_WR_OTA is enabled.

41.3.16.1 GCM READ

Every AHB read request from SOC will be aligned to 32 Bytes boundary automatically by IP, and ADATSZ should be N*32 Byte aligned.

Software must guarantee that the RX buffer size for initiator that will issue GCM region read must be multiple times of 32 Bytes.

It is because PRINCE IP's authentication check is based on 32 Bytes of data. So any valid read must be the unit of 32 Bytes.

IP will trigger the converted read address and read size based on the rule that every 32 Bytes of raw data = 32 Bytes of encrypted data + 8 Bytes tag.

NOTE

In GCM AHB read transactions, for wrap bursts with the unaligned start address, the burst size should be 32 Bytes.

Below is key software setting and sequences to conduct correct AHB GCM read:

- Set

  IPEDCTRL[AHBGCMRD] = 1

  IPEDCTRL[IPED_EN] = 1

  IPEDCTRL[CONFIG] = 1/0 depends on test vectors
- Write IPEDCTX0IV0 = lower 32 bits of IV

  Write IPEDCTX0IV1 = higher 32 bits of IV
- Write IPEDCTX0START = start address and IPEDCTX0START[GCM] = 1
- Write IPEDCTX0END = end address
- Write IPEDCTX0AAD0 lower 32 bits of AAD
- Write IPEDCTX0AAD1 higher 32 bits of AAD

If current external device has boundary limitation (like pSRAM has 1 KB limitation) below additional setting is needed

- BFGENCR[ALIGN]: This configuration is needed to stop the AHB prefetch at the given boundary crossing 01-256B, 10-512B, and 11-1KB.

If PRINCE IP indicates there is an authentication check fail, the entire 32 Bytes will be discarded, and error defined value will be put on to AHB bus. Also FR[IPEDERR] interrupt will be set.

41.3.16.2 GCM AHB Write

IP also supports AHB GCM write. But write start address should be 32 bytes aligned, and size should be multiple times of 32 bytes, it is not maintained by IP.

Any GCM write request violating this request will lead to undefined behavior.

All software setting are the same as GCM AHB read except IPEDCTRL[AHBGCMWR] = 1.

If the current external device has a boundary limitation (such as pSRAM has 1KB limitation) then below additional setting is needed:

- BFGENCR[ALIGN]: This configuration is needed to stop the AHB prefetch at the given boundary crossing 01-256B, 10-512B, and 11-1KB

41.3.16.3 GCM IP command write

IP support IP command write with PRINCE IP GCM.

Application needs to align Payload size to 32B boundary with the IPED-GCM block.

Software sequence: (program size = 256 bytes as example)

- Set

  IPEDCTRL[IPED_EN] = 1

  Set IPEDCTRL[CONFIG] = 1/0 depends on test vectors
- Write IPEDCTX0IV0 = lower 32 bits of IV

  Write IPEDCTX0IV1 = higher 32 bits of IV
- Write IPEDCTX0START = start address and IPEDCTX0START[GCM] = 1
- Write IPEDCTX0END = end address
- Write IPEDCTX0AAD0 lower 32 bits of AAD

  Write IPEDCTX0AAD1 higher 32 bits of AAD
- Set IPEDCTRL[IPGCMWR] = 1

NOTE

1. AHB frequency must be greater than flash frequency to avoid underrun with PSRAM devices, when x16 mode and DDR mode with IPS IPED is present.
2. Flash programming should be done before enabling static bits of IPEDCTRL.
3. Watermark should be defined by considering TAG, that is:

$$TBCT[WMRK] = TX_FIFO_DEPTH - (\frac{IPCR[IDATSZ]x1.25}{4} - 1)$$

Equation 4. Watermark calculation

   Watermark is programmed by polling IPEDCTRL[IPS_GCM], to check the GCM context region.
4. Application needs to follow the below configuration sequence when the page boundary crosses on physical address in the case of IPS GCM:
   a. Set AWRCR[PPW_WR_DIS] = 1b for flash devices that require program wait period, to enable the auto RDSR mechanism. See Page program wait handling for more configuration details.
   b. Initiate a locked transfer by setting SFP_TG_IPCR[ARB_LOCK].
   c. When the transaction crosses the page boundary, hardware will stop the current transfer and initiate a new transfer to flush the residue data either immediately (PSRAM/HyperRAM devices) or after waiting for T_{pp}(NOR Flash devices).
   d. Poll FSMSTAT[STATE] = 00b to confirm that the write transfer is completed.
   e. Software needs to wait for T_{pp} (NOR Flash devices) after residue data transfer and release the ARB Lock.

In IPS GCM mode, application must guarantee that IPCR[IDATSZ] should be in multiple of 32 Bytes, if application is unable to maintain this, XSPI will not generate trigger to flash even if correct Watermark Programming TBCT[WMRK]) is done and initiator timeout (MTO) will occur and ERRSTAT[TO_ERR] is set to 1b. Once ERRSTAT[TO_ERR] is set to 1b, it indicates that incorrect programming is done, as XSPI is unable to generate trigger to flash, so to allow external IPED to come in IDLE state, main environment needs to clear the IPED through IPEDCTRL[IPED_SWRESET] and needs to release IPEDCTRL[IPED_SWRESET] before any environment gets grant. This will prevent unwanted hang situation in XSPI.

Early abort does not arise if we follow the below assumptions:

Assumptions:

1. Transaction max size = TX buffer size (1KB) of XSPI (IPS payload data + GCM tag). Software can write 128B of data in TX buffer in a single shot. For payload size of more than 128B, XSPI needs a waiting period after writing 128B to avoid the IPED GCM Encryption buffer overflow condition.
2. Water mark = IPS payload data + GCM tag.

NOTE

If IPCR[IDATSZ] is more than TBCT[WMRK] (256 is programmed in TBCT[WMRK]) then it waits for the threshold to cross before writing SFAR and IPCR registers to XSPI to avoid TX async FIFO underrun.

41.3.16.4 IPED XEX

IPED XEX is an XEX-based memory encryption block with several configuration options.

Read/write sends to IPED.i_data should always be 64 bits(8 byte).

For read, start address will always be 8 bytes aligned and fetch size is also 8 bytes aligned.

For write, start address will always be 8 bytes aligned and should be handled by software.

XEX logic is controlled by parameter IP_XEX_EN, when IP_XEX_EN will be there only then XEX based encryption will be enabled.

Software setting:

- IPED function control IPEDCTRL[IPED_XEX_EN] - Drive IPED XEX interface i_enable

NOTE

Rest of the programming will be same as CTR based encryption.

Configuration supported:

Data Prince 12 Rounds

Tweak Prince 8 Rounds

Configurable Pipeline

Enhanced Timing for Data

i_rnd shall be tied to logic zero by integration team

41.3.16.5 Context Register Programming

CTR and XEX Context Register settings

Context region size = N * Page size

GCM Context Register setting

Context region should be define in such a way that after defining context_n range start and end address, 20% for TAG data should be kept as holes. Rest of the configuration is defined above.

NOTE

- Context comparison is done with application address.
- Any encrypted area should be defined after 1K boundary to avoid encryption of the flash configuration register data.

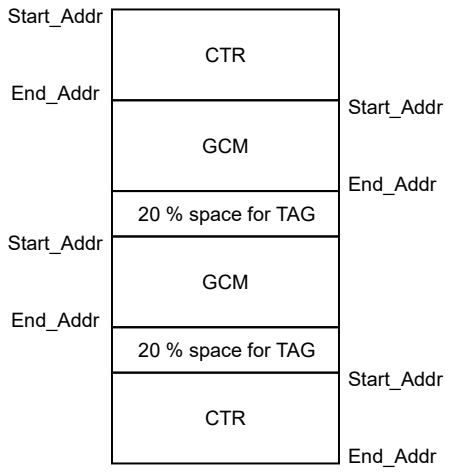


Figure 155. Context programming

41.3.16.6 Reset sequence with IPED

1. XSPI should be enabled (MCR[MDIS] = 0), when asserting MCR[SWRSTHD] and MCR[SWRSTSD].
2. Write 1 to MCR[SWRSTHD], MCR[SWRSTSD], and MCR[IPS_TG_RST] to assert the resets.
3. Assert IPEDCTRL[IPED_SWRESET].

NOTE

Assert IPEDCTRL[IPED_SWRESET], only when MCR[SWRSTHD], MCR[SWRSTSD], and MCR[IPS_TG_RST] are already asserted.

4. Wait for MCR[IPS_TG_RST] to be 0 which means software reset for IPS target queue is completed.
5. Wait until IPEDCTRL[IPED_SWRESET] is completed.
6. Disable XSPI with writing 1 to MCR[MDIS] before de-asserting MCR[SWRSTHD] and MCR[SWRSTSD].
7. Clear MCR[SWRSTHD] and MCR[SWRSTSD]. Clear IPEDCTRL[IPED_SWRESET].
8. Enable XSPI with writing 0 to MCR[MDIS].

41.3.17 Secure flash protection

This secure flash module enforces access control policies based on the ID, privilege and secure attributes of the command transactions. All accesses throughout the flash device need to be monitored to determine their validity. If a transaction from a given IPS initiator has appropriate access rights, it is forwarded to flash, else the access is denied and an error is generated.

There are two levels of checks present in this module, based on MDAD and FRAD descriptors programmed. First level checks the input transactions based on secure attributes and ID associated with that transaction, which are checked according to MDAD. Second level check selects the appropriate FRAD based on address range of transaction. Each FRAD allows the transaction to pass through only when secure, privilege and ID related attributes are matching the criteria set. If either check level fails, the transaction is not forwarded and IPS bus error/interrupt is generated. Flash regions can be programmed to cover the entire address space to provide a default set of access permissions. For flash read transactions only first level of MDAD check is enabled.

Software can bypass both or anyone of these checks by writing to MGC[GVLDFRAD], MGC[GVLDMDAD], or MGC[GVLD] fields.

The IPS initiators must write on SFP register (SFP_TG_SFAR, SFP_TG_IPCR, SFP_TG_SUB_n_SFAR, and SFP_TG_SUB_n_IPCR registers) addresses for generating transfers to flash based on the TG queue that is being targeted for transfer. If the transaction passes both MDAD and FRAD checks then it is passed through to XSPI, otherwise a bus error or interrupt is generated. There is only a single IPS target interface which can be used for XSPI IPS register access and SFP IPS register access.

XSPI SFP related register sets are present here.

41.3.17.1 MDAD

These descriptors form the first level of checks. There are 2'd2 target group queues present in SFP block and each of these queues can be programmed with different set of MDAD descriptor checks in TG*n*MDAD registers. These registers are under access control and can only be programmed by initiators with privilege attributes. These descriptors must be programmed prior to the start of any flash transaction else any IP command initiated flash transaction will generate an IPS bus error.

Multiple transactions can be queued in target groups in parallel, and one of them is launched towards XSPI after the arbitration.

41.3.17.1.1 MDAD checks

- Secure check – This is decided by the TG*n*MDAD[SA] field of the MDAD descriptor. If TG*n*MDAD[SA] = 01b then the queue is reserved for only non-secure transactions. If TG*n*MDAD[SA] = 10b then the queue is reserved for only secure transactions. If TG*n*MDAD[SA] = 11b then this check is bypassed.
- Initiator ID check – This check compares the initiator ID of the transaction with the MID match value set in the TG*n*MDAD[MIDMATCH] field and allows only matching transactions inside the queue. There are 6 mask fields (TG*n*MDAD[MASK]) inside the descriptor. If the mask type (TG*n*MDAD[MASKTYPE]) is set to 0 then the input transactions initiator ID is ANDed with the mask bits before comparing with the MID match value. If the mask type is set to 1 then the input initiator ID is ORed with the masks bits and then compared with the MID match bits. Multiple initiator IDs can be allowed inside the queue using this mask.

When an IPS write is done to TG queue SFAR and IPCR registers (Serial Flash Memory Address (SFP_TG_SFAR), IP Configuration (SFP_TG_IPCR),) then SFP block checks the attributes with MDAD descriptor of that queue. If the attributes match with the descriptor checks it is written inside that queue, else an IPS transfer error is generated and that entry is discarded. The

queue is considered locked once the IPCR SEQID fields are written. So, software should ensure that IPCR SEQID bits are written only after SFAR and other IPCR bits like DATSZ are written in that queue. The queue will be unlocked when flash transfer is completed for that transaction.

Once a initiator ID is assigned to a queue then transactions with only that initiator ID are allowed inside the queue. The data in the queue is cleared once the flash transaction is finally completed by the XSPI.

The data path for initiating IP-command transaction with SFP is shown in the following figure.

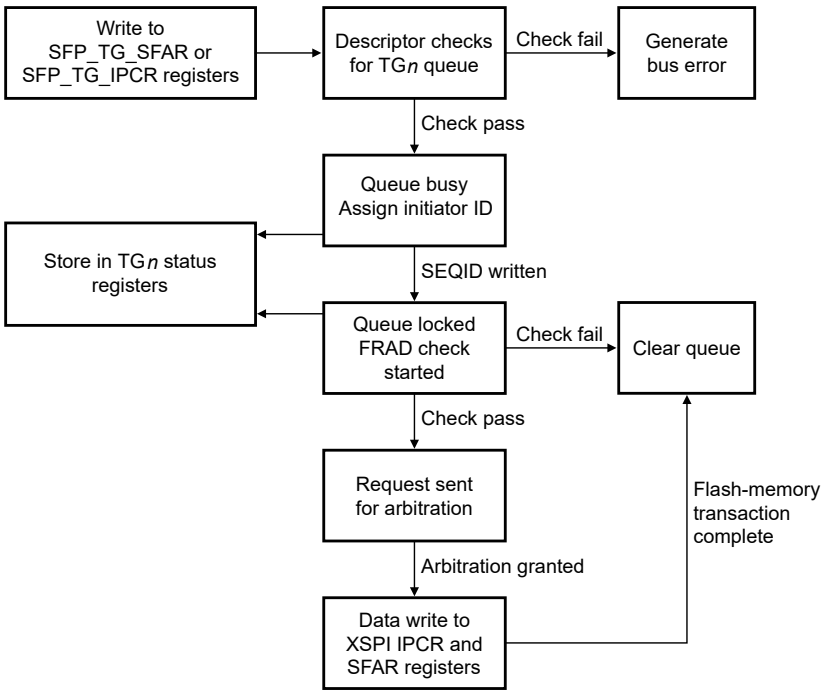


Figure 156. Data path for initiating IP- command flash transaction through SFP

If the SFAR write passes first level of MDAD checks then the value is written in SFP register TG*n*SFAR, valid bit is set in TG*n*SFARS register and queue is considered as busy. Similarly when IPCR write passes the checks, its value is written into TG*n*IPRCS register, TG*n*IPRCS[VLD] field is set to 1 and queue is considered as locked. It is mandatory that IPCR is written only after SFAR has been written for a queue and queue is busy. If IPCR is written before SFAR then it will result in IPS transfer error. As soon as queue is locked (IPCR SEQID written) and passes FRAD checks, the transaction is sent for arbitration. See IP and AHB command arbitration for more details.

NOTE

TBDR register write is not allowed until the SFP_TG_SFAR and SFP_TG_IPCR entries of that TG queue have passed MDAD and FRAD checks and the transaction is granted the arbitration. If TBDR registers are written before arbitration win, transfer error will be generated. IPS initiator can read FSMSTAT register to check the status of their transaction. TBDR access is allowed when FSMSTAT[VLD] field is set and for the initiator whose initiator ID matches with bits [13:8] of this register, and bits [1:0] are set to 01 or 10.

41.3.17.1.2 MDAD error conditions

The error conditions for first level of MDAD checks and the response is given in table below. These error conditions can occur when SFP_TG_SFAR, SFP_TG_IPCR or TBDR registers are being written.

NOTE

Queue is considered busy if it has been written successfully one time with any of the values SFAR, IDATSZ or PAR. Queue is considered locked when SEQID has been written successfully by passing MDAD checks.

Table 349. MDAD check error conditions

Error condition	Response	Common/Queue error
Queue descriptor SFP_TG_MDAD is not programmed	IPS transfer error is generated	Common error
Target group queue is locked and SFAR/IPCR written again	IPS transfer error is generated	Common error
Queue busy and same IPS initiator writes SFAR or IPCR registers with wrong security attribute	IPS transfer error is generated and queue is cleared (SFAR and IPCR needs to be written again for transfer to start).	Queue error
Queue is locked and any IPS initiator writes	IPS transfer error is generated	Common error
If initiator ID of transaction does not qualify MID check or fails the attribute checks of the queues	IPS transfer error is generated	Common error
TBDR register is written before SFAR and IPCR SEQID qualifies MDAD checks	IPS transfer error is generated	-
If none of FRAD descriptors are programmed	IPS transfer error is generated	Common error
IPCR is written before SFAR	IPS transfer error is generated	Common Error - If the transaction doesn't matches the security attribute check or MID check of any MDAD. Queue error - If it matches security and MID attribute of the queue.

When a common error is generated its details are latched in IPS Error (IPSERROR) or register and corresponding error field is set to 1. This register will keep the last error transaction latched unless cleared by writing 1 to CLR field of this register. In case a queue is busy and same IPS initiator writes IPCR or SFAR with wrong security attributes the queue is cleared, the Target Group SFAR Status (TGSFARS) valid field goes low. In this case the IPS initiator should write the SFAR an IPCR again for proper transaction to start.

In case of queue error the details are latched in Target Group IP Configuration Status (TGIPCRS) or Target Group SFAR Status (TGSFARS) registers. These registers will also keep the error status latched until cleared by writing 1 to CLR field of respective registers.

41.3.17.2 FRAD

These descriptors form the second level check for IPS flash transactions. IPS read transaction are not processed by this level of check and are passed through after MDAD checks. There are 8 FRADs which can be programmed by initiators with privilege attributes. So the access checks can be programmed for up to 8 unique address ranges. Each of the address range is equipped with different access attribute checks and based on this write access can be controlled for the flash region. These descriptors must be programmed before starting transactions on flash.

After programming the FRAD*n*_WORD0, FRAD*n*_WORD1, FRAD*n*_WORD2 and FRAD*n*_WORD3, registers, FRAD*n*_WORD3[VLD] field should be written 1. SFP block doesn't consider the FRAD descriptors which are not valid (FRAD*n*_WORD3[VLD] field is not set).

There is a descriptor lock feature to prevent spurious changes to the FRAD. FRAD*n*_WORD3[LOCK] can be programmed to make all FRAD*n* registers as read only or to allow write access to all or some initiators with specific initiator ID.

41.3.17.2.1 FRAD checks

- Address range checks – Start and end address for FRAD region are programmed in FRAD*n*_WORD0 and FRAD*n*_WORD1 registers. Start address of IPS flash transaction written in SFP_TG_SFAR register and data size programmed in SFP_TG_IPCR register are used to check if the transaction lies within any of the FRAD address ranges. This address range match will be used to select the corresponding FRAD from which other access checks will be done. The start and end address are programmed around 64 KB boundary so 16 MSB are only used for range check.

  The data size should not be programmed as 0 in SFP_TG_IPCR register for normal data transfers to flash and actual size should be programmed. In the cases where only flash instruction has to be sent and no data has to be written in TBDR register, this DATSZ can be programmed as 0. If DATSZ is programmed as 0 in SFP_TG_IPCR, SFP module blocks the write access to TBDR register, and generates IPS transfer error.

- Exclusive access lock – Exclusive IPS initiator access lock can be enabled over any flash address defined under a FRAD. The initiator ID of the IPS initiator which enables/owns the exclusive access is captured in FRAD*n*_WORD2[EALO] fields. Once enabled, the exclusive lock can be released only by the same IPS initiator which enabled the lock by writing the FRAD*n*_WORD3[EAL] field. The exclusive write access permissions of a flash region are decided as given in table below:

Table 350. Exclusive access lock

FRAD*n*_WORD3[EAL]	Write permissions
00	No lock. The write permissions are decided as set in FRAD*n*_WORD2[MDnACP] fields for respective IPS initiator domain.
10	Write permissions are revoked for all IPS initiators. Any write transaction coming to this flash address region will not be forwarded through and will result in an error.
11	Write permissions are revoked for all IPS initiators except the initiator ID matching the ID specified by FRAD*n*_WORD2[EALO] fields.

- Flash region privilege checks – If the FRAD*n*_WORD3[EAL] fields are set to 00 then flash write permissions for any IPS initiator is decided based on the settings done in FRAD*n*_WORD2[MDnACP] fields as given in table below. One of the FRAD*n*_WORD2[MDnACP] is selected for permission checks based on which target group queue the transaction arrives from.

Table 351. Flash region privilege checks

MDxACP value	Privilege access	Secure access	Write access
111	No	No	Not allowed
	No	Yes	Not allowed
	Yes	No	Allowed
	Yes	Yes	Allowed
110	No	No	Allowed
	No	Yes	Allowed
	Yes	No	Allowed
	Yes	Yes	Allowed
101	No	No	Not allowed
	No	Yes	Not allowed
	Yes	No	Not allowed
	Yes	Yes	Allowed

Table continues on the next page...

Table 351. Flash region privilege checks (continued)

MDxACP value	Privilege access	Secure access	Write access
100	No	No	Not allowed
	No	Yes	Allowed
	Yes	No	Not allowed
	Yes	Yes	Allowed
0xx	-	-	Not allowed

If any transaction fails FRAD check then error interrupt is generated if enabled.

FRAD status registers keep the details of last transaction which lie within the address range of that respective FRAD and will be updated only when a new transaction comes which corresponds to same FRAD address range.

Flash Region Compare Address Status register stores the flash start address that was compared and Flash Region Compare Status Data contains the initiator ID, privilege and secure attributes of the transaction. The transaction details are stored in FRAD register for which the address compare check passed. These status registers are not updated in case of read transfers or when FRAD check is disabled. For example, if the transaction address lies in the range set in FRAD2_WORD0 and FRAD2_WORD1 then the transaction details will be stored in FRAD2_WORD4 and FRAD2_WORD5 registers and FRAD2_WORD5[CMPVALID] will be set. If transaction fails the security or privilege attribute check for the corresponding FRAD, then FRAD*n*_WORD5[CMP_ERR]. This bit and other transaction details will remain latched unless cleared by writing '1' to ERRSTAT[FRAD*n*ACC].

If transaction address does not match any of the FRAD address range then also error interrupt is generated but the details will not be stored in any of the FRAD status registers.

FRAD checks are done only for flash write transactions, flash read transactions are passed through to XSPI if MDAD checks are passed. At least one FRAD region must be made valid (if the FRAD check is enabled from MGC register) for flash read instructions to go through even though FRAD check is not done for read instructions. SFP module identifies the transfer as read/write based on the command programmed in Initiator Read Command (MRC) register. If the sequence selected in LUT for IP transfer has any one of the command from MRC register the whole sequence is considered as read, else it is considered as write. There are two command codes which are predefined READ_DDR and READ commands. Software can program two other command code as per its convenience by writing into fields [21:16] or [29:24] and then writing respective valid bit as 1.

For any type of Flash Read command in which payload size is not required, that transaction will be treated as 'Write' transaction by SFP block. In this case MRC register can't be used to define it as a 'Read' instruction. Certain examples for these type of commands are - In SNAND Flash, PAGE READ (‘h13), PAGE READ CACHE RANDOM (‘h30) etc.

41.3.17.2.2 Error conditions

Any transaction passing the MDAD checks will be checked for FRAD checks and can lead to following error scenarios.

Table 352. Error conditions

Error condition	Response
None of the FRAD are programmed	IPS transfer error will be generated and status will be latched in IPS_ERROR register
Transaction address doesn’t fall within any of the FRAD address range	Error interrupt will be generated if enabled
Transaction match with multiple FRAD address range	Error interrupt will be generated if enabled. Error bit will be set, and status will be stored in FRADx_WORD5 status registers for whom the address check passed

Table continues on the next page...

Table 352. Error conditions (continued)

Error condition	Response
Transaction passes FRAD address check but fails MDxACP checks or if SEQID is not allowed to be executed from that FRAD by SFP_LUT_EN register.	Error interrupt will be generated if enabled. Error bit will be set, and status will be stored in FRADx_WORD5 status registers for whom the address check passed
Transaction passes FRAD address check but range was locked by EAL bits	Error interrupt will be generated if enabled. Error bit will be set, and status will be stored in FRADx_WORD5 status registers for whom the address check passed
Read transaction	No FRAD check will be done and no error will be generated for FRAD failure. But if FRAD check is enabled in MGC then at least one FRAD must be valid else it will give error during SFAR/IPCR write

NOTE

There are different error registers corresponding to each TG queue (IPSERROR, IPSERROR_SUB_n, FRAD*_WORD5, FRAD*_WORD5_SUB_n). These registers will be updated according to the TG queue from which transaction entered.

41.3.17.2.3 LUT access enable

SFP block can control which SEQID is executed through which of the target group queues. This gives more control for limiting any crucial security related instructions to some specific target group. These access controls can be defined in LUT Access Enable (SFP_LUT_EN0 - SFP_LUT_EN1). There is one register corresponding to each target group queue. Each register contains 16 bits corresponding to 16 locations of the LUT. Whenever a transaction passes through the MDAD checks and is presented for FRAD checks, the SEQID written in IPCR register is checked with corresponding bit in LUT Access Enable (SFP_LUT_EN0 - SFP_LUT_EN1). If the bit is set to '0' it will pass through FRAD checks if MDxACP permissions are also present. Else it will cause a FRAD access error. This LUT access check is applicable only for write transactions if FRAD checks are enabled in Initiator Global Configuration (MGC).

41.3.17.3 TBDR register write lock

SFP module keeps the TBDR write permissions locked for IPS transactions by default. When a IPS initiator writes SFAR and IPCR transactions and this transaction passes MDAD and FRAD checks then the accesses are unlocked for that initiator ID. Lock and unlock procedure is shown with an example in figure below:

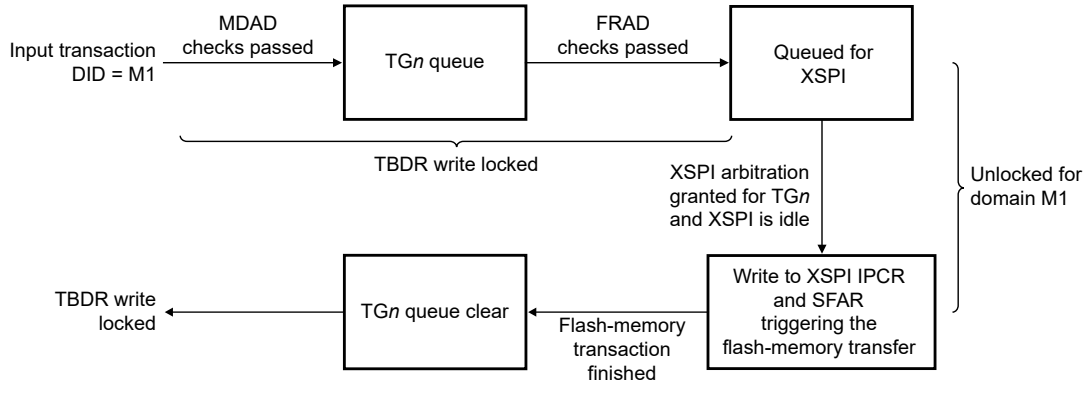


Figure 157. TBDR lock and unlock procedure

If any IPS initiator writes on TBDR register when it is not unlocked for that initiator ID it will result in IPS transfer error. FSMSTAT[MID] indicates which initiator ID is currently active, and if the value of FSMSTAT[STATE] is 01 or 10 and FSMSTAT[VLD] is set as 1, then the TBDR lock is open for write transactions. It is advisable that for write transactions

where IDATSZ is non zero, initiator clears the TX buffer from any residue data and program the TBCT[WMRK] watermark after its request is granted in arbitration (FSMSTAT bit [1:0] are 01). In case of transfers having IDATSZ programmed as 0 the TX Buffer Data (TBDR) register remains locked and data can't be written to this register.

In case DMA is being used to fill the TBDR buffer, program RSER[TBFDE] field to 1 only when write access is granted and TBDR access is unlocked after checking FSMSTAT[STATE] field. Once DMA completes the TBDR write, this field is set back to 0. After this, the SFP will write SFAR and IPCR to XSPI triggering the flash transfer. It should be made sure that the DMA should have same initiator ID as the TG, which won the arbitration.

NOTE

RBCT[WMRK] can only be programmed when no IP command is triggered on XSPI (SR[IPACC] is 0). So It is not possible to program two different RBCT[WMRK] for two different TG queues if both are used in parallel. SFP triggers the read transfers to flash as soon as TG queue gets grant and XSPI is IDLE. So it is recommended to use a common RBCT[WMRK] value for both TG queues or change the value when both the queues are empty.

41.3.17.4 Transaction status registers

SFP module has two registers (FSM Status (FSMSTAT) and Flash Sequence Request (FLSEQREQ)) which show the current and last transaction status. These registers are updated only after MDAD and FRAD check are passed and arbitration is granted for this request. See IP and AHB command arbitration for details.

- FSM Status (FSMSTAT) register shows the request for the target group queue has been granted by the XSPI arbiter and FSMSTAT[VLD] is set. If FSMSTAT[CMD] is set to 1 then it is a non-read instruction sequence and it is set to 0 then it means that this is a read sequence. FSMSTAT[MID] shows the initiator ID from which this transaction is generated. FSMSTAT[STATE] shows the current state of active transaction when valid FSMSTAT[VLD] is set.

  The figure below shows the IPS transaction flow inside SFP block along with various values of bits [1:0]:

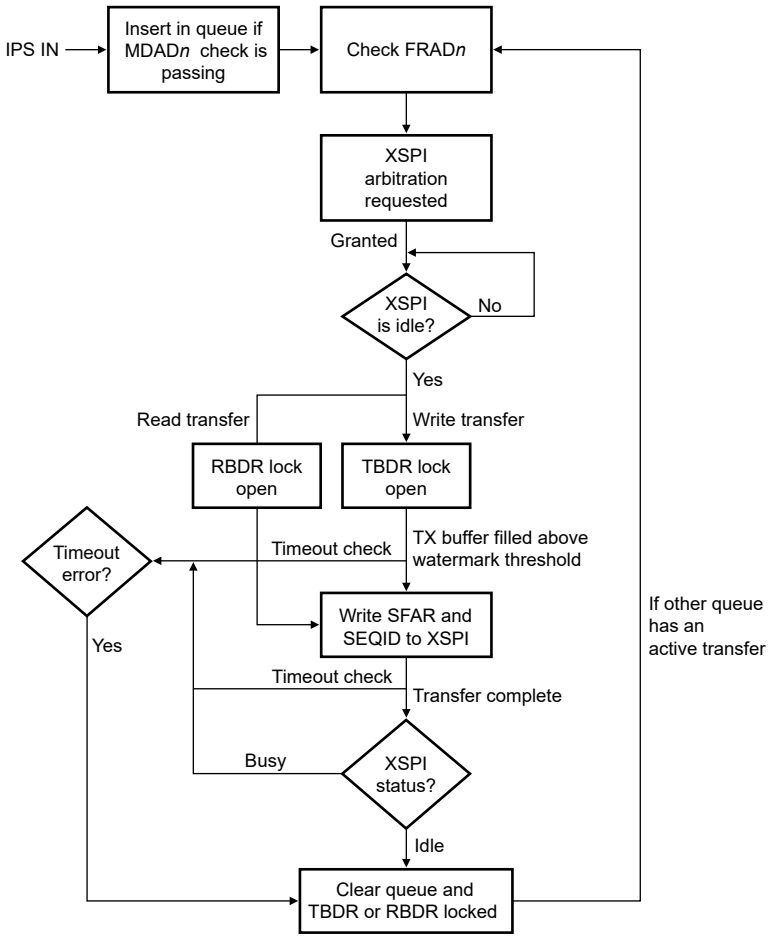


Figure 158. Transaction status

NOTE

1. If IPCR[DATSZ] is more than the TX buffer watermark threshold value (256- Watermark programmed in TBCT) then it waits for threshold to cross before writing the SFAR and IPCR registers to XSPI. Else SFP block waits for one entry to be written to TBDR register and after that it writes the SFAR and IPCR to XSPI thus triggering the transfer. In case of read transfers or if the datasize is zero, SFAR and IPCR are triggered as soon as grant is received.
2. XSPI is considered IDLE if transaction is finished at flash, No DMA and interrupt pending for following FR fields: TBUF, RBDF, IUEF, ILLINE. RX data is completely read.

FSM state values in FSM Status (FSMSTAT) for different states are in the following table:

Table 353. FSMSTAT register field details

FSMSTAT register fields	
FSMSTAT[VLD]	0 - No IPS transfer is present in TG queues or request is send to XSPI arbiter for grant. 1 - IPS transfer is granted by the XSPI arbiter or running in XSPI
FSMSTAT[STATE]	00 - Transaction request is granted but XSPI is not IDLE due to any previous DMA transaction ongoing, any residue data left in RDBFL or if any interrupt is pending (FR[TBUF], FR[RBDF], FR[ILLINE]).

Table continues on the next page...

Table 353. FSMSTAT register field details (continued)

FSMSTAT register fields	
	01 - TBDR lock is open. XSPI considers IPS transfer and IPS initiator can write TBDR. 10 - TX buffer filled above threshold. Write transfer is triggered. SEQID is written to XSPI. 11 - Read transfer is triggered. SEQID is written to XSPI.

For flash write transfers which pass MDAD and FRAD checks and XSPI is not busy with any previous transfer, TBDR lock is opened and FSMSTAT[STATE] shows value of 01. After the IPS initiator fills the TX buffer till the level where free entries in TX buffer are less than the watermark threshold set in TX Buffer Control (TBCT) and SR[TXWA] flag is de-asserted, SFP block writes the SFAR and IPCR entries to XSPI registers and transaction is triggered. To avoid underrun, program the watermark accordingly for size of the data to be transferred. This programming can be done when arbitration is granted. TBCT Watermark can be programmed as 256-(IDATSZ/4-1) in most of the scenarios. If IPCR[IDATSZ] is less such that TX buffer can never be filled till watermark threshold, in such cases SFP block does not wait for the TBCT[WMRK] to trigger and writes the Serial Flash Memory Address (SFAR) and IP Configuration (IPCR) registers as soon as one data entry is written to TX Buffer Data (TBDR) register. In case of flash memory read transfers SFAR and IPCR entries are written to XSPI as soon as XSPI is IDLE and FSMSTAT[STATE] is set to 11. FSMSTAT[VLD] goes to 0 once the IPS transfer is completed and XSPI is IDLE. IPS initiator should clear the TX buffer before writing to TBDR register to ensure that no data from previous transaction is left in TX buffer. Also if any read data is left in the buffer from previous transfer, then it should also be cleared by writing '1' to MCR[CLR_RXF] field.

For any IPS initiated RW transaction which needs two separate transactions at flash interface, like Write-Enable transaction followed by Write-data transaction (Data-Learn flash transaction followed by Read-data transaction) at flash interface must be joined together with JMP_TO_SEQ instruction.

- Flash Sequence Request (FLSEQREQ) register holds the details of transaction which was last sent to XSPI and completed. After a flash transaction is completed this register is updated and its valid field (FLSEQREQ[VLD]) is set to 1. This register holds the command type (read or write), transaction initiator ID (FLSEQREQ[REQ_MID]), target group queue from which this transaction passed in field (FLSEQREQ[REQ_TG]), privilege and secure attributes for the transaction, FLSEQREQ[SEQID] and number of flash descriptor FRAD within whose address range the transaction lies (FLSEQREQ[FRAD]). This flash descriptor number always remains 0 in case of read transfers.

  This register also contains a timeout error field FLSEQREQ[TIMEOUT] which is set when a timeout error occurs, and transaction is aborted. When the error occurs this register keeps the status latched unless it is cleared by writing 1 to ERRSTAT[TO_ERR].

41.3.17.4.1 XSPI IDLE conditions

XSPI is considered IDLE if the following conditions are all true:

- No ongoing AHB/IP transfer.
- No DMA transaction.
- No residue data left in RDBFL after read transfer.
- None of these interrupt types are enabled and pending: FR[TBUF], FR[RBDF], FR[ILLINE]

After the TG queue is granted, SFP waits for XSPI to be in IDLE state before triggering any transfer over to XSPI. Also, once the transaction has finished on external memory, SFP again waits for XSPI to reach IDLE condition before considering the transaction as complete and clearing the queue. If XSPI remains BUSY, it will lead to SFP timeout error generation.

NOTE

Any TG queue which doesn't want to read the complete RX buffer must clear it after the read to avoid timeout error generation and to avoid exposing this data to another TG queue.

41.3.17.5 Timeout error

Initiator Timeout (MTO) register can be programmed with a timeout value above which the transaction should be aborted. When the transaction passes MDAD and FRAD level checks and request is granted by XSPI arbiter and XSPI is IDLE, this counter is started working on IPS/system clock. If the transaction is not completed and XSPI does not become IDLE before this timeout counter reaches the value programmed, the transaction is considered aborted by SFP and an error interrupt is generated, if enabled. The error bit is also set in Flash Sequence Request (FLSEQREQ) register along with the transaction details. After the timeout error, next transaction is requested for arbitration, which was queued in another target group queue once SR[BUSY] flag goes low.

NOTE

Timeout counter doesn't starts in case queue was granted but XSPI was not in IDLE condition. So any environment on TG queue, after receiving the grant should track FSMSTAT[STAT] bits. If they remain in 00 even after receiving the grant then it should take some corrective action to bring XSPI to IDLE state and start the transfer.

41.3.17.6 Arbitration lock

The arbitration lock feature is used to lock the arbitration with a particular queue and can be used for handling the page program wait requirement for some external flashes. This lock can be requested by main environment (EENV0) by writing '1' into SFP_TG_IPCR[ARB_LOCK] field. Once the transaction passes both the checks (MDAD and FRAD), wins the arbitration and transfer is completed, this lock will be granted. If the TG queue had requested the arbitration lock and was granted but TG SW reset came before transfer gets completed, then also the arbitration lock will be granted. After this, the transaction from the other queue or any AHB access will not be granted by driving HREADY=0 and HREADY-timeout counter is started unless the arbitration is unlocked by previous queue. The main environment (EENV0) which locked the arbitration can unlock it by writing '1' to SFP_TG_IPCR[ARB_UNLOCK] field.

NOTE

The user environment always needs to request the arbitration lock.

Example scenario for handling IP commands resulting in page program wait period.

1. Write the Serial Flash Memory Address (SFP_TG_SFAR) with the required address.
2. Write required transaction attributes in IP Configuration (SFP_TG_IPCR) with SFP_TG_IPCR[ARB_LOCK] field set as '1'.
3. Complete the transfer after request is granted by XSPI.
4. Arbitration lock is received on completion of transfer.
5. Wait for approx time required for page program wait period to end.
6. Generate a flash status register read transfer (for checking WIP bit) with SFP_TG_IPCR[ARB_LOCK] set as '1'.
7. If the WIP bit is read back as '0' it means page program wait period is over (this is applicable for NOR Flash, for other devices; respective settings need to be followed).
8. Generate another flash status register read transfer (for checking WIP bit) with SFP_TG_IPCR[ARB_UNLOCK] set as '1'.
9. This will unlock the arbitration.

There is a arbitration lock timeout counter which starts once the transaction has finished and arbitration is locked for a certain queue. Once this counter reaches the value set in SFP Arbitration Lock Timeout Counter (SFP_ARB_TIMEOUT) register it will generate an interrupt and error bit is latched in ERRSTAT[ARB_LOCK_TO] field. Once the arbitration lock timeout is set, the arbitration can not be unlocked using normal process as shown above. Then clearing this error field (ERRSTAT[ARB_LOCK_TO]) will unlock the arbitration.

NOTE

- In case of TX Buffer residue data left (due to wrong programming) by the IPS initiator and AHB write was pending during that time then TX Buffer residue data can interfere with AHB write data. To avoid this interference, the arbitration lock feature should be used for all IPS transactions, so that TX Buffer can be cleared before providing AHB write grant.

41.3.17.7 Error Interrupt Enable

INT_EN allows software to disable or enable the interrupt signal generation in case of any of the listed conditions.

- Time out error (INT_EN[TO_ERR])
- IPCR register write error (INT_EN[TG*n*IPCR])
- SFAR register write error (INT_EN[TG*n*SFAR])
- IPS common error (INT_EN[IPS_ERR])
- FRAD access error (INT_EN[FRAD*n*ACC])
- No FRAD address range match error (INT_EN[FRADMTCH])
- Lock register write error (INT_EN[LCK_ERR_IE])
- Arbitration win interrupt (INT_EN[ARB_WIN_IE])
- Arbitration lock timeout interrupt (INT_EN[ARB_TO_IE])

If the respective field is set to 1 for any of the error conditions, then the interrupt signal is set to 1 when that error occurs and it remains set until the error is cleared by writing 1 to error clear bit as listed in register descriptions.

Other than these interrupts SFP can also generate interrupt on following conditions:

1. Arbitration win interrupt (INT_EN[ARB_WIN_IE]).
2. Arbitration lock timeout interrupt (INT_EN[ARB_TO_IE]).

41.3.17.8 Register lock access error

There are some lock fields present in SFP registers which can lock the write access to some of the registers. When these fields are locked and any write access occurs on these registers error fields is set in ERRSTAT[LOCK_ERR] and interrupt is generated if enabled. The register address which was accessed is captured in SFP_LOCK_ERR_ADDR register. The first address captured will remain latched unless the error bit is cleared by writing '1' to ERRSTAT[LOCK_ERR].

The list of registers and their corresponding lock bit are listed in table below:

Table 354. Lock error details

Register name	Lock fields
FRADn_WORD0, where n = 0 to 7	FRADn_WORD3[LOCK]=01b
FRADn_WORD1, where n = 0 to 7	FRADn_WORD3[LOCK]=01b
FRADn_WORD2 where n = 0 to 7	FRADn_WORD3[LOCK]=01b
TGnMDAD, where n = 0 to 7	TGnMDAD[LCK]=1b
INT_EN	INT_EN[LCK]=1b
LUTn, where n=0 to 79	MGC[GCLCK]=01b,10b,11b
BFGENCR	MGC[GCLCK]=01b,10b,11b
SFP_LUT_ENn, where n=0 to 1	SFP_LUT_ENn[LOCK]=1b or MGC[GCLCK]=01b

Table continues on the next page...

Table 354. Lock error details (continued)

Register name	Lock fields
MGC	MGC[GCLCK]=01
MRC	MGC[GCLCK]=01
MTO	MGC[GCLCK]=01
SFP_ARB_TIMEOUT	MGC[GCLCK]=01

41.3.18 Soft reset considerations

In case the soft reset is being used for XSPI, software should take some considerations so that correct functionality is maintained. MCR[IPS_TG_RST] field can be used to reset all TG queues and their error and status registers. If any TG queue was granted and transaction was ongoing towards the flash then XSPI waits for that transaction to complete on flash and chip-select de-assertion before triggering this IPS TG reset and clearing the TG queues and related logic. If no IP command was executing on the flash then this soft reset is immediately executed. IPS_TG_RST only resets the SFP logic related to TG queues it doesn't affects any XSPI native registers. AHB SW reset (MCR[SWRSTSD]) resets the complete AHB related logic, XSPI internal IP command logic and also clears the TX and RX buffers. It also resets the AHB buffer pointers hence invalidating the data in AHB buffer. AHB soft reset during an ongoing AHB read transaction, may lead to write data corruption or read data mismatch. Flash domain SW reset (MCR[SWRSTHD]) resets the state machine and logic on the flash domain side of XSPI.

NOTE

It is advisable that all three reset are done at same time (IPS_TG_RST, SWRSTHD, and SWRSTSD) to clear the complete XSPI transaction path. Using these resets independently may lead to some side-effects.

The AHB soft reset should be used when no AHB transaction(WRITE/READ) is pending on AHB interface. This is to ensure that data loss/corruption does not happen. The main environment should ensure that the pending AHB transaction needs to be re-issued if AHB soft reset is asserted during a pending AHB transaction. SR[AHBTRN] should be used to check if there is any pending request on the AHB interface. If AHB soft reset is asserted during an ongoing AHB transaction, the main environment should also ensure that the ongoing transaction needs to be re-issued. SR[AWRACC] and SR[AHB_ACC] should be used to check if there is any ongoing Write or Read transaction on the AHB interface. Software should take care of the page program wait period while doing soft reset. Doing IPS or AHB reset doesn't affects the arbitration lock status. Hence if any TG queue has locked the arbitration it will remain locked even after IPS_TG_RST has been asserted. After the soft reset if no IPS transactions are done then there will be a SFP arbitration lock timeout and clearing that timeout error will unlock the arbitration. This is done so that in case any previous transfer done has resulted in page program period on flash then we don't allow any AHB or IPS transfer to go through after the soft reset.

In case of AHB write transfers if the AHB and flash domain soft reset is asserted before the transfer is completed on the flash then it may result in no PPWF flag being set. In this case, SW should ensure that no transaction is done on XSPI if the page program wait period on flash was ongoing. If the soft reset is asserted after the transaction has completed on the flash and the PPWF flag is set then this flag remains set even after soft reset completion. RDSR read will start once the soft reset has been de-asserted.

In case of IPED, IPEDCTRL[IPED_SWRESET] need to be asserted with software reset for AHB (MCR[SWRSTHD]), serial flash memory domain (MCR[SWRSTSD]), and IPS TG queue (MCR[IPS_TG_RST]).

If SR[AW_SWRST] is not set after soft reset, software must use the last HADDR, which was written before soft reset assertion, to determine which write bursts must be re-issued.

NOTE

Software must adhere to IPED address boundary rules while determining which AHB write bursts to re-issue after soft reset de-assertion.

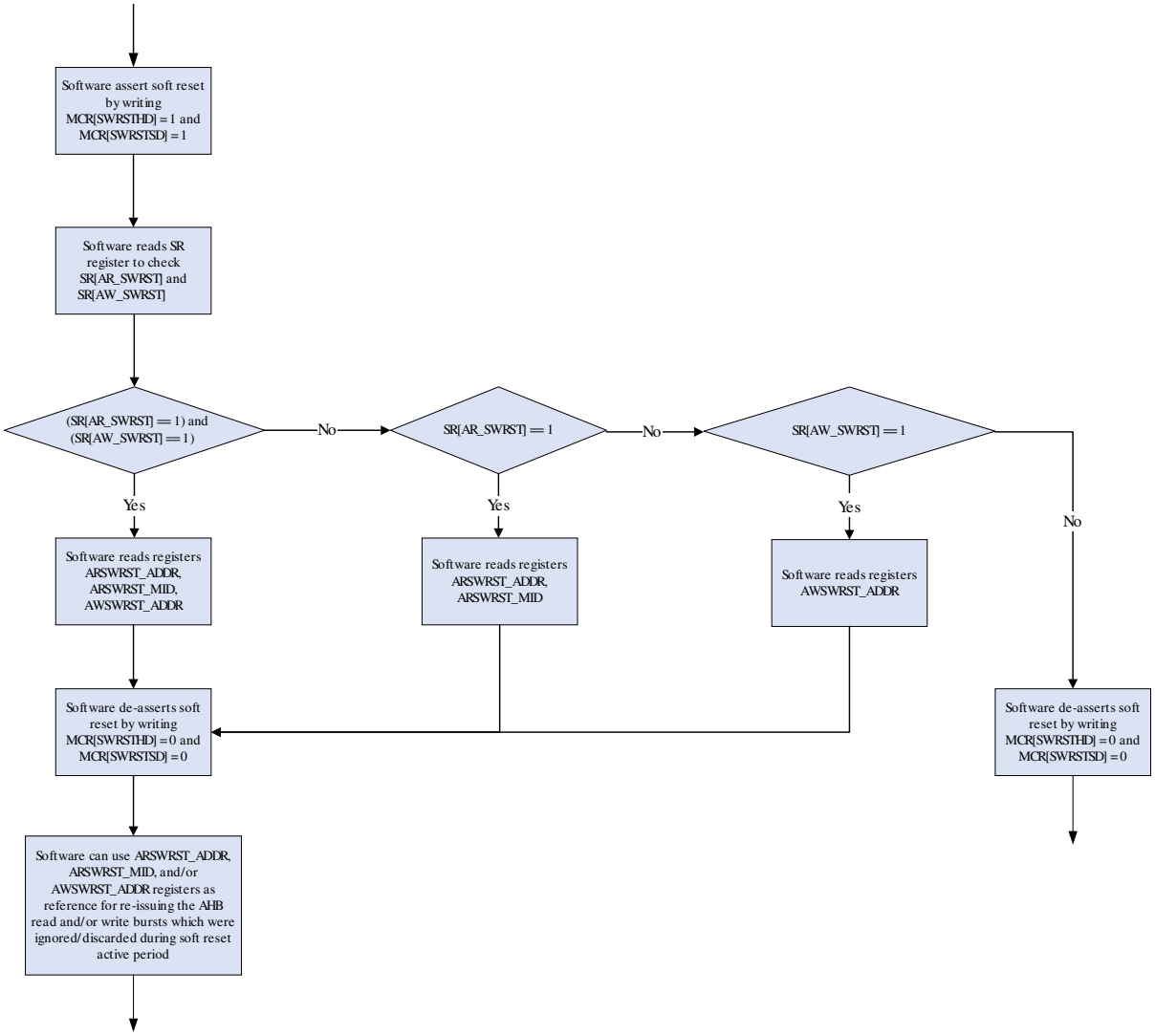


Figure 159. Soft reset programming sequence

41.3.19 Clocking

XSPI has the following clocks.

Table 355. Clocking

XSPI Clock Domain	Input clocks to XSPI-IP	Functional descriptions
Host System clock domain	SoC Internal: AHB clock (ahb_clk)	Input clock for AHB clock domain logic within XSPI-IP. It is used for handling the AHB read and AHB write accesses/operations through AHB interface to the external memory.

Table continues on the next page...

Table 355. Clocking (continued)

XSPI Clock Domain	Input clocks to XSPI-IP	Functional descriptions
		NOTE 1. AHB clock can be gated by ahb_enable_clk signal. 2. Supported AHB : IPS clock ratio == 2:1 or 1:1 only (Edge aligned Synchronous clocks) 3. 2: 1 clock ratio with IPED is restricted with AHB MIN Freq=133Mhz
	SoC Internal: IPS clock (ipg_clk_s)	**In-Direct memory-mapped accesses:** Peripheral clock used for controlling the In-Direct memory-mapped accesses through IPS interface. To allow external logic to stop this clock by writing 1 to MCR[MDIS]. **NOTE** This clock must be running for software reset (MCR[IPS_TG_RST]) to propagate it to the TG queue.
	SoC Internal: IPS clock (ipg_clk)	**Register-mapped accesses:** Peripheral clock used for controlling the register accesses on Control and Configuration registers within XSPI-IP.
Serial Memory (Volatile/ Non-Volatile) domain	SoC External: Free running Serial Memory domain clock (ipg_clk_sfck)	Free running Serial Memory (Volatile/Non-Volatile) domain clock: 1. Source clock for External Memory Device Clock through PAD: SCK 2. Also a source clock for internally generated Dummy/DQS PAD loopback DQS signal for data sampling.
	SoC internal: Gated Serial Memory domain clock (ipg_clk_sfif)	Serial Memory (Volatile/Non-Volatile) domain clock for XSPI-IP logic. It is synchronous to input clock: ipg_clk_sfck. **NOTE** Clock can be gated by "enable_clk_sfif" signal
	SoC internal: Gated 2x Serial Memory domain clock (ipg_clk_2xsfif)	2x Serial Memory (Volatile/Non-Volatile) domain clock for XSPI-IP logic. It is synchronous to input clock: ipg_clk_sfck. **NOTE** Clock can be gated by enable_clk_2xsfif signal in non-DDR mode.
Serial Memory Interface domain	SoC External: DQS (External DQS or Dummy/DQS PAD loopback DQS)	Serial memory device DQS signal. Asynchronous to the other clocks from both Host System bus & Serial Memory (Volatile/Non-Volatile) domains.

41.3.20 Reset

XSPI has the following reset sources.

Table 356. Resets

Reset source	Description
Hard reset	Resets whole XSPI module from external reset signal.
Software reset for serial flash memory domain	MCR[SWRSTSD]
Software reset for AHB domain	MCR[SWRSTHD]
Software reset for IPS target group queue	MCR[IPS_TG_RST]

41.3.21 DMA

This section provides XSPI-specific DMA information. It describes DMA operation in normal mode. For a complete description of the DMA module, see the DMA controller chapter.

41.3.21.1 AHB bus-side read: case 1

The total number of bus cycles for each DMA minor loop completion is added from the following components:

In this use case, DMA needs to read 4 bytes from memory and provide them to XSPI, which takes 4 bus clock cycles. Then, DMA handshake adds an additional 6 cycles, resulting in a total of [6 + 4 * (32/4) = 38] bus clock cycles.

This table provides typical read examples. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

Table 357. Access duration examples for AHB bus-side read: case 1

RBCT[WMRK]	Bytes per DMA minor loop[1]	Bus clock cycles per DMA minor loop	Duration of DMA minor loop at 60 MHz bus clock frequency
3	16	6 + (16/4) * 4 = 22	~366ns
7	32	6 + (32/4) * 4 = 38	~633ns
11	48	6 + (48/4) * 4 = 54	~900ns
15	64	6 + (64/4) * 4 = 70	~1166ns

1. A DMA minor loop is the process of filling the RX buffer to the watermark and then reading the buffer.

41.3.21.2 AHB bus-side read: case 2

The total number of bus cycles for each DMA minor loop completion is added from the following components:

Case 2: DMA needs to read 32 bytes from SRAM and provide to XSPI. DMA handshake takes an additional six bus cycles, with 32 bytes DMA read from SRAM costs (8 + 3) core clock cycles. DMA writes 32 bytes to XSPI, takes 2 * (32/4) = 16 bus cycles with one additional CPU access to XSPI, costing two bus clock cycles. This results in a total 6 + (8+3)/2 + 2 * (32/4) +2 = 30 bus clock cycles.

This table provides typical read examples. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

Table 358. Access duration examples for bus clock side

RBCT[WMRK]	Bytes per DMA minor loop[1]	Bus clock cycles per DMA minor loop	Duration of DMA minor loop at 80 MHz bus clock frequency
3	16	6 + (4+3) /2 + (16/4)*2 + 2 = 20	~333ns
7	32	6 + (8+3) /2 + (32/4)*2 + 2 = 30	~500ns
11	48	6 + (4+3)/2*3 + (48/4)*2 + 2 = 44	~733ns
15	64	6 + (8+3) + (64/4)*2 + 2 = 51	~810ns

1. A DMA minor loop is the process of filling the RX buffer to the watermark and then reading the buffer.

41.3.21.3 Memory device-side read

For the memory device-side read data flow, the duration in bus clock cycles for a DMA minor loop completion is the sum of the following components:

- Clock cycles to read 4 bytes, corresponding to one RX buffer entry (not counting setup of command and address):
  - Octal DDR mode in the individual flash memory mode: 2
  - Quad SDR mode instructions in individual flash memory mode: 8

  and so on.
- Overhead due to clock domain crossing: 1 cycle

This table lists the clock cycles required to read data from memory depending on typical RX buffer watermark (RBCT[WMRK]) values. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

Table 359. Access duration examples for serial flash memory side

RBCT[WMRK]	Bytes per DMA minor loop[1]	SCKFx clock cycles at 60 MHz		Duration of flash memory data readout at 80 MHz SCKFx (~12.5 ns period)	
		IFM [2] quad	IFM quad DDR	IFM quad	IFM quad DDR
0	4	8	4	~100 ns	~50 ns
1	8	16	8	~200 ns	~100 ns
3	16	32	16	~400 ns	~200 ns
7	32	64	32	~800 ns	~400 ns

1. A DMA minor loop is the process of filling the RX buffer to the watermark and then reading the buffer.
2. Individual flash memory mode

A complementary example is when the watermark is set to be too high. In such a case, the time taken by the DMA to read out the RX buffer entries should be lesser than the time taken by the controller to push in the remaining entries in the buffer.

41.3.21.4 IPS bus-side write

For the IP bus-side write data flow, the duration in bus clock cycles for a DMA minor loop completion is the sum of the following components:

- Overhead for each minor loop, as determined by DMA controller: assume 10 cycles

- Overhead resulting from clock domain crossing: assume 2 cycles
- Number of cycles required for 16 bytes (128-bit write size): assume 4 cycles (read/write sequence of DMA controller)

The TX buffer watermark (TBCT[WMRK]) determines the size of the DMA minor loop. Therefore, XSPI distributes the DMA controller overhead across (TBCT[WMRK] + 1) write accesses of 32 bits each.

This table provides typical write examples. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

Table 360. Access duration examples for IP bus-side write

TBCT[WMRK]	Bytes per DMA minor loop[1]	Bus clock cycles per DMA minor loop	Duration of DMA minor loop at 80 MHz bus clock frequency
3	16	12 + 4 = 16	~200 ns
7	32	12 + 8 = 20	~250 ns

1. A DMA minor loop is the process of filling the RX buffer to the watermark and then reading the buffer.

41.3.21.5 Memory device-side write

For the memory device-side write data flow, the duration in bus clock cycles for a DMA minor loop completion is the sum of the following components:

- Memory clock cycles required to write 16 bytes, corresponding to four TX buffer entries (not counting setup of command and address):
  - Octal DDR mode instructions in individual flash memory mode: 8
  - Quad SDR writes in individual flash memory mode: 32
- Overhead due to clock domain crossing: 1

The TX buffer watermark (TBCT[WMRK]) determines the size of the DMA minor loop. This table provides typical write examples. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

This table provides typical write examples. The data represents an ideal scenario; actual performance depends on the chip-specific integration of XSPI and the DMA controller.

Table 361. Access duration examples for serial flash memory side

TBCT[WMRK]	Bytes per DMA minor loop[1]	SCKFx clock cycles per DMA minor loop		Duration for consuming data on memory interface at 100 MHz SCKFx (10 ns period)[2]		Duration for emptying FIFO buffer[3]	
		IFM [4] quad	IFM octal DDR	IFM quad	IFM octal DDR	IFM quad	IFM octal DDR
3	16	32	8	320 ns	80 ns	2240 ns	560 ns
7	32	64	16	640 ns	160 ns	1920 ns	480 ns

1. A DMA minor loop is the process of filling the RX buffer to the watermark and then reading the buffer.
2. Not all flash memory devices support writes at 100 MHz. For the actual page program frequency supported, see the flash memory data sheet.
3. These timings assume that the TX buffer is full when the transaction is initiated.
4. Individual flash memory mode

Considering the examples provided in the two tables above for TX FIFO, it is evident that depending on the relationship between the bus clock and serial flash memory clock frequencies, there are settings possible where the serial flash memory consumes

data faster than the IPS bus can write data in TX buffer. In these cases, a TX buffer underrun situation occurs. To avoid TX buffer underrun, the data transaction size should be large enough.

41.4 External signals

Table 362. External signals

Signal name	Function	Direction	Description
PCSFA1	Peripheral chip select flash memory A1	O	Chip select for the serial flash memory device A1, the first device in a dual-die package flash memory A or the first of two flash memory devices that share IOFA. See Dual-die flash memories.
PCSFA2	Peripheral chip select flash memory A2	O	Chip select for the serial flash memory device A2, the second device in a dual-die package flash memory A or the second of two flash memory devices that share IOFA. See Dual-die flash memories.
SCKFA	Serial clock flash memory A	O	Serial clock output to serial flash memory device A.
SLCKFA_N	Serial loopback clock flash memory A	O	Inverted serial clock output to serial flash memory device A.
IOFA[7:0]	Serial I/O flash memory A	I/O	Data I/O lines to and from serial flash memory device A. For information about signal drive and timing behavior, see Driving external signals. The function of signal pins of the serial flash memory device may change according to the SFM command executed, leaving them as control inputs when executing single and dual instructions. XSPI supports driving these inputs to dedicated values. In single I/O mode, XSPI drives data on IOFA[0] and expects data on IOFA[1].
IOFA_x16[7:0]	x16 Serial I/O Flash memory A	I/O	x16 data I/O lines to and from memory device A. For signal drive and timing behavior, see Driving external signals. The signal pins of the memory device may change function depending on the SFM command being executed, acting as control inputs when executing single and dual instructions. XSPI supports driving these inputs to dedicated values.
DQSFA	Data strobe signal flash memory A	I/O	Data strobe signal for port A. Some flash memory vendors provide the Data Read Strobe (DQS) signal to which the read data is aligned in DDR mode. It also functions as an output signal during the write data phase.
DQS1FA	x16 data strobe signal flash memory A	I/O	x16 data strobe signal for port A. Some flash memory vendors provide the x16 Data Read Strobe (DQS1) signal to which the read data is aligned in DDR mode. It also functions as an output signal during write data phase.

NOTE

See the chip-specific information to confirm the XSPI signals that are available.

41.4.1 Driving external signals

Single/dual/quad/octal/x16 instructions

Depending on the serial flash memory device connected to XSPI, there are instructions that use different numbers of data lines:

- Single pad: Single line I/O with one data out and one data in line to/from the serial flash memory device
- Dual pad: Dual line I/O with two bidirectional I/O lines, driven alternatively by the serial flash memory device or the XSPI module

- Quad pad: Quad line I/O with four bidirectional I/O lines, driven alternatively by the serial flash memory device or the XSPI module
- Octal pad: Octal line I/O with eight bidirectional I/O lines, driven alternatively by the serial flash memory device or the XSPI module

The different phases of the serial flash memory access scheme are shown in the following figure.

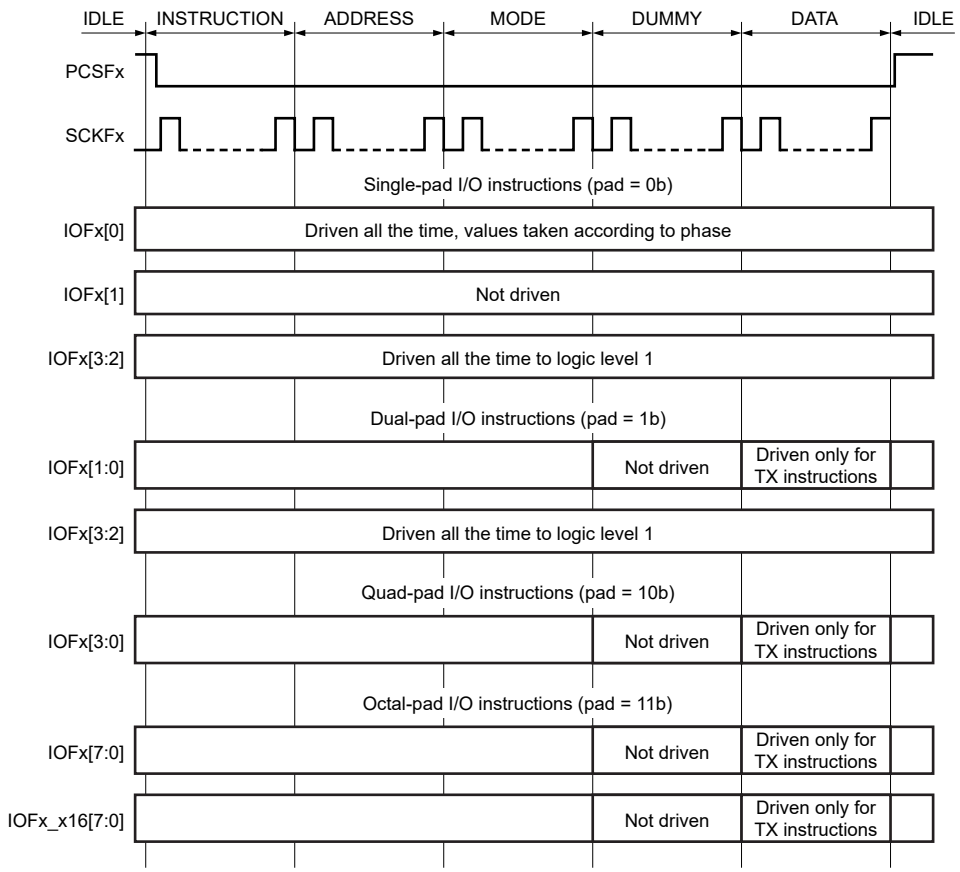


Figure 160. Serial flash memory access scheme

Note1:The IOs are driven from XSPI as per the number of pads configured for ongoing phase.

Note:The lines status will change based on command mode in case of instruction, address and mode phases. It can be either 1,2 or 4 lines

Following are the different phases and the I/O driving characteristics of the XSPI module:

- Idle: Serial flash memory device not selected – no interaction with the serial flash memory device and the IOFx signals are driven.
- Instruction: Serial flash memory device selected – the instruction is sent to the serial device and all the IOFx signals are driven.
- Address: Serial flash memory address is sent to the device – all the IOFx signals are driven and this phase is not applicable for all SFM commands.
- Mode: Mode bytes are sent to the serial flash memory device – all the IOFx signals are driven and this phase is not applicable for all SFM commands.
- Dummy: Dummy clocks are provided to the serial flash memory device. See Figure 160 for the IOFx signals driven. The actual data lines required for the SFM command executed are not driven for data read commands.

NOTE

— This phase is not applicable for all the SFM commands.

— All read commands in Dual pad, Quad pad /or Octal pad modes must use dummy phase before read phase. Note that this restriction is not applicable to Single-pad mode.

- Data: Serial flash memory data are sent to or received from the serial flash memory device. See the preceding figure for the IOFx signals driven. The actual data lines required for the SFM command executed are not driven for data read commands.

NOTE

This phase is not applicable for all the SFM commands.

The PCSFx and SCKFx signals are driven permanently throughout all the phases. In the individual flash memory mode, this applies to the selected flash memory device.

Access to a single, individual serial flash memory device

See Serial flash memory access schemes for details.

Read access to two serial flash memory devices attached to the XSPI module in parallel. See Serial flash memory access schemes for details.

41.5 Initialization

41.5.1 Power up and reset

This is an example sequence for resetting flash memory device A:

You must ensure:

- The inputs of a memory device connected to XSPI meet the required special voltage characteristics for that device during power up or reset.
- The memory device supports reset when CS*n* is high and IOFA[3] is low.

CAUTION

To avoid corrupted flash pages, all erase or program commands must complete before you issue a reset or power cycle. To enable recovery from corrupted flash pages, your application must ensure there is a backup of critical data stored at a different location.

Step	Purpose	Programming
1	Initiate software reset and hardware reset.	• Write 1 to MCR[SWRSTSD]. • Write 1 to MCR[SWRSTHD].
2	End software reset and hardware reset.	• Write 0 to MCR[SWRSTSD]. • Write 0 to MCR[SWRSTHD].
3	Reset the SFP TG queues.	Write 1 to MCR[IPS_TG_RST] .
4	Disable XSPI.	Write 1 to MCR[MDIS].
5	Deassert the idle signal for flash memory A (IOFA[3]).	Write 0 to MCR[ISD3FA].
6	Enable XSPI.	Write 0 to MCR[MDIS].
7	Disable XSPI.	Write 1 to MCR[MDIS].

Table continues on the next page...

Table continued from the previous page...

Step	Purpose	Programming
8	Assert the idle signal for flash memory A (IOFA[3]).	Write 1 to MCR[ISD3FA].
9	Enable XSPI.	Write 0 to MCR[MDIS].

41.5.2 Available status/flag information

This section provides an overview of the different flags and statuses available, and their interdependencies for different use cases. The SR and FR are the related registers. SR[ARB_STATE] can be checked to know which state the arbitration state machine is presently in.

41.5.2.1 IP commands

See IP Configuration (IPCR) for additional details not explicitly covered in this paragraph.

- IP commands — normal operation

  Once the IP command is triggered, after arbitration is granted the SFAR and IPCR registers are written to XSPI. Given that this is a legal command, SR[IPACC] and SR[BUSY] are asserted simultaneously, immediately after the execution starts.

  XSPI IPCR[SEQID] is written only after SFP related checks have been cleared by that TG queue and arbitration is won. See Secure flash protection for details.

  After the instruction on the serial flash memory device is complete, these fields deassert and FR[TFF] is configured.

- IP commands — error situations

  See FR error flags overview .

41.5.2.2 AHB commands

See the "Reading serial flash memory data into the XSPI module internal buffers" topic in the Flash Memory Read section for details.

- AHB commands - normal operation

  Memory-mapped read access to a serial flash memory address not contained in the AHB buffer triggers the execution of an AHB command. Given that this is a legal command, SR[AHB_ACC] and SR[BUSY] are asserted simultaneously, immediately after the execution starts. After the instruction on the serial flash memory device is complete, these fields are deasserted.

- AHB commands - error situations

  See FR error flags overview.

41.5.2.3 SFM commands

An SFM command consists of an instruction code and all other parameters (for example, size or mode bytes) needed for that specific instruction code. Triggering a command either initiates a transaction on the external serial flash memory or results in an error. See FR error flags overview.

41.5.2.4 FR error flags overview

This table provides an overview of the flags in FR, as well as additional error-related details.

Table 363. FR error flags overview

Error	Flag	Description	Command execution on serial flash memory device FR[TFF] behavior (in case of IP commands only and AHB behavior)
AHB Buffer Overflow	ABOF	Indicates an attempt to push too much data into the AHB buffer. Occurs only as a result of an incorrect value in BUF*n*CR[ADATSZ].	Flash memory transaction continues until completion. No AHB error response.
AHB Illegal Burst Size	AIBSEF	Indicates that the total burst size (size x beat) of an AHB transaction is greater than the specified prefetch data size (BUF*n*CR[ADATSZ]).	XSPI terminates the flash memory transaction and rejects incoming AHB read transactions with an AHB error response (HRESP). If the HRESP error mask is asserted (SOCCFG[HRESP_ERR_MASK] = 1) then XSPI does not return an AHB error but instead provides the AHB error payload on the AHB read data bus and processes the next transaction regardless of this flag.
AHB Illegal Transaction Error	AITEF	Indicates that XSPI has not generated a response to the AHB bus due to an illegal transaction and expiration of the watchdog timer.	XSPI terminates the memory transaction and provides the error payload to the AHB read data bus for the beat where the timeout error occurred. It executes subsequent AHB read transactions. While this flag is set, XSPI generates the AHB error response (HRESP) for all AHB write transfers. If the AHB Illegal Transaction Error interrupt is enabled (RSER[AITIE] = 1) and the interrupt is not cleared, XSPI blocks the SFP/IP interface.
Data Learning Pattern Failure	DLPFF	Indicates that XSPI encountered a DATA_LEARN instruction in a sequence, but no sampling point is found for the data learning pattern.	The memory transaction continues until completion. While this flag is set, XSPI generates the AHB error response (HRESP). If the HRESP error mask is asserted (SOCCFG[HRESP_ERR_MASK] = 1) then XSPI does not return an AHB error but instead provides the AHB error payload on the AHB read data bus and processes the next transaction regardless of this flag.
Illegal Instruction Error	ILLINE	Indicates that XSPI encountered an illegal instruction in any sequence.	XSPI terminates the memory transaction. When this error occurs, you must reconfigure the correct sequence instruction and assert software reset for both the serial flash memory and AHB domains (see FR[ILLINE]).
IP Command Trigger Fail Error	IPIEF	Indicates when one of several conditions occur. For details, see the field description.	XSPI ignores any command that causes assertion of this error. XSPI does not assert IP Command Transaction Finished (TFF) for the triggering command.
RX Buffer Overflow[1]	RBOF	Indicates when the RX buffer cannot receive any more data from the serial flash memory device.	XSPI asserts IP Command Transaction Finished (TFF) for the triggering command. If the RX Buffer Drain Interrupt is enabled (RSER[RBDIE] = 1), then you must clear the interrupt for completion of the IP command; otherwise, you may encounter an SFP timeout error. XSPI executes the next SFP TG transaction only after you clear this interrupt.

Table continues on the next page...

Table 363. FR error flags overview (continued)

Error	Flag	Description	Command execution on serial flash memory device FR[TFF] behavior (in case of IP commands only and AHB behavior)
TX Buffer Underrun[1]	TBUF	Indicates that XSPI has attempted to pull data when the TX buffer is empty.	XSPI asserts IP Command Transaction Finished (TFF) for the triggering command. If the TX Buffer Underrun Interrupt is enabled (RSER[TBUIE] = 1), then you must clear the interrupt for completion of the IP command; otherwise, you may encounter an SFP timeout error. XSPI executes the next SFP TG transaction only after you clear this interrupt. The AHB write transfer remains waiting until this flag is cleared.

1. Only the buffer-related errors are associated with a transaction on the external memory.

41.5.3 Flash memory device selection

Regardless of the SFM command (IP or AHB), the access mode (access to serial flash memory A1 or A2) is selected by specifying the 32-bit address value for the following SFM command.

For IP commands, the access mode is selected with the address programmed into the SFAR register. See Serial Flash Memory Address (SFAR) for details. SFAR register is read-only if SFP block is present. When SFP is present the address is programmed in Serial Flash Memory Address (SFP_TG_SFAR) registers. See Secure flash protection for details.

For AHB commands, the access mode is determined by the memory-mapped address. See Flash memory mapped AMBA bus for details.

41.5.4 Flash memory devices address mapping

XSPI is configured in Single mode for the supported flash memory port A

The sizes of the flash memory devices are mapped with the system memory space based on the configurations of the following registers:

- SFA1AD
- SFA2AD

The total memory region for the flash memory devices is mapped between XSPI_AMBA_BASE and TOP_ADDR_MEMA2 such that the corresponding CS is asserted based on SFA1AD and SFA2AD register configurations.

41.5.4.1 Single mode

For single-die flash memories, you must write the same value to SFA2AD register that you write to the SFA1AD register.

Following is a programming example for single mode single-die flash memory:

- XSPI_AMBA_BASE - 1000_0000h
- SFA1AD[TPADA1] - 2000_0000h
- SFA2AD[TPADA2] - 2000_0000h

The following figure illustrates the memory mapping for single mode XSPI configuration.

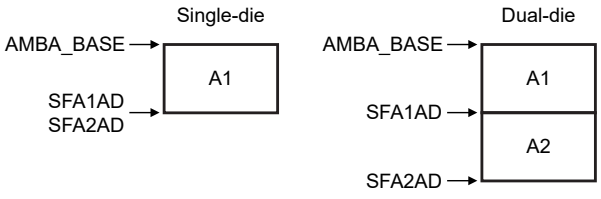


Figure 161. Memory map for Single mode

41.6 Application information

41.6.1 Configuration for x8 and x16 operation

x8 and x16 operation requires the following configuration:

- In x16 operation, the PAD*n* value should be 11b (eight pads).
- To enable x16 mode, write 1 to MCR[X16_EN].
- Based on the requirements of the target device:
  - To enable 16-bit WORD addressable mode, write 1 to SFACR[WA].
  - To enable 32-bit WORD addressable mode, write 1 to SFACR[WAWA_4B_EN].
- To send address and command information on a x16 IO databus, write 1 to MCR[X16_DB_CA_EN].
- To enable variable latency, write 1 to MCR[VAR_LAT_EN].
- PSRAM (such as APMEM) does not support variable latency.
- XSPI does not support wrap burst transactions.
- For dummy cycle operation configuration, see Dummy cycles generation.

41.6.2 Example LUT sequences

The following sections provide example sequences for use with various memory brands and types.

- Sequence: Exit 4 x IO read enhance performance mode (XIP) and read status—Macronix
- Sequence: Read command—HYPERFLASH/HyperRAM
- Sequence: Read status register—HYPERFLASH/HyperRAM
- Sequence: Word program—HYPERFLASH/HyperRAM
- Sequence: Fast read—Macronix, Numonyx, Spansion, Winbond
- Sequence: Fast dual IO DT read—Macronix
- Sequence: Fast read quad output—Winbond
- Sequence: 4 x IO read in enhanced performance mode (XIP)—Macronix
- Sequence: Dual command page program—Numonyx
- Sequence: Sector erase—Macronix, Numonyx, Spansion
- Sequence: Read status register—Macronix, Numonyx, Spansion, Winbond
- Sequence: Data learn instruction for 4 x IO memory
- Sequence: Data learn instruction for 8 x IO memory
- Sequence: x8 and x16 WRITE with DUMMY split and 16-bit word addressing

- Sequence: x8 or x16 READ with DUMMY split and 16-bit word addressing
- Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 1
- Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 1
- Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 0
- Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 0
- Sequence: x8 and x16 WRITE without DUMMY split, variable latency
- Sequence: x8 and x16 READ without DUMMY split, variable latency
- Sequence: x8 and x16 WRITE with DUMMY split, fixed latency
- Sequence: x8 and x16 READ with DUMMY split, fixed latency
- Sequence: x8 and x16 WRITE without DUMMY split, fixed latency
- Sequence: x8 and x16 READ without DUMMY split, fixed latency

For all x8 and x16 sequences, see also Configuration for x8 and x16 operation.

41.6.2.1 Sequence: Exit 4 x IO read enhance performance mode (XIP) and read status—Macronix

Table 364. Sequence: Exit 4 x IO read enhance performance mode (XIP) and read status

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	EBh	4 x I/O read command.
LUT0 1	ADDR	10b	18h	Send 24 address bits on four pads.
LUT1 0	MODE	10b	00h	Execute two mode cycles (exit XIP).
LUT1 1	DUMMY	00b	04h	Execute four dummy cycles.
LUT2 0	READ	10b	08h	Read 8 bytes (64 bits) on four pads.
LUT2 1	CMD	00b	05h	Read status register command.
LUT3 0	READ	00b	01h	Read status register data.
LUT3 1	STOP	00b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.2 Sequence: Read command—HYPERFLASH/HyperRAM

HYPERFLASH/HyperRAM memory is word-addressable memory. Each address accesses a word-wide (two-byte) data value. When connecting HyperFLASH/HyperRAM memory to the controller, you must configure that memory as word-addressable (write 1 to SFACR[WA]), which causes XSPI to remap a byte access to a word access.

Table 365. Sequence: Read command—HYPERFLASH/HyperRAM

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Read command with continuous burst type.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address with lower 3 bits valid and the rest 0.

Table continues on the next page...

Table 365. Sequence: Read command—HYPERFLASH/HyperRAM (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT1 1	DUMMY	11b	0Fh	Execute 15 dummy cycles.
LUT2 0	READ_DDR	11b	04h	Read 32 bits of data on 8 pads.
LUT2 1	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.3 Sequence: Read status register—HYPERFLASH/HyperRAM

Table 366. Sequence: Read status register—HYPERFLASH/HyperRAM

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
Sequence: Read Pre Command				
LUT0 0	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address (0000AAh) in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	16-bit column address with lower 3 bits valid and the rest 0 (0005h) treated as command.
LUT1 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT2 0	CMD_DDR	11b	70h	Write data to memory as pre-command.
Sequence: Command phase (fourth/final chip select phase)				
LUT2 1	CMD_DDR	11b	A0h	Read command with continuous burst type.
LUT3 0	ADDR_DDR	11b	18h	Send 24-bit row address.
LUT3 1	CADDR_DDR	11b	10h	Send 16-bit column address with lower 3 bits as valid and the rest 0.
LUT4 0	DUMMY	11b	0Fh	Execute 15 dummy cycles
LUT4 1	READ_DDR	11b	04h	Read 4 bytes (32 bits) on 8 pads.
LUT5 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.4 Sequence: Word program—HYPERFLASH/HyperRAM

Table 367. Sequence: Word program—HYPERFLASH/HyperRAM

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
Unlock sequence 1 (first chip select phase)				
LUT0 0	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT0 1	CMD_DDR	11b	00h	Treat 8-bit address 00h as command.

Table continues on the next page...

Table 367. Sequence: Word program—HYPERFLASH/HyperRAM (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT1 0	CMD_DDR	11b	00h	Treat 8-bit address 00h as command.
LUT1 1	CMD_DDR	11b	AAh	Treat 8-bit address AAh as command.
LUT2 0	CADDR_DDR	11b	10h	Treat 16-bit column address with lower 3 bits valid and the rest 0 (0005h) as command
LUT2 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT3 0	CMD_DDR	11b	AAh	Write data to memory as pre-command.
Unlock sequence 2 (second chip select phase)				
LUT3 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT4 0	CMD_DDR	11b	00h	Treat 8-bit address 00h as command
LUT4 1	JMP_TO_SEQ	11b	01h	Jump to sequence 1 (which begins with LUT5).
LUT5 0	CMD_DDR	11b	00h	Treat 8-bit address 00h as command.
LUT5 1	CMD_DDR	11b	55h	Treat 8-bit address 55h as command.
LUT6 0	CADDR_DDR	11b	10h	Treat 16-bit column address with lower 3-bits valid and the rest 0 (0002h) as command.
LUT6 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT7 0	CMD_DDR	11b	55h	Write data to memory as pre-command
Program setup phase (third chip select phase)				
LUT7 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT8 0	CMD_DDR	11b	00h	Treat 8-bit address 00h as command.
LUT8 1	CMD_DDR	11b	00h	Treat 8-bit address 00h as command.
LUT9 0	CMD_DDR	11b	AAh	Treat 8-bit address AAh as command.
LUT9 1	JMP_TO_SEQ	11b	02h	Jump to sequence 2 (which begins with LUT10).
LUT10 0	CADDR_DDR	11b	10h	Treat 16-bit column address with lower 3 bits valid and the rest 0 (0005h) as command.
LUT10 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT11 0	CMD_DDR	11b	A0h	Write data to memory as pre-command.
Command phase (fourth/final chip select phase)				
LUT11 1	CMD_DDR	11b	00h	Write command with wrapped burst type.
LUT12 0	ADDR_DDR	11b	18h	Send 24-bit row address in DDR mode.
LUT12 1	CADDR_DDR	11b	10h	Send 16-bit column address with lower 3 bits as valid and the rest 0.
LUT13 0	WRITE_DDR	11b	02h	Write 2 bytes (16 bits) on 8 pads (D1D2).
LUT13 1	STOP	11b	00h	Stop, sequence completed.

1. For demonstration purposes, sequence numbering assumes a sequence depth of five instruction-operand pairs and the use of LUT sequences 0-2, with JMP_TO_SEQ commands to bridge the sequences. You can program this example into any available LUT sequences.

2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.5 Sequence: Fast read—Macronix, Numonyx, Spansion, Winbond

Table 368. Sequence: Fast read

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	0Bh	Perform a fast read.
LUT0 1	ADDR	00b	18h	Send 24 address bits on one pad.
LUT1 0	DUMMY	00b	08h	Execute eight dummy cycles.
LUT1 1	READ	00b	04h	Read 32 bits on one pad.
LUT2 0	JMP_ON_CS[3]	00b	00h	Jump to instruction 0 (LUT0 0: CMD).

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. If DLL is enabled (DLLCR*x*[DLLEN] = 1), you must use a STOP instruction.

41.6.2.6 Sequence: Fast dual IO DT read—Macronix

Table 369. Sequence: Fast dual IO DT read

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT	CMD	00b	BDh	Fast dual IO DT read command.
LUT0 0	ADDR_DDR	01b	18h	Send 24 address bits on two pads in DDR mode.
LUT0 1	MODE4_DDR	01b	00h	P2 = P0 or P3 = P1 is necessary. See the Macronix data sheet for details. One clock cycle for mode.
LUT1 0	DUMMY	00b	06h	Execute six dummy cycles.
LUT1 1	READ_DDR	01b	04h	Read 32 bits on two pads in DDR mode.
LUT2 0	JMP_ON_CS[3]	00b	00h	Jump to instruction 0 (LUT0 0: CMD).

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. If DLL is enabled (DLLCR*x*[DLLEN] = 1), you must use a STOP instruction.

NOTE

If DLL is enabled (DLLCRx[DLLEN] = 1), then we should use only STOP and if DLL is disabled, then we can use either STOP or JMP_ON_CS as per requirement.

41.6.2.7 Sequence: Fast read quad output—Winbond

Table 370. Sequence: Fast read quad output

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	6Bh	Fast read quad output command.
LUT0 1	ADDR	00b	18h	Send 24 address bits on one pad.

Table continues on the next page...

Table 370. Sequence: Fast read quad output (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT1 0	DUMMY	00b	08h	Execute eight dummy cycles.
LUT1 1	READ	10b	04h	Read 32 bits on four pads.
LUT2 0	JMP_ON_CS[3]	00b	00h	Jump to instruction 0 (LUT0 0: CMD).

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. If DLL is enabled (DLLCR*x*[DLLEN] = 1), you must use a STOP instruction.

41.6.2.8 Sequence: 4 x IO read in enhanced performance mode (XIP)—Macronix

In XIP mode, you must ensure that all flash memories connected to XSPI are in this mode. As part of initialization, all the flash memories might be enabled with XIP by carrying out dummy reads.

NOTE

When using XIP mode, you must request arbitration lock for the queue executing the XIP transfer. Otherwise, after the first transfer completes another high priority manager may acquire the grant.

Table 371. Sequence: 4 x IO read in enhanced performance mode (XIP)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	EBh	4 x IO read command.
LUT0 1	ADDR	10b	18h	Send 24 address bits on four pads.
LUT1 0	MODE	10b	A5h	Two mode cycles.
LUT1 1	DUMMY	00b	04h	Execute four dummy cycles.
LUT2 0	READ	10b	04h	Read 32 bits on four pads,
LUT2 1	JMP_ON_CS	00b	01h	Jump to instruction 1 (LUT0 1: ADDR).

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.9 Sequence: Dual command page program—Numonyx

Table 372. Sequence: Dual command page program

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	01b	02h	Dual command page program.
LUT0 1	ADDR	01b	18h	Send 24 address bits on two pads.
LUT1 0	WRITE	01b	20h	Write 32 bytes on two pads.
LUT1 1	STOP	00b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.10 Sequence: Sector erase—Macronix, Numonyx, Spansion

Table 373. Sequence: Sector erase

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	20h	Sector erase command.
LUT0 1	ADDR	00b	18h	Send 24 address bits on one pad.
LUT1 0	STOP	00b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.11 Sequence: Read status register—Macronix, Numonyx, Spansion, Winbond

Table 374. Sequence: Read status register

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	05h	Read status register command.
LUT0 1	READ	00b	01h	Read status register data.
LUT1 0	STOP	00b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.

41.6.2.12 Sequence: Data learn instruction for 4 x IO memory

Table 375. Sequence: Data learn instruction for 4 x IO memory

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD	00b	6Bh	Fast read quad command.
LUT0 1	ADDR_DDR	00b	18h	Send 24 address bits on four pads in DDR mode.
LUT1 0	DUMMY	10b	08h	Execute eight dummy cycles.
LUT1 1	DATA_LEARN	10b	01h	Send 1-byte data learn pattern.
LUT2 0	READ_DDR	10b	06h	Read 6 bytes on 4 pads.[3]
LUT2 1	JMP_ON_CS[4]	00b	00h	Jump to instruction 0 (LUT0 0: CMD).

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For 4 x IO mode, you must follow the DATA_LEARN instruction with a READ_DDR instruction of a minimum of eight bytes.
4. If DLL is enabled (DLLCR*x*[DLLEN] = 1), you must use a STOP instruction.

41.6.2.13 Sequence: Data learn instruction for 8 x IO memory

Table 376. Sequence: Data learn instruction for 8 x IO memory

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Read command for hyperflash memory.

Table continues on the next page...

Table 376. Sequence: Data learn instruction for 8 x IO memory (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode. [3]
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	0Fh	Execute 15 dummy cycles.
LUT2 0	DATA_LEARN	11b	01h	Send 1-byte data learn pattern.
LUT2 1	READ_DDR	11b	0Ch	Read 12 bytes on 8 pads in DDR mode.
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. The address must be aligned to prevent latency between the RWDS edges.

41.6.2.14 Sequence: x8 and x16 WRITE with DUMMY split and 16-bit word addressing

See Configuration for x8 and x16 operation.

Table 377. Sequence: x8 and x16 WRITE with DUMMY split and 16-bit word addressing

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Write command at positive edge of the clock.
LUT0 1	CMD_DDR	11b	A0h	Write command at negative edge of the clock.
LUT1 0	ADDR_DDR	11b	20h	Send 32-bit address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	84h	First DUMMY instruction: Execute four dummy cycles.[3]
LUT2 0	DUMMY	11b	82h	Second DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	WRITE_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.15 Sequence: x8 or x16 READ with DUMMY split and 16-bit word addressing

See Configuration for x8 and x16 operation.

Table 378. Sequence: x8 and x16 READ with DUMMY split and 16-bit word addressing

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Read command at positive edge of the clock.
LUT0 1	CMD_DDR	11b	20h	Read command at negative edge of the clock.
LUT1 0	ADDR_DDR	11b	20h	Send 32-bit address on eight pads in DDR mode.

Table continues on the next page...

Table 378. Sequence: x8 and x16 READ with DUMMY split and 16-bit word addressing (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT1 1	DUMMY	11b	8Ch	First DUMMY instruction: Execute twelve dummy cycles.[3]
LUT2 0	DUMMY	11b	82h	Second DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	READ_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.16 Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 1

See Configuration for x8 and x16 operation.

Table 379. Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 1

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Write command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	MODE	11b	00h	Single cycle mode.
LUT2 0	DUMMY	11b	84h	First split DUMMY instruction: For a total of 7 dummy cycles, specify 4 in this first instruction.[3]
LUT2 1	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT3 0	WRITE_DDR	11b	FFh	Write the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 1	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUMMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.17 Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 1

See Configuration for x8 and x16 operation.

Table 380. Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 1

Instruction sequence [1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Read command at positive edge of the clock.

Table continues on the next page...

Table 380. Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 1 (continued)

Instruction sequence [1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	MODE	11b	00h	Single cycle mode.
LUT2 0	DUMMY	11b	84h	First split DUMMY instruction: For a total of 7 dummy cycles, specify 4 in this first instruction.[3]
LUT2 1	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT3 0	READ_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 1	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.18 Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 0

See Configuration for x8 and x16 operation.

Table 381. Sequence: x8 and x16 WRITE with DUMMY split, variable latency, RWDS = 0

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Write command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	84h	First split DUMMY instruction: For a total of 7 dummy cycles, specify 4 in this first instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	WRITE_DDR	11b	FFh	Write the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.19 Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 0

See Configuration for x8 and x16 operation.

Table 382. Sequence: x8 and x16 READ with DUMMY split, variable latency, RWDS = 0

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Read command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	84h	First split DUMMY instruction: For a total of 7 dummy cycles, specify 4 in this first instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	READ_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.20 Sequence: x8 and x16 WRITE without DUMMY split, variable latency

See Configuration for x8 and x16 operation.

Table 383. Sequence: x8 and x16 WRITE without DUMMY split, variable latency

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Write command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	0Dh	For a total of 7 dummy cycles, specify 13 for this instruction.[3]
LUT2 0	WRITE_DDR	11b	FFh	Write the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT2 1	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the DUMMY instruction must specify 2**n* – 1 cycles.

41.6.2.21 Sequence: x8 and x16 READ without DUMMY split, variable latency

See Configuration for x8 and x16 operation.

Table 384. Sequence: x8 and x16 READ without DUMMY split, variable latency

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Read command at positive edge of the clock.

Table continues on the next page...

Table 384. Sequence: x8 and x16 READ without DUMMY split, variable latency (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	0Dh	For a total of 7 dummy cycles, specify 13 for this instruction.[3]
LUT2 0	READ_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT2 1	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the DUMMY instruction must specify 2**n* – 1 cycles.

41.6.2.22 Sequence: x8 and x16 WRITE with DUMMY split, fixed latency

See Configuration for x8 and x16 operation.

Table 385. Sequence: x8 and x16 WRITE with DUMMY split, fixed latency

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Write command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	8Bh	First split DUMMY instruction: For a total of 13 dummy cycles, specify 11 in this first instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	WRITE_DDR	11b	FFh	Write the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.23 Sequence: x8 and x16 READ with DUMMY split, fixed latency

See Configuration for x8 and x16 operation.

Table 386. Sequence: x8 and x16 READ with DUMMY split, fixed latency

Sequence [1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Send command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.

Table continues on the next page...

Table 386. Sequence: x8 and x16 READ with DUMMY split, fixed latency (continued)

Sequence [1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	8Bh	First split DUMMY instruction: For a total of 13 dummy cycles, specify 11 in this first instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	READ_DDR	11b	FFh	Read the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the first DUUMY instruction must specify *n* – 2 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.24 Sequence: x8 and x16 WRITE without DUMMY split, fixed latency

See Configuration for x8 and x16 operation.

Table 387. Sequence: x8 and x16 WRITE without DUMMY split, fixed latency

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	20h	Write command at positive edge of the clock.
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	8Bh	First split DUMMY instruction: For a total of 7 dummy cycles, specify 13 for this instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	WRITE_DDR	11b	FFh	Write the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the DUMMY instruction must specify 2**n* – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.25 Sequence: x8 and x16 READ without DUMMY split, fixed latency

See Configuration for x8 and x16 operation.

Table 388. Sequence: x8 and x16 READ without DUMMY split, fixed latency

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 0	CMD_DDR	11b	A0h	Write command at positive edge of the clock.

Table continues on the next page...

Table 388. Sequence: x8 and x16 READ without DUMMY split, fixed latency (continued)

Sequence[1]	INSTR*n*[2]	PAD*n*	OPRND*n*	Description
LUT0 1	ADDR_DDR	11b	18h	Send 24-bit row address on eight pads in DDR mode.
LUT1 0	CADDR_DDR	11b	10h	Send 16-bit column address on eight pads in DDR mode.
LUT1 1	DUMMY	11b	8Bh	First split DUMMY instruction: For a total of 7 dummy cycles, specify 13 for this instruction.[3]
LUT2 0	DUMMY	11b	82h	Second split DUMMY instruction: Execute two dummy cycles.[3]
LUT2 1	READ_DDR	11b	FFh	READ the specified number of bytes on eight pads in DDR mode (FFh is the maximum).
LUT3 0	STOP	11b	00h	Stop, sequence completed.

1. Sequence numbering assumes the use of LUT sequence 0 for demonstration purposes. You can program this example into any available LUT sequence.
2. For instruction descriptions, see Programmable sequence engine instructions.
3. For *n* dummy cycles, the DUMMY instruction must specify 2**n* – 1 cycles. The second DUMMY instruction then specifies two dummy cycles.

41.6.2.26 SNAND sequences (IPS write and IPS read)

This section contains example sequences for accessing SNAND flash devices in NAND and NOR modes. Before generating read commands, appropriate modes such as QUAD, NAND/NOR are required to be enabled. You need to read the DLP pattern for the selection of proper DLL data sampling point.

41.6.2.26.1 NAND mode

ENPGM = 1

Table 389. CMD_WRITE_STATUS, Addr: 10h (through SFAR send this address), status: 01h

Instruction	Pad	Operand	Comment
CMD	00b	1Fh	Command for Set Feature
ADDR	00b	8d	8-bit write address
WRITE	00b	1d	Write 1 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Write QE = 1

Table 390. CMD_WRITE_STATUS, Addr: B0h (through SFAR send this address), status: 01h

Instruction	Pad	Operand	Comment
CMD	00b	1Fh	Command for Set Feature
ADDR	00b	8d	8-bit write address
WRITE	00b	1d	Write 1 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Read QE = 1

Table 391. CMD_READ_STATUS, Addr: B0h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

NOTE

ENPGM is Enable Program, QE is Quad Enable, are configuration bits from SNAND Device configuration register.

DLP Read

Table 392. CMD_READ_DATA, Addr: 8h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Read status

Table 393. CMD_READ_STATUS, Addr: C0h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

NOTE

SNAND flash require read commands to happen in two steps. Step one to open a page and next command to read number of bytes. IPS interface is used for transactions.

Page selection

Table 394. FLASH_TRANS_WRITE, CMD_QUAD_PAGE_PROGRAM, Addr: 100h (through SFAR send this address), size: 192

Instruction	Pad	Operand	Comment
CMD	00b	32h	Command for program Load x4N in quad mode
ADDR	00b	16d	16-bit write address
WRITE	10b	size	This is the total size of the data transfer in WRITE mode. Based on the requirement, this operand value may vary.
STOP	00b	0d	Stop, instruction over

Read bytes

Table 395. CMD_FAST_READ_QUAD_IO, Addr: 100h (through SFAR send this address), size: 192 (with DUMMY split)

Instruction	Pad	Operand	Comment
CMD	00b	6Bh	Command for read From Cache x4 in quad mode
ADDR	00b	16d	16-bit read address
DUMMY	10b	6d	First DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then first DUMMY instruction value is 6
DUMMY	10b	2d	Second DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then second DUMMY instruction value is 2
Read	10b	size	This is the total size of the data transfer in READ mode. Based on the requirement, this operand value may vary.
STOP	00b	0d	Stop, instruction over

41.6.2.26.2 NOR mode

Write QE = 1

Table 396. CMD_WRITE_STATUS, Addr: B0h (through SFAR send this address), status: 01h

Instruction	Pad	Operand	Comment
CMD	00b	1Fh	Command for Set Feature
ADDR	00b	8d	8-bit write address
WRITE	00b	1d	Write 1 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Read QE = 1

Table 397. CMD_READ_STATUS, Addr: B0h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

NOTE

ENPGM is Enable Program, QE is Quad Enable, are configuration bits from SNAND Device configuration register.

DLP Read

Table 398. CMD_READ_DATA, Addr: 8h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Read status

Table 399. CMD_READ_STATUS, Addr: 60h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	05h	Command for Read Status
ADDR	00b	8d	8-bit read address
READ	00b	2d	Read 2 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Write status

Table 400. CMD_WRITE_STATUS, Addr: 60h (through SFAR send this address), status: 02h

Instruction	Pad	Operand	Comment
CMD	00b	1Fh	Command for Set Feature
ADDR	00b	8d	8-bit write address
READ	00b	1d	Write 1 byte of data on single PAD
STOP	00b	0d	Stop, instruction over

Page open

Table 401. FLASH_TRANS_WRITE, CMD_QUAD_PAGE_PROGRAM, Addr: 100h (through SFAR send this address)

Instruction	Pad	Operand	Comment
CMD	00b	32h	Command for program Load x4N in quad mode
ADDR	00b	16d	16-bit write address
WRITE	10b	size	This is the total size of the data transfer in WRITE mode. Based on the requirement, this operand value may vary.
STOP	00b	0d	Stop, instruction over

Read bytes

Table 402. CMD_FAST_READ, Addr: 100h (through SFAR send this address), size: 192

Instruction	Pad	Operand	Comment
CMD	00b	0Bh	Command for Read From Cache x1
DUMMY	00b	6d	First DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then first DUMMY instruction value is 6
DUMMY	00b	2d	Second DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then second DUMMY instruction value is 2
ADDR	00b	16d	16-bit read address
DUMMY	00b	6d	First DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then first DUMMY instruction value is 6
DUMMY	00b	2d	Second DUMMY instruction. If N=8 is the required total number of DUMMY cycles, then second DUMMY instruction value is 2

Table continues on the next page...

Table 402. CMD_FAST_READ, Addr: 100h (through SFAR send this address), size: 192 (continued)

Instruction	Pad	Operand	Comment
Read	10b	size	This is the total size of the data transfer in READ mode. Based on the requirement, this operand value may vary
STOP	00b	0d	Stop, instruction over

41.7 Memory map and register definition

This section provides the memory map and register definitions for the XSPI module

41.7.1 Register write access

Following are the write access restriction terms that apply to all the registers:

- Register write access restriction

  For each register field, the write access conditions are specified in the detailed register description.

  The following table provides a description of the write access conditions. If, for a specific register bit or field, none of the given write access conditions is fulfilled, any write attempt to this register bit or field is ignored without any notification. The values of the bits or fields are not changed.

  The condition term [A or B] indicates that the register or field can be written to if at least one of the conditions is fulfilled.

Table 403. Register write access restrictions

Condition	Description
Anytime mode	No write access restriction
Disabled mode	Write access only if MCR[MDIS] = 1
Normal mode	Write access only if the module is in normal mode

- Register write access requirements

  You can access all registers using 8-bit, 16-bit, and 32-bit wide operations. For some of the registers, at least a 16-bit or 32-bit wide write access is required to ensure correct operation. This write access requirement is stated in the detailed register description for each affected register.

41.7.2 XSPI register descriptions

Attempts to access the following addresses do not result in transfer errors:

- 4h
- 64h
- 138h
- 188h
- 18Ch
- 198h

41.7.2.1 XSPI memory map

XSPI0 base address: 4018_4000h

XSPI1 base address: 4018_5000h

XSPI2 base address: 4041_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Configuration (MCR)	32	RW	000F_4040h
8h	IP Configuration (IPCR)	32	R	0000_0000h
Ch	Flash Memory Configuration (FLSHCR)	32	RW	0001_0303h
10h	Buffer Configuration (BUF0CR)	32	RW	0000_0000h
14h	Buffer Configuration (BUF1CR)	32	RW	0000_0001h
18h	Buffer Configuration (BUF2CR)	32	RW	0000_0002h
1Ch	Buffer Configuration (BUF3CR)	32	RW	8000_0003h
20h	Buffer Generic Configuration (BFGENCR)	32	RW	0000_0010h
24h	SOC Configuration (SOCCR)	32	RW	0000_0000h
30h - 38h	Buffer n Top Index (BUF0IND - BUF2IND)	32	RW	0000_0000h
50h	AHB Write Configuration (AWRCR)	32	RW	0000_0000h
60h	DLL Flash Memory A Configuration (DLLCRA)	32	RW	0120_0000h
100h	Serial Flash Memory Address (SFAR)	32	R	0000_0000h
104h	Serial Flash Memory Address Configuration (SFACR)	32	RW	0000_0800h
108h	Sampling (SMPR)	32	RW	0000_0000h
10Ch	RX Buffer Status (RBSR)	32	R	0000_0000h
110h	RX Buffer Control (RBCT)	32	RW	0000_0000h
120h	AHB Write Status (AWRSR)	32	R	0000_0000h
12Ch	DLL Status (DLLSR)	32	R	8000_8000h
130h	Data Learning Configuration (DLCR)	32	RW	40FF_40FFh
134h	Data Learning Status Flash Memory (DLSR_FA)	32	R	0000_0000h
150h	TX Buffer Status (TBSR)	32	R	0000_0000h
154h	TX Buffer Data (TBDR)	32	RW	0000_0000h
158h	TX Buffer Control (TBCT)	32	RW	0000_0000h
15Ch	Status (SR)	32	R	0200_0000h
160h	Flag (FR)	32	RW	0800_0000h
164h	Interrupt and DMA Request Enable (RSER)	32	RW	0000_0000h
168h	Sequence Suspend Status (SPNDST)	32	R	0000_0000h
16Ch	Sequence Pointer Clear (SPTRCLR)	32	RW	0100_0000h
180h	Serial Flash Memory Top Address (SFA1AD)	32	RW	See section
184h	Serial Flash Memory Top Address (SFA2AD)	32	RW	See section

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
190h	Data Learning Pattern (DLPR)	32	RW	AA55_3443h
194h	Flash Memory A Failing Address Status (FAILA_ADDR)	32	R	FFFF_FFFFh
200h - 2FCh	RX Buffer Data (RBDR0 - RBDR63)	32	R	0000_0000h
300h	LUT Key (LUTKEY)	32	RW	5AF0_5AF0h
304h	LUT Lock Configuration (LCKCR)	32	RW	0000_0002h
310h - 44Ch	Lookup Table (LUT0 - LUT79)	32	RW	See section
480h - 4BCh	AHB Buffer n Sub Buffer m Start and End Address Range (BUF0_0_ADDR_RANGE - BUF3_3_ADDR_RANGE)	32	RW	0000_0000h
4C0h	AHB Buffer Status (AHB_BUF_STATUS)	32	R	0000_0000h
4C4h	AHB Buffer Hit/Miss Performance Monitor Control (AHB_PERF_CTRL)	32	RW	0000_0000h
4C8h	AHB Performance Monitor Time Counter (AHB_PERF_TIME_CNT)	32	R	0000_0000h
4CCh - 4D8h	AHB Buffer n Performance Monitor (AHB_PERF_BUF0 - AHB_PERF_BUF3)	32	R	0000_0000h
4DCh	AHB HREADY Timeout (AHRDYTO)	32	RW	0000_0003h
4E0h	AHB Error Payload High (AHB_ERR_PAYLOAD_HI)	32	RW	0000_0000h
4E4h	AHB Error Payload Low (AHB_ERR_PAYLOAD_LO)	32	RW	0000_0000h
4E8h	AHB Read Error Address (AHB_RD_ERR_ADDR)	32	R	0000_0000h
4ECh	AHB Read Error Initiator ID (AHB_RD_ERR_MID)	32	R	0000_0000h
4F0h	Suspend Transaction Address (SPNDST_ADDR)	32	R	0000_0000h
4F4h	Page Program Wait Time Counter (PPWF_TCNT)	32	RW	00FF_FFFFh
4F8h	Page Program Wait Read Status (PPW_RDSR)	32	RW	0000_0000h
520h	IPED Context Initial Vector (IPEDCTX0IV0)	32	RW	0000_0000h
524h	IPED Context Initial Vector (IPEDCTX0IV1)	32	RW	0000_0000h
528h	IPED Start Address Region (IPEDCTX0START)	32	RW	0000_0000h
52Ch	IPED End Address Region (IPEDCTX0END)	32	RW	0000_0000h
530h	IPED Context Additional Authentication Data (IPEDCTX0AAD0)	32	RW	0000_0000h
534h	IPED Context Additional Authentication Data (IPEDCTX0AAD1)	32	RW	0000_0000h
540h	IPED Context Initial Vector (IPEDCTX1IV0)	32	RW	0000_0000h
544h	IPED Context Initial Vector (IPEDCTX1IV1)	32	RW	0000_0000h
548h	IPED Start Address Region (IPEDCTX1START)	32	RW	0000_0000h
54Ch	IPED End Address Region (IPEDCTX1END)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
550h	IPED Context Additional Authentication Data (IPEDCTX1AAD0)	32	RW	0000_0000h
554h	IPED Context Additional Authentication Data (IPEDCTX1AAD1)	32	RW	0000_0000h
560h	IPED Context Initial Vector (IPEDCTX2IV0)	32	RW	0000_0000h
564h	IPED Context Initial Vector (IPEDCTX2IV1)	32	RW	0000_0000h
568h	IPED Start Address Region (IPEDCTX2START)	32	RW	0000_0000h
56Ch	IPED End Address Region (IPEDCTX2END)	32	RW	0000_0000h
570h	IPED Context Additional Authentication Data (IPEDCTX2AAD0)	32	RW	0000_0000h
574h	IPED Context Additional Authentication Data (IPEDCTX2AAD1)	32	RW	0000_0000h
580h	IPED Context Initial Vector (IPEDCTX3IV0)	32	RW	0000_0000h
584h	IPED Context Initial Vector (IPEDCTX3IV1)	32	RW	0000_0000h
588h	IPED Start Address Region (IPEDCTX3START)	32	RW	0000_0000h
58Ch	IPED End Address Region (IPEDCTX3END)	32	RW	0000_0000h
590h	IPED Context Additional Authentication Data (IPEDCTX3AAD0)	32	RW	0000_0000h
594h	IPED Context Additional Authentication Data (IPEDCTX3AAD1)	32	RW	0000_0000h
5A0h	IPED Context Initial Vector (IPEDCTX4IV0)	32	RW	0000_0000h
5A4h	IPED Context Initial Vector (IPEDCTX4IV1)	32	RW	0000_0000h
5A8h	IPED Start Address Region (IPEDCTX4START)	32	RW	0000_0000h
5ACh	IPED End Address Region (IPEDCTX4END)	32	RW	0000_0000h
5B0h	IPED Context Additional Authentication Data (IPEDCTX4AAD0)	32	RW	0000_0000h
5B4h	IPED Context Additional Authentication Data (IPEDCTX4AAD1)	32	RW	0000_0000h
5C0h	IPED Context Initial Vector (IPEDCTX5IV0)	32	RW	0000_0000h
5C4h	IPED Context Initial Vector (IPEDCTX5IV1)	32	RW	0000_0000h
5C8h	IPED Start Address Region (IPEDCTX5START)	32	RW	0000_0000h
5CCh	IPED End Address Region (IPEDCTX5END)	32	RW	0000_0000h
5D0h	IPED Context Additional Authentication Data (IPEDCTX5AAD0)	32	RW	0000_0000h
5D4h	IPED Context Additional Authentication Data (IPEDCTX5AAD1)	32	RW	0000_0000h
5E0h	IPED Context Initial Vector (IPEDCTX6IV0)	32	RW	0000_0000h
5E4h	IPED Context Initial Vector (IPEDCTX6IV1)	32	RW	0000_0000h
5E8h	IPED Start Address Region (IPEDCTX6START)	32	RW	0000_0000h
5ECh	IPED End Address Region (IPEDCTX6END)	32	RW	0000_0000h
5F0h	IPED Context Additional Authentication Data (IPEDCTX6AAD0)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
5F4h	IPED Context Additional Authentication Data (IPEDCTX6AAD1)	32	RW	0000_0000h
600h	IPED Context Initial Vector (IPEDCTX7IV0)	32	RW	0000_0000h
604h	IPED Context Initial Vector (IPEDCTX7IV1)	32	RW	0000_0000h
608h	IPED Start Address Region (IPEDCTX7START)	32	RW	0000_0000h
60Ch	IPED End Address Region (IPEDCTX7END)	32	RW	0000_0000h
610h	IPED Context Additional Authentication Data (IPEDCTX7AAD0)	32	RW	0000_0000h
614h	IPED Context Additional Authentication Data (IPEDCTX7AAD1)	32	RW	0000_0000h
620h	IPED Context Initial Vector (IPEDCTX8IV0)	32	RW	0000_0000h
624h	IPED Context Initial Vector (IPEDCTX8IV1)	32	RW	0000_0000h
628h	IPED Start Address Region (IPEDCTX8START)	32	RW	0000_0000h
62Ch	IPED End Address Region (IPEDCTX8END)	32	RW	0000_0000h
630h	IPED Context Additional Authentication Data (IPEDCTX8AAD0)	32	RW	0000_0000h
634h	IPED Context Additional Authentication Data (IPEDCTX8AAD1)	32	RW	0000_0000h
640h	IPED Context Initial Vector (IPEDCTX9IV0)	32	RW	0000_0000h
644h	IPED Context Initial Vector (IPEDCTX9IV1)	32	RW	0000_0000h
648h	IPED Start Address Region (IPEDCTX9START)	32	RW	0000_0000h
64Ch	IPED End Address Region (IPEDCTX9END)	32	RW	0000_0000h
650h	IPED Context Additional Authentication Data (IPEDCTX9AAD0)	32	RW	0000_0000h
654h	IPED Context Additional Authentication Data (IPEDCTX9AAD1)	32	RW	0000_0000h
660h	IPED Context Initial Vector (IPEDCTX10IV0)	32	RW	0000_0000h
664h	IPED Context Initial Vector (IPEDCTX10IV1)	32	RW	0000_0000h
668h	IPED Start Address Region (IPEDCTX10START)	32	RW	0000_0000h
66Ch	IPED End Address Region (IPEDCTX10END)	32	RW	0000_0000h
670h	IPED Context Additional Authentication Data (IPEDCTX10AAD0)	32	RW	0000_0000h
674h	IPED Context Additional Authentication Data (IPEDCTX10AAD1)	32	RW	0000_0000h
680h	IPED Context Initial Vector (IPEDCTX11IV0)	32	RW	0000_0000h
684h	IPED Context Initial Vector (IPEDCTX11IV1)	32	RW	0000_0000h
688h	IPED Start Address Region (IPEDCTX11START)	32	RW	0000_0000h
68Ch	IPED End Address Region (IPEDCTX11END)	32	RW	0000_0000h
690h	IPED Context Additional Authentication Data (IPEDCTX11AAD0)	32	RW	0000_0000h
694h	IPED Context Additional Authentication Data (IPEDCTX11AAD1)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
6A0h	IPED Context Initial Vector (IPEDCTX12IV0)	32	RW	0000_0000h
6A4h	IPED Context Initial Vector (IPEDCTX12IV1)	32	RW	0000_0000h
6A8h	IPED Start Address Region (IPEDCTX12START)	32	RW	0000_0000h
6ACh	IPED End Address Region (IPEDCTX12END)	32	RW	0000_0000h
6B0h	IPED Context Additional Authentication Data (IPEDCTX12AAD0)	32	RW	0000_0000h
6B4h	IPED Context Additional Authentication Data (IPEDCTX12AAD1)	32	RW	0000_0000h
6C0h	IPED Context Initial Vector (IPEDCTX13IV0)	32	RW	0000_0000h
6C4h	IPED Context Initial Vector (IPEDCTX13IV1)	32	RW	0000_0000h
6C8h	IPED Start Address Region (IPEDCTX13START)	32	RW	0000_0000h
6CCh	IPED End Address Region (IPEDCTX13END)	32	RW	0000_0000h
6D0h	IPED Context Additional Authentication Data (IPEDCTX13AAD0)	32	RW	0000_0000h
6D4h	IPED Context Additional Authentication Data (IPEDCTX13AAD1)	32	RW	0000_0000h
6E0h	IPED Context Initial Vector (IPEDCTX14IV0)	32	RW	0000_0000h
6E4h	IPED Context Initial Vector (IPEDCTX14IV1)	32	RW	0000_0000h
6E8h	IPED Start Address Region (IPEDCTX14START)	32	RW	0000_0000h
6ECh	IPED End Address Region (IPEDCTX14END)	32	RW	0000_0000h
6F0h	IPED Context Additional Authentication Data (IPEDCTX14AAD0)	32	RW	0000_0000h
6F4h	IPED Context Additional Authentication Data (IPEDCTX14AAD1)	32	RW	0000_0000h
700h	IPED Context Initial Vector (IPEDCTX15IV0)	32	RW	0000_0000h
704h	IPED Context Initial Vector (IPEDCTX15IV1)	32	RW	0000_0000h
708h	IPED Start Address Region (IPEDCTX15START)	32	RW	0000_0000h
70Ch	IPED End Address Region (IPEDCTX15END)	32	RW	0000_0000h
710h	IPED Context Additional Authentication Data (IPEDCTX15AAD0)	32	RW	0000_0000h
714h	IPED Context Additional Authentication Data (IPEDCTX15AAD1)	32	RW	0000_0000h
720h	IPED Function Control (IPEDCTRL)	32	RW	0000_0000h
728h	IPED Context Control 0 (IPEDCTXCTRL0)	32	RW	5555_5555h
72Ch	IPED Context Control 1 (IPEDCTXCTRL1)	32	RW	AAAA_AAAAh
800h	Flash Region Word 0 – Start Address (FRAD0_WORD0)	32	RW	0000_0000h
804h	Flash Region Word 1 – End Address (FRAD0_WORD1)	32	RW	0000_FFFFh
808h	Flash Region Word 2 – Privileges (FRAD0_WORD2)	32	RW	0000_0000h
80Ch	Flash Region Word 3 – Lock Control (FRAD0_WORD3)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
810h	Flash Region Word 4 – Compare Address Status (FRAD0_WORD4)	32	R	0000_0000h
814h	Flash Region Word 5 – Compare Status Data (FRAD0_WORD5)	32	R	0000_0000h
820h	Flash Region Word 0 – Start Address (FRAD1_WORD0)	32	RW	0000_0000h
824h	Flash Region Word 1 – End Address (FRAD1_WORD1)	32	RW	0000_FFFFh
828h	Flash Region Word 2 – Privileges (FRAD1_WORD2)	32	RW	0000_0000h
82Ch	Flash Region Word 3 – Lock Control (FRAD1_WORD3)	32	RW	0000_0000h
830h	Flash Region Word 4 – Compare Address Status (FRAD1_WORD4)	32	R	0000_0000h
834h	Flash Region Word 5 – Compare Status Data (FRAD1_WORD5)	32	R	0000_0000h
840h	Flash Region Word 0 – Start Address (FRAD2_WORD0)	32	RW	0000_0000h
844h	Flash Region Word 1 – End Address (FRAD2_WORD1)	32	RW	0000_FFFFh
848h	Flash Region Word 2 – Privileges (FRAD2_WORD2)	32	RW	0000_0000h
84Ch	Flash Region Word 3 – Lock Control (FRAD2_WORD3)	32	RW	0000_0000h
850h	Flash Region Word 4 – Compare Address Status (FRAD2_WORD4)	32	R	0000_0000h
854h	Flash Region Word 5 – Compare Status Data (FRAD2_WORD5)	32	R	0000_0000h
860h	Flash Region Word 0 – Start Address (FRAD3_WORD0)	32	RW	0000_0000h
864h	Flash Region Word 1 – End Address (FRAD3_WORD1)	32	RW	0000_FFFFh
868h	Flash Region Word 2 – Privileges (FRAD3_WORD2)	32	RW	0000_0000h
86Ch	Flash Region Word 3 – Lock Control (FRAD3_WORD3)	32	RW	0000_0000h
870h	Flash Region Word 4 – Compare Address Status (FRAD3_WORD4)	32	R	0000_0000h
874h	Flash Region Word 5 – Compare Status Data (FRAD3_WORD5)	32	R	0000_0000h
880h	Flash Region Word 0 – Start Address (FRAD4_WORD0)	32	RW	0000_0000h
884h	Flash Region Word 1 – End Address (FRAD4_WORD1)	32	RW	0000_FFFFh
888h	Flash Region Word 2 – Privileges (FRAD4_WORD2)	32	RW	0000_0000h
88Ch	Flash Region Word 3 – Lock Control (FRAD4_WORD3)	32	RW	0000_0000h
890h	Flash Region Word 4 – Compare Address Status (FRAD4_WORD4)	32	R	0000_0000h
894h	Flash Region Word 5 – Compare Status Data (FRAD4_WORD5)	32	R	0000_0000h
8A0h	Flash Region Word 0 – Start Address (FRAD5_WORD0)	32	RW	0000_0000h
8A4h	Flash Region Word 1 – End Address (FRAD5_WORD1)	32	RW	0000_FFFFh
8A8h	Flash Region Word 2 – Privileges (FRAD5_WORD2)	32	RW	0000_0000h
8ACh	Flash Region Word 3 – Lock Control (FRAD5_WORD3)	32	RW	0000_0000h
8B0h	Flash Region Word 4 – Compare Address Status (FRAD5_WORD4)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
8B4h	Flash Region Word 5 – Compare Status Data (FRAD5_WORD5)	32	R	0000_0000h
8C0h	Flash Region Word 0 – Start Address (FRAD6_WORD0)	32	RW	0000_0000h
8C4h	Flash Region Word 1 – End Address (FRAD6_WORD1)	32	RW	0000_FFFFh
8C8h	Flash Region Word 2 – Privileges (FRAD6_WORD2)	32	RW	0000_0000h
8CCh	Flash Region Word 3 – Lock Control (FRAD6_WORD3)	32	RW	0000_0000h
8D0h	Flash Region Word 4 – Compare Address Status (FRAD6_WORD4)	32	R	0000_0000h
8D4h	Flash Region Word 5 – Compare Status Data (FRAD6_WORD5)	32	R	0000_0000h
8E0h	Flash Region Word 0 – Start Address (FRAD7_WORD0)	32	RW	0000_0000h
8E4h	Flash Region Word 1 – End Address (FRAD7_WORD1)	32	RW	0000_FFFFh
8E8h	Flash Region Word 2 – Privileges (FRAD7_WORD2)	32	RW	0000_0000h
8ECh	Flash Region Word 3 – Lock Control (FRAD7_WORD3)	32	RW	0000_0000h
8F0h	Flash Region Word 4 – Compare Address Status (FRAD7_WORD4)	32	R	0000_0000h
8F4h	Flash Region Word 5 – Compare Status Data (FRAD7_WORD5)	32	R	0000_0000h
8FCh	SFP Arbitration Lock Timeout Counter (SFP_ARB_TIMEOUT)	32	RW	000F_FFFFh
900h	Target Group Initiator Domain Access Descriptor (TG0MDAD)	32	RW	0000_0000h
904h	Target Group SFAR Address (TGSFAR)	32	R	0000_0000h
908h	Target Group SFAR Status (TGSFARS)	32	RW	0000_0000h
90Ch	Target Group IP Configuration Status (TGIPCRS)	32	RW	0000_0000h
910h	Target Group Initiator Domain Access Descriptor (TG1MDAD)	32	RW	0000_0000h
920h	Initiator Global Configuration (MGC)	32	RW	A800_0000h
924h	Initiator Read Command (MRC)	32	RW	0050_0E07h
928h	Initiator Timeout (MTO)	32	RW	0000_FFFFh
92Ch	Flash Sequence Request (FLSEQREQ)	32	R	0000_0000h
930h	FSM Status (FSMSTAT)	32	R	0000_0000h
934h	IPS Error (IPSERROR)	32	RW	0000_0000h
938h	Error Status (ERRSTAT)	32	RW	0000_0000h
93Ch	Interrupt Enable (INT_EN)	32	RW	0000_0000h
958h	IP Configuration (SFP_TG_IPCR)	32	RW	0000_0000h
95Ch	Serial Flash Memory Address (SFP_TG_SFAR)	32	RW	0000_0000h
960h - 964h	LUT Access Enable (SFP_LUT_EN0 - SFP_LUT_EN1)	32	RW	0000_0000h
A7Ch	SFP Lock Error Address (SFP_LOCK_ERR_ADDR)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
A80h	IP Configuration (SFP_TG_SUB_1_IPCR)	32	RW	0000_0000h
A84h	Serial Flash Memory Address (SFP_TG_SUB_1_SFAR)	32	RW	0000_0000h
A88h	Target Group SFAR Address (TGSFAR_SUB_1)	32	R	0000_0000h
A8Ch	Target Group SFAR Status (TGSFARS_SUB_1)	32	RW	0000_0000h
A90h	Target Group n IP Configuration Status (TGIPCRS_SUB_1)	32	RW	0000_0000h

41.7.2.2 Module Configuration (MCR)

Offset

Register	Offset
MCR	0h

Function

Configures XSPI operation.

NOTE

Coming out of reset and after the initial write to MCR, when XSPI exits module disable mode (MDIS is 0), you must write 1 to SWRSTSD and release it.

Special write access is permitted in different modes:

- Disabled mode: CKN_FA_EN, DQS_FA_SEL, ISD3FA, ISD2FA,
- Anytime mode: All other fields

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	Reserved	Reserved		Reserved	CKN_FA_...	DQS_FA_SEL		Reserved		X16_EN	X16_DB_...	Reserved	Reserved	ISD3FA	ISD2FA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOZE	MDIS	Reserved	DLPEN	0	0		VAR_LAT...	DDR_EN	DQS_EN	DQS_LAT...	DQS_OUT...	END_CFG		SWRSTHD	SWRSTSD
W					CLR_TXF	CLR_RXF	IPS_TG_...									
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29-28 —	Reserved
27 —	Reserved
26 CKN_FA_EN	CKN Pad For Flash Memory A Enables the clock on the differential SLCKFA_N pad for flash memory A. 0b - Disables 1b - Enables
25-24 DQS_FA_SEL	DQS Clock For Sampling Read Data Flash Memory A Selects the DQS clock for sampling read data at the flash memory A XSPI port. **NOTE** • If you select DQS pad loopback to access port A, you cannot program the same port for eternal DQS and vice-versa. • To support DQS pad loopback, the DQS pin on the board must be left floating. 00b - Reserved 01b - Dummy pad loopback 10b - DQS pad loopback 11b - External DQS
23-22 —	Reserved
21 X16_EN	X16 Mode Enable Enable the X16 mode of operation. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
XSPI0	—	MCR

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>XSPI1</td><td>MCR</td><td>—</td></tr>
<tr><td>XSPI2</td><td>MCR</td><td>—</td></tr>
</table>
0b - Disables (x8, x4, x2, x1 modes remain enabled)
1b - Enables
</td></tr>
<tr><td>20
X16_DB_CA_EN</td><td>X16 Databus Enable
Enables address and command information to pass through the X16 databus along with data.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>XSPI0</td><td>—</td><td>MCR</td></tr>
<tr><td>XSPI1</td><td>MCR</td><td>—</td></tr>
<tr><td>XSPI2</td><td>MCR</td><td>—</td></tr>
</table>
0b - Data only
1b - Address and command information along with data
</td></tr>
<tr><td>19
—</td><td>Reserved</td></tr>
<tr><td>18
—</td><td>Reserved</td></tr>
<tr><td>17
ISD3FA</td><td>Idle Signal Drive IOFA[3] Flash Memory A
Specifies the logic level of the XSPI IOFA[3] output in the inactive state. See Driving flash memory control signals in single and dual modes.
0b - Logic low
1b - Logic high</td></tr>
<tr><td>16
ISD2FA</td><td>Idle Signal Drive IOFA[2] Flash Memory A
Specifies the logic level of the XSPI IOFA[2] output in the inactive state. See Driving flash memory control signals in single and dual modes.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Logic low 1b - Logic high
15 DOZE	Doze Enable Enables the Module Disable mode mechanism. See Module Disable mode for details. 0b - Disables 1b - Enables
14 MDIS	Module Disable Allows external logic to stop the clock to XSPI non-memory mapped logic. 0b - Enables clocks 1b - Allows external logic to disable clocks
13 —	Reserved
12 DLPEN	Data Learning Pattern Enable Enables the data learning mechanism. See Data learning. 0b - Disables 1b - Enables
11 CLR_TXF	Clear TX FIFO Clears (invalidates) the TX FIFO buffer by resetting the buffer read and write pointers to 0 and returning the transmit counter (TBSR[TRCTR]) to 0. When clearing is complete, the bit returns to 0. **NOTE** You must wait at least five system cycles and three flash cycles after writing 1 to this field. 0b - No action. Always reads 0. 1b - Clears buffer and returns pointers to 0
10 CLR_RXF	Clear RX FIFO Clears (invalidates) the RX FIFO buffer by resetting the buffer read and write pointers to 0 and returning the RX buffer fill level (RBSR[RDBFL]) to 0. When clearing is complete, the bit returns to 0. 0b - No action. Always reads 0. 1b - Clears buffer and returns pointers to 0
9 IPS_TG_RST	Software Reset for IPS Target Group Queue Resets the IPS TG queue. It clears the transaction currently present in the queue along with its status and error registers. If the transaction was already triggered towards external flash (SR[IP_ACC] = 1), then XSPI waits for the CS to deassert to be sure the transaction has finished on external flash. When the transaction

Table continues on the next page...

Table continued from the previous page...

Field	Function
	completes (CS deasserts), XSPI initiates the software reset. IPS_TG_RST remains 1 until software reset completes, then toggles to 0. XSPI also clears the error register for the queue. This field does not reset SFP or XSPI configuration registers. **NOTE** The IPG clock must be running for the software reset to propagate to the queue. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No action 1b - Reset of IPS TG queue is in progress if is having TG Grant When writing 0b - No action 1b - Resets
8 VAR_LAT_EN	Variable Latency Enable Enables variable latency. Where data strobe acts as an output from memory during the command and address (CA) cycles of a read or write transaction. Enable variable latency when dynamic memory refresh operations require additional initial access latency. See OctalRAM support. **NOTE** If enabled, the serial flash memory CS setup time (FLSHCR[TCSS]) must be >= 2. 0b - Fixed latency 1b - Additional initial latency. "Once" the initial latency based on data strobe during the CA phase.
7 DDR_EN	DDR Mode Enable Enables DDR mode. 0b - 2x clock disabled. Only SDR instructions processed 1b - 2x clock enabled. Both SDR and DDR instructions processed
6 DQS_EN	DQS Enable Enables DQS strobe. This field is valid for both the SDR and DDR modes. See DQS sampling method. When DDR is enabled (DDR_EN = 1), XSPI samples the incoming data on both edges of the DQS input. Otherwise, it samples only on one edge. **NOTE** Do not write 0 to this field. 0b - Reserved 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
5 DQS_LAT_EN	DQS Latency Enable Enables latency between read accesses from flash memory in DQS sampling. See DQS sampling method. 0b - Disables 1b - Enables
4 DQS_OUT_EN	DQS Output Enable Enables DQS for output operation. This field is valid when data strobe is also used as an output from the controller during the write data phase. This field is valid for HyperRAM where the data strobe acts as a Read Write Data Strobe (RWDS). See OctalRAM support. **NOTE** DQS Output Enable feature not available for X4 and X1 RAM devices. 0b - Disables 1b - Enables
3-2 END_CFG	Byte Order Selects the byte order for XSPI operation. See Byte ordering – endianness. 00b - 64-bit BE 01b - 32-bit LE 10b - 32-bit BE 11b - 64-bit LE
1 SWRSTHD	Software Reset for AHB Domain Resets the AHB domain. Resetting only the serial flash memory domain or the AHB domain may cause undesirable side effects, so you should reset both domains at the same time. This field does not reset configuration registers. **NOTE** • XSPI must be enabled (MDIS = 0) when you reset the domains. Also, before you deassert the software reset (write 0 to this field), you should disable XSPI (write 1 to MDIS). After you deassert the software resets, you can restart XSPI (write 0 to MDIS). • You must wait for at least three system cycles and three flash cycles after changing the value of this field. 0b - Deasserts reset 1b - Resets
0 SWRSTSD	Software Reset for Serial Flash Memory Domain Resets the serial flash memory domain. Resetting only the serial flash memory domain or the AHB domain may cause undesirable side effects, so you should reset both domains at the same time. This field does not reset configuration registers.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE • XSPI must be enabled (MDIS = 0) when you reset the domains. Also, before you deassert the software reset (write 0 to this field), you should disable XSPI (write 1 to MDIS). After you deassert the software resets, you can restart XSPI (write 0 to MDIS). • You must wait for at least three system cycles and three flash cycles after changing the value of this field. 0b - Deasserts reset 1b - Resets

41.7.2.3 IP Configuration (IPCR)

Offset

Register	Offset
IPCR	8h

Function

This register stores the configured value for an IP-initiated command. It is updated when SFP triggers the IP-initiated command from any target group queue after it passes the required checks and arbitration.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				SEQID				0							Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDATSZ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-24 SEQID	LUT Sequence ID Indicates the sequence index of the LUT. See Lookup table (LUT). Each sequence index can accommodate up to 10 instructions (two instructions per register). A write to this field by the SFP triggers a transaction on the serial flash memory interface.
23-17 —	Reserved
16 —	Reserved
15-0 IDATSZ	IP Data Transfer Size Indicates the data transfer size, in bytes, of the IP command. **NOTE** To avoid tCSM timing violation, the IDATSZ needs to be configured during read or prefetch operation with respect to flash frequency. For details, refer to the memory device datasheet about the tCSM period.

41.7.2.4 Flash Memory Configuration (FLSHCR)

Offset

Register	Offset
FLSHCR	Ch

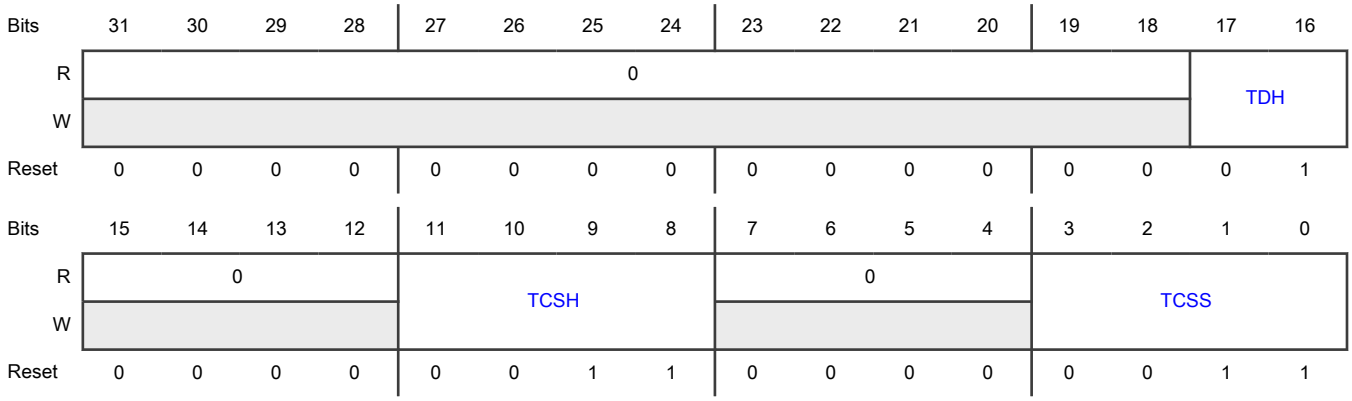
Function

Specifies the timings specific to the flash memory device. XSPI must meet these timings for the device to function correctly.

Special write access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- No IP access in progress (SR[IP_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-18 —	Reserved
17-16 TDH	Serial Flash Memory Data in Hold Time Specifies which clock to use for data alignment. This field helps meet the data-in-hold-time requirement of a flash memory. This field is valid only in the DDR mode. See Data input hold requirement of flash memory. **NOTE** You must write 0 to this field in SDR mode (MCR[DDR_EN] = 0). 00b - Posedge of XSPI internal reference clock 01b - 2x serial flash memory half clock All other values are reserved.
15-12 —	Reserved
11-8 TCSH	Serial Flash Memory CS Hold Time Specifies the CS hold time in serial flash memory clock cycles. This value must be greater than or equal to two. See the memory Datasheet for the exact value.
7-4 —	Reserved
3-0 TCSS	Serial Flash Memory CS Setup Time Specifies the CS setup time in serial flash memory clock cycles. This value must be greater than or equal to two. See the memory Datasheet for the exact value.

41.7.2.5 Buffer Configuration (BUF0CR)

Offset

Register	Offset
BUF0CR	10h

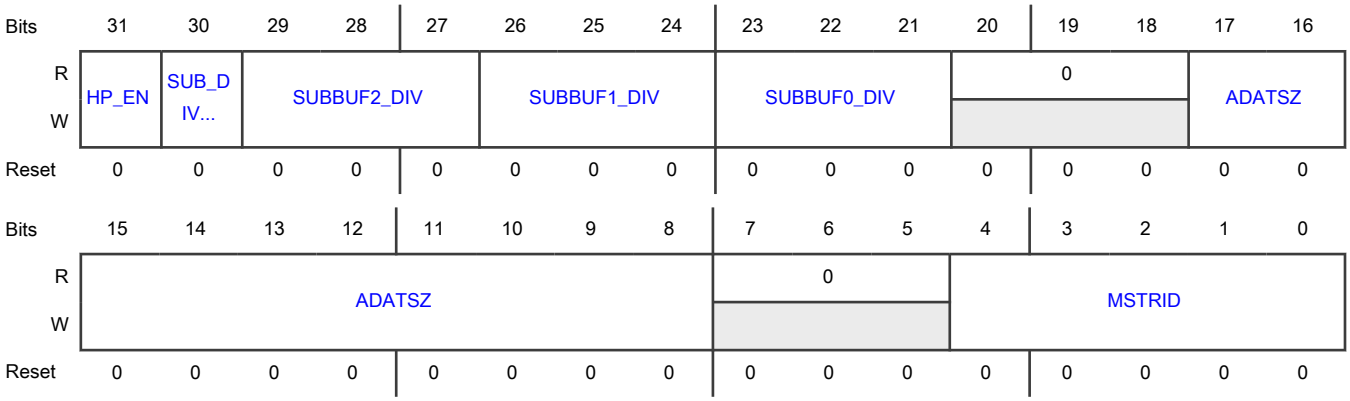
Function

Configures read accesses routed to this AHB buffer, which happens when the initiator ID of the incoming AHB request matches the value in MSTRID. A buffer miss triggers a serial flash memory transaction per the sequence pointed to by BFGENCR[SEQID]. You may also configure the buffer as a high-priority buffer (write 1 to HP_EN).

Special write-access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31 HP_EN	High Priority Enable Enables high priority for the initiator associated with this buffer by assigning a priority higher than the non-priority initiators. A read access by a high priority initiator suspends any ongoing prefetch by a non-priority AHB initiator and XSPI services the high priority initiator. 0b - Disables 1b - Enables
30 SUB_DIV_EN	Buffer Sub-Division Enable Enables sub-buffer division for this buffer. Also enables the address range-based entry scheme for sub-buffer allocation. See Flexible read xAHB buffers. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
29-27: SUBBUF2_DIV 26-24: SUBBUF1_DIV 23-21: SUBBUF0_DIV	Sub-Buffer Division Factor Specifies the value (divisor) to be divided into the buffer size to determine the size of the sub-buffer. 001b - 2 010b - 4 011b - 8 100b - 16 All other values are reserved.
20-18 —	Reserved
17-8 ADATSZ	AHB Data Transfer Size Specifies the read data transfer size of an AHB-triggered read access to serial memory targeted to this buffer in multiples of 8 bytes. For example, a value of 2 sets the transfer size to 16 bytes. If you write 0 to this field, the data size specified in the sequence pointed to by SEQID applies. You must ensure that this transfer size is not greater than the size of the buffer or sub-buffer. **NOTE** For IPED CTR, ADATSZ should be N*8 bytes, where N is number of pages in CTR context region. For IPED GCM, ADATSZ Should be N*32 bytes, where N is number of pages in GCM context region. **NOTE** To avoid tCSM timing violation, the ADATSZ needs to be configured during read or prefetch operation with respect to flash frequency. For details, refer to the memory device datasheet about the tCSM period.
7-5 —	Reserved
4-0 MSTRID	Initiator ID Specifies the ID of the AHB initiator to be associated with this buffer. XSPI routes any AHB read access with this initiator ID to this buffer. You must ensure that the initiator ID associated with each buffer is unique.

41.7.2.6 Buffer Configuration (BUF1CR)

Offset

Register	Offset
BUF1CR	14h

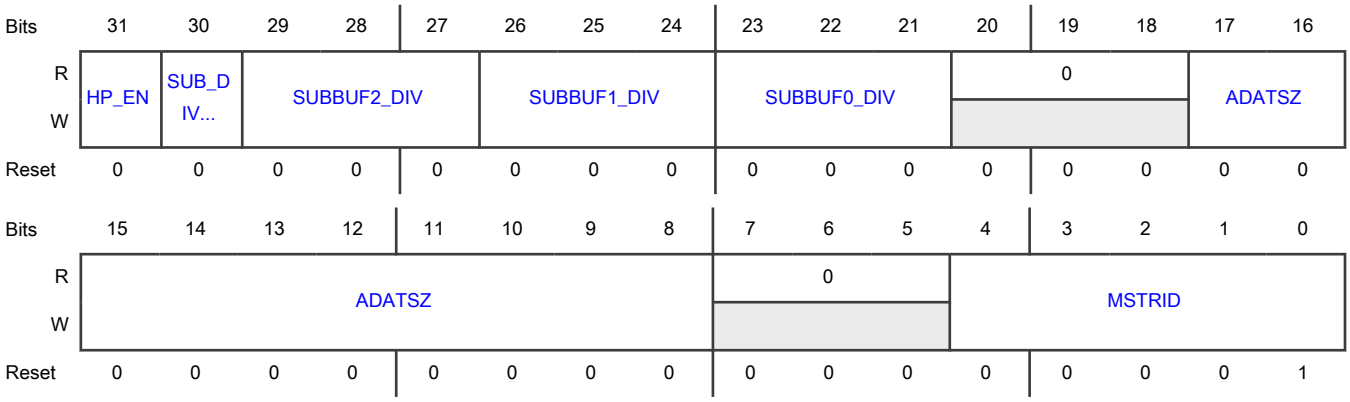
Function

Configures read accesses routed to this AHB buffer, which happens when the initiator ID of the incoming AHB request matches the value in MSTRID. A buffer miss triggers a serial flash memory transaction per the sequence pointed to by BFGENCR[SEQID]. You may also configure the buffer as a high-priority buffer (write 1 to HP_EN).

Special write-access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31 HP_EN	High Priority Enable Enables high priority for the initiator associated with this buffer by assigning a priority higher than the non-priority initiators. A read access by a high priority initiator suspends any ongoing prefetch by a non-priority AHB initiator and XSPI services the high priority initiator. 0b - Disables 1b - Enables
30 SUB_DIV_EN	Buffer Sub-Division Enable Enables sub-buffer division for this buffer. Also enables the address range-based entry scheme for sub-buffer allocation. See Flexible read xAHB buffers. 0b - Disables 1b - Enables
29-27: SUBBUF2_DIV 26-24: SUBBUF1_DIV 23-21: SUBBUF0_DIV	Sub-Buffer Division Factor Specifies the value (divisor) to be divided into the buffer size to determine the size of the sub-buffer. 001b - 2 010b - 4 011b - 8

Table continues on the next page...

Table continued from the previous page...

Field	Function
	100b - 16 All other values are reserved.
20-18 —	Reserved
17-8 ADATSZ	AHB Data Transfer Size Specifies the read data transfer size of an AHB-triggered read access to serial memory targeted to this buffer in multiples of 8 bytes. For example, a value of 2 sets the transfer size to 16 bytes. If you write 0 to this field, the data size specified in the sequence pointed to by SEQID applies. You must ensure that this transfer size is not greater than the size of the buffer or sub-buffer. **NOTE** For IPED CTR, ADATSZ should be N*8 bytes, where N is number of pages in CTR context region. For IPED GCM, ADATSZ Should be N*32 bytes, where N is number of pages in GCM context region. **NOTE** To avoid tCSM timing violation, the ADATSZ needs to be configured during read or prefetch operation with respect to flash frequency. For details, refer to the memory device datasheet about the tCSM period.
7-5 —	Reserved
4-0 MSTRID	Initiator ID Specifies the ID of the AHB initiator to be associated with this buffer. XSPI routes any AHB read access with this initiator ID to this buffer. You must ensure that the initiator ID associated with each buffer is unique.

41.7.2.7 Buffer Configuration (BUF2CR)

Offset

Register	Offset
BUF2CR	18h

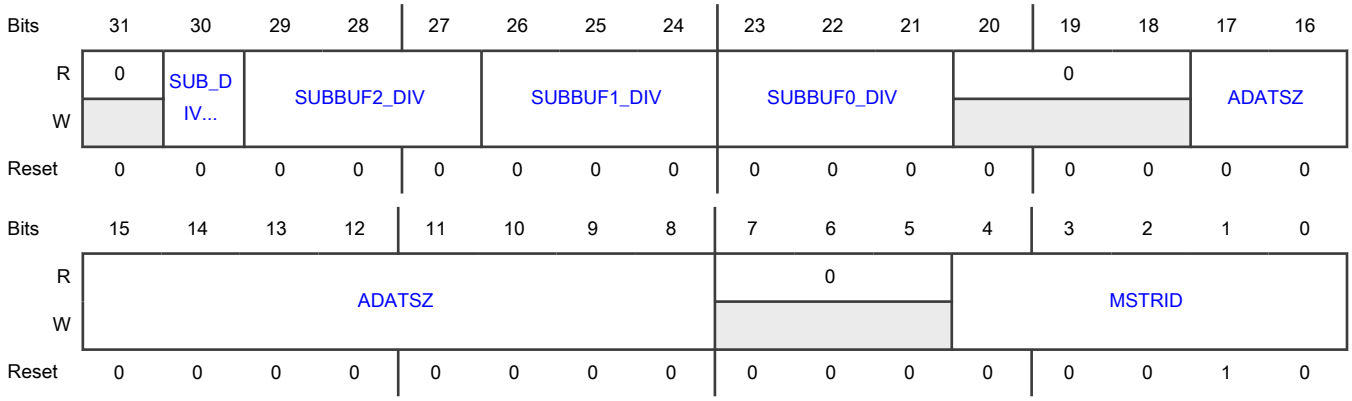
Function

Configures read accesses routed to this AHB buffer, which happens when the initiator ID of the incoming AHB request matches the value in MSTRID. A buffer miss triggers a serial flash memory transaction per the sequence pointed to by BFGENCR[SEQID]. .

Special write-access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31 —	Reserved
30 SUB_DIV_EN	Buffer Sub-Division Enable Enables sub-buffer division for this buffer. Also enables the address range-based entry scheme for sub-buffer allocation. See Flexible read xAHB buffers. 0b - Disables 1b - Enables
29-27: SUBBUF2_DIV 26-24: SUBBUF1_DIV 23-21: SUBBUF0_DIV	Sub-Buffer Division Factor Specifies the value (divisor) to be divided into the buffer size to determine the size of the sub-buffer. 001b - 2 010b - 4 011b - 8 100b - 16 All other values are reserved.
20-18 —	Reserved
17-8 ADATSZ	AHB Data Transfer Size Specifies the read data transfer size of an AHB-triggered read access to serial memory targeted to this buffer in multiples of 8 bytes. For example, a value of 2 sets the transfer size to 16 bytes. If you write 0 to this field, the data size specified in the sequence pointed to by SEQID applies. You must ensure that this transfer size is not greater than the size of the buffer or sub-buffer.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE For IPED CTR, ADATSZ should be N*8 bytes, where N is number of pages in CTR context region. For IPED GCM, ADATSZ Should be N*32 bytes, where N is number of pages in GCM context region. **NOTE** To avoid tCSM timing violation, the ADATSZ needs to be configured during read or prefetch operation with respect to flash frequency. For details, refer to the memory device datasheet about the tCSM period.
7-5 —	Reserved
4-0 MSTRID	Initiator ID Specifies the ID of the AHB initiator to be associated with this buffer. XSPI routes any AHB read access with this initiator ID to this buffer. You must ensure that the initiator ID associated with each buffer is unique.

41.7.2.8 Buffer Configuration (BUF3CR)

Offset

Register	Offset
BUF3CR	1Ch

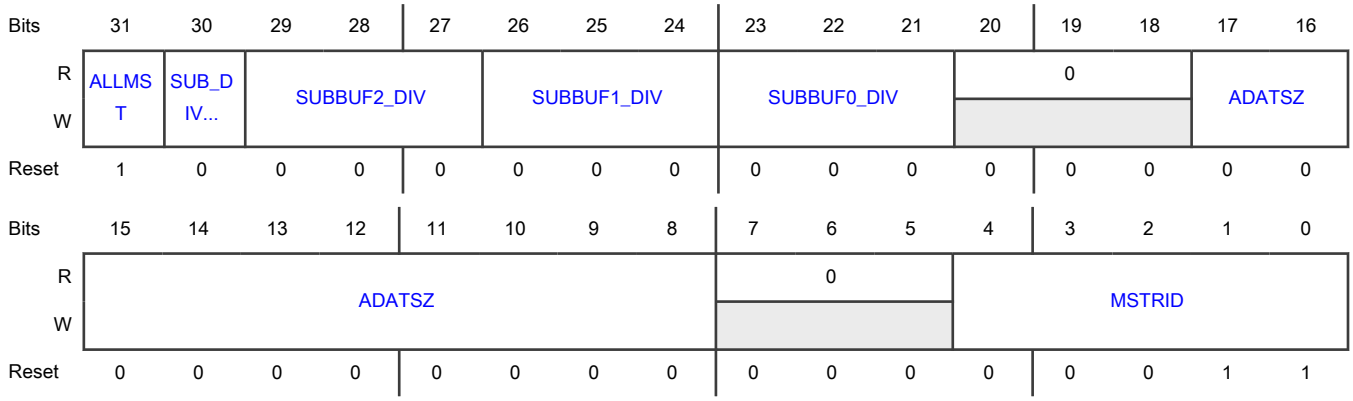
Function

Configures read accesses routed to this AHB buffer, which happens when the initiator ID of the incoming AHB request matches the value in MSTRID. A buffer miss triggers a serial flash memory transaction per the sequence pointed to by BFGENCR[SEQID]. You may also configure the buffer as an all-initiator buffer (write 1 to ALLMST).

Special write-access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31 ALLMST	All Initiator Enable Enables buffer 3 to act as an all-initiator buffer. Any AHB access with an initiator ID not matching the initiator IDs of buffers 0, 1, or 2 is routed to buffer 3. When enabled, the XSPI ignores the MSTRID field of this register. 0b - Disables 1b - Enables
30 SUB_DIV_EN	Buffer Sub-Division Enable Enables sub-buffer division for this buffer. Also enables the address range-based entry scheme for sub-buffer allocation. See Flexible read xAHB buffers. 0b - Disables 1b - Enables
29-27: SUBBUF2_DIV 26-24: SUBBUF1_DIV 23-21: SUBBUF0_DIV	Sub-Buffer Division Factor Specifies the value (divisor) to be divided into the buffer size to determine the size of the sub-buffer. 001b - 2 010b - 4 011b - 8 100b - 16 All other values are reserved.
20-18 —	Reserved
17-8 ADATSZ	AHB Data Transfer Size Specifies the read data transfer size of an AHB-triggered read access to serial memory targeted to this buffer in multiples of 8 bytes. For example, a value of 2 sets the transfer size to 16 bytes. If you write 0 to this field,

Table continues on the next page...

Table continued from the previous page...

Field	Function
	the data size specified in the sequence pointed to by SEQID applies. You must ensure that this transfer size is not greater than the size of the buffer or sub-buffer. **NOTE** For IPED CTR, ADATSZ should be N*8 bytes, where N is number of pages in CTR context region. For IPED GCM, ADATSZ Should be N*32 bytes, where N is number of pages in GCM context region. **NOTE** To avoid tCSM timing violation, the ADATSZ needs to be configured during read or prefetch operation with respect to flash frequency. For details, refer to the memory device datasheet about the tCSM period.
7-5 —	Reserved
4-0 MSTRID	Initiator ID Specifies the ID of the AHB initiator to be associated with this buffer. XSPI routes any AHB read access with this initiator ID to this buffer. You must ensure that the initiator ID associated with each buffer is unique. If ALLMST is set, XSPI ignores this field.

41.7.2.9 Buffer Generic Configuration (BFGENCR)

Offset

Register	Offset
BFGENCR	20h

Function

Configures the behavior of all buffer accesses. Any buffer miss causes the buffer to be flushed and triggers a serial flash memory transaction per the sequence pointed to by SEQID.

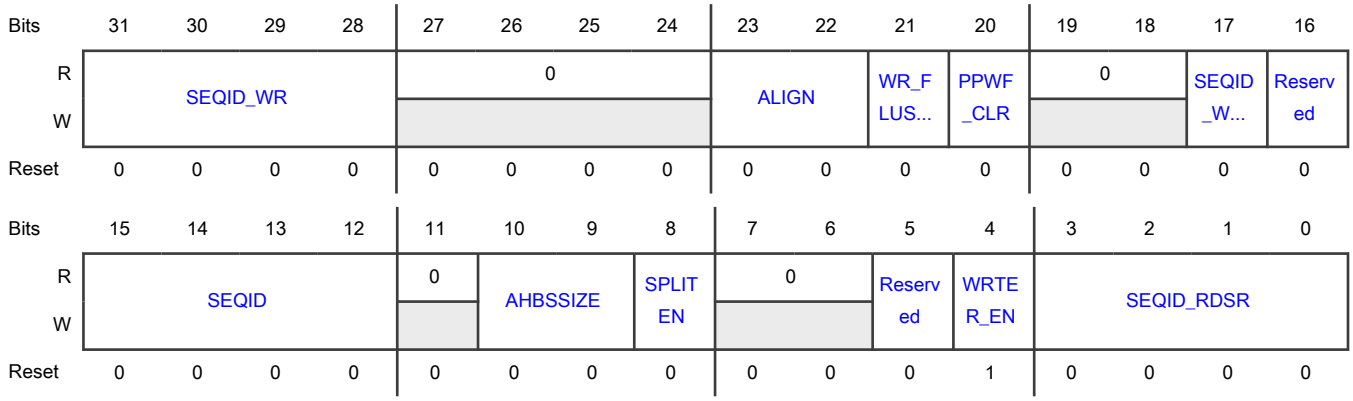
Special write access is permitted if all these conditions apply:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)
- Global configuration lock is disabled (MGC[GCLCK] = 0)

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

Only privilege initiators can write to this register. An attempt to write to it by any other initiator generates a bus transfer error.

Diagram



Fields

Field	Function
31-28 SEQID_WR	Write Sequence ID If you enable write sequence ID (write 1 to SEQID_WR_EN), specifies the write sequence ID of LUT to be selected for an AHB write transaction.
27-24 —	Reserved
23-22 ALIGN	AHB Alignment Specifies the AHB alignment boundary. XSPI breaks the AHB transactions when the below boundary is crossed. **NOTE** This field is not supported in every instance. The following table includes only supported registers. 00b - No limit 01b - 256 bytes 10b - 512 bytes 11b - 1 KB
21 WR_FLUSH_EN	Write Flush Enable Enables the clearing of AHB read prefetch buffers when the same flash address is written by an AHB or IP command.

Instance	Field supported in	Field not supported in
XSPI0	BFGENCR	—
XSPI1	BFGENCR	—
XSPI2	—	BFGENCR

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not cleared 1b - Cleared
20 PPWF_CLR	Page Program Wait Flag Clear Selects the clearing mechanism for the page program wait flag (FR[PPWF]). See AHB command page program wait handling. 0b - Cleared by software 1b - Cleared by XSPI. No software intervention required.
19-18 —	Reserved
17 SEQID_WR_EN	Enable Write Sequence ID Enables the write sequence ID (SEQID_WR) as the index into the LUT for AHB write transactions. Otherwise, XSPI uses SEQID. 0b - Disables (use SEQID) 1b - Enables
16 —	Reserved
15-12 SEQID	LUT Sequence Pointer Specifies the sequence index of the LUT to be used for an AHB read transaction. If write sequence ID (BFGENCR[SEQID_WR_EN] = 0) is not enabled, it is also used for AHB write transactions. See Lookup table (LUT). **NOTE** If the sequence pointer differs in the new and the previous sequences, you should reset it. For more information, see SPTRCLR.
11 —	Reserved
10-9 AHBSSIZE	AHB Split Size Specifies split size when transaction split is enabled (SPLITEN = 1). See AHB transaction split. **NOTE** When you enable IPED GCM (write 1 to IPEDCTX*n*START[GCM]), you must specify 32 bytes or 64 bytes, and the split happens on 40 bytes or 80 bytes so that the header and authentication is included with every block. 00b - 8 bytes

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - 16 bytes 10b - 32 bytes 11b - 64 bytes
8 SPLITEN	AHB Transaction Split Enable Enables the splitting of all AHB read/write bursts into smaller transactions at the flash SPI bus side. You typically use this function for RAM devices with self-refresh limit or when using IPED GCM mode. See AHB transaction split. 0b - Disables 1b - Enables
7-6 —	Reserved
5 —	Reserved
4 WRTER_EN	Write Terminate Enable Specifies whether an AHB write transaction can terminate the ongoing AHB read-prefetch. 0b - Disables 1b - Enables
3-0 SEQID_RDSR	Read Status Register Sequence ID in LUT Specifies the sequence ID in the LUT to be used when reading the status register from flash in the event of a page program wait period. See AHB command page program wait handling.

41.7.2.10 SOC Configuration (SOCCR)

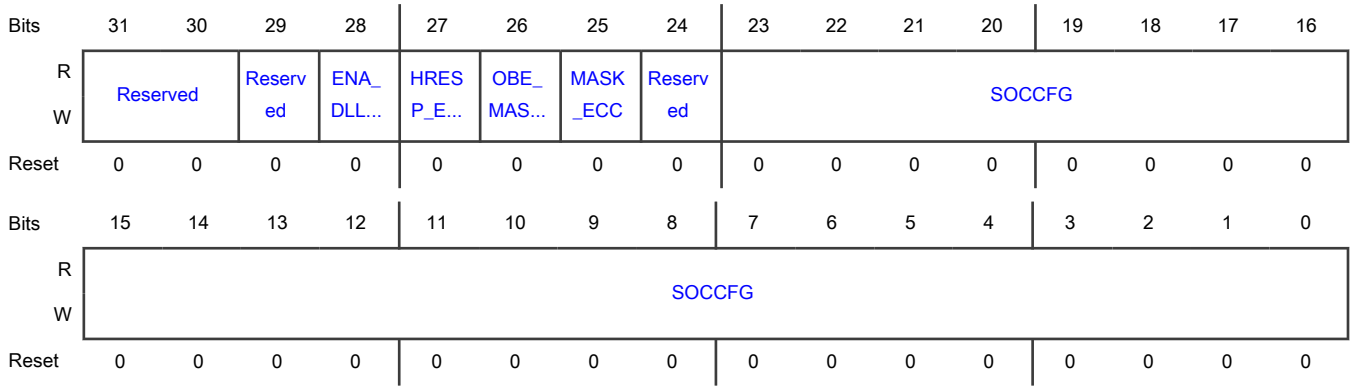
Offset

Register	Offset
SOCCR	24h

Function

Configures XSPI for chip-specific delay chain operation. See chip-specific XSPI information.

Diagram



Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 ENA_DLLABRT	DLL Abort Request Status Enable Unmasks the DLL Abort request status on soc_config I/F. 0b - Masks 1b - Unmasks
27 HRESP_ERR_MASK	HRESP Error Mask Masks the AHB HRESP error for some error scenarios. 0b - Allows HRESP error generation 1b - Masks HRESP error generation
26 OBE_MASK_EN	Output Buffer Enable Mask Enable Disable/Enable all OBE in output direction for x8 mode. 0b - Enable all OBE in output direction for x8 mode during write transaction (in a non-Dummy phase) 1b - Normal OBE handling for both read and write operation
25 MASK_ECC	Mask ECC for False AHB Reads Specifies the masking of ECC errors for predicted reads for INCR transfers. This signal can be used outside XSPI to gate ECC error generation in the event of false read scenarios. 0b - Unmask ECC error 1b - Mask ECC error

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 —	Reserved
23-0 SOCCFG	SOC Configuration Specifies chip-specific configuration of XSPI. See the chip-specific XSPI information.

41.7.2.11 Buffer n Top Index (BUF0IND - BUF2IND)

Offset

Register	Offset
BUF0IND	30h
BUF1IND	34h
BUF2IND	38h

Function

Specifies the size of the corresponding AHB buffer. The total size of the AHB buffers is 4 KB. Any left over space is automatically used for buffer 3. See Flexible read xAHB buffers.

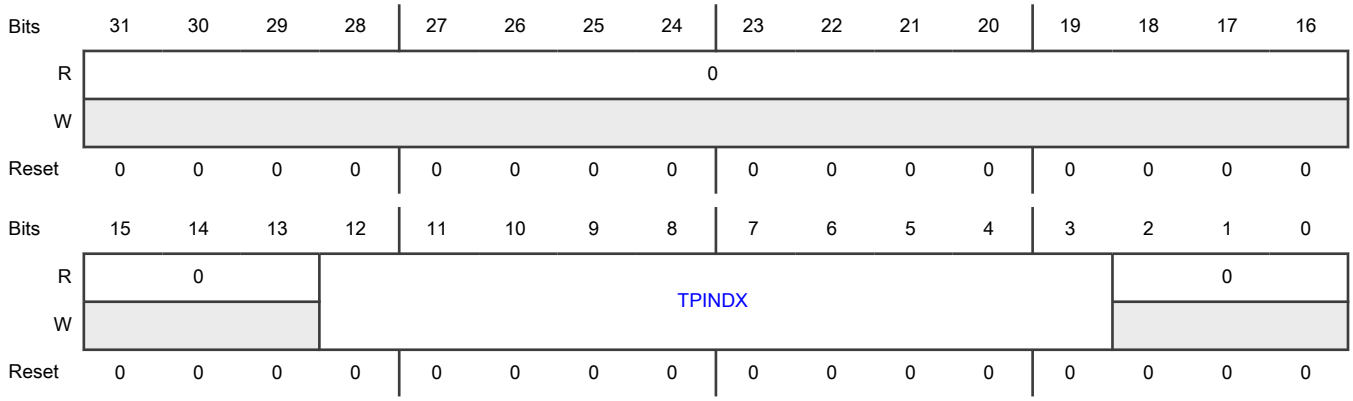
Special write-access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

NOTE

Because each buffer entry is 64-bits (8 bytes) long, the three least-significant bits of this register are 0 to maintain 64-bit alignment.

Diagram



Fields

Field	Function
31-13 —	Reserved
12-3 TPINDX	Top Index Buffer Specifies the top index of buffer 0 in multiples of 8 bytes. For example, a value of 0 equals 0 bytes in the buffer, 1 equals 8 bytes (1 buffer entry), 2 equals 16 bytes (2 buffer entries), and so on. The size of buffer 0 is the difference between BUF0IND and 0. The size of buffer 1 is the difference between BUF1IND and BUF0IND. For example, if BUF0IND[TPINDX] = 20h (BUF0IND = 100h), then writing 26h to BUF1IND[TPINDX] (BUF1IND = 130h) sets the size of buffer 1 to 30h bytes (6 buffer entries). The size of buffer 2 is the difference between BUF2IND and BUF1IND. For example, if BUF1IND[TPINDX] = 26h (BUF1IND = 130h), then writing 30h to BUF2IND[TPINDX] (BUF2IND = 180h) sets the size of buffer 2 to 50h bytes (10 buffer entries).
2-0 —	Reserved

41.7.2.12 AHB Write Configuration (AWRCR)

Offset

Register	Offset
AWRCR	50h

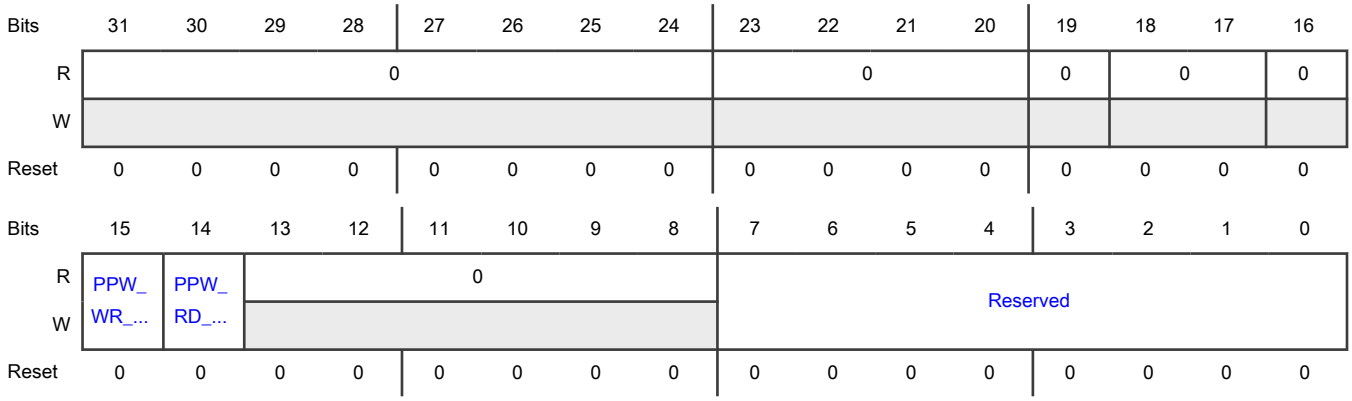
Function

Configures the allowed operations after an AHB write operation. See AHB command page program wait handling.

Special write-access is permitted if:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-24 —	Reserved
23-20 —	Reserved
19 —	Reserved
18-17 —	Reserved
16 —	Reserved
15 PPW_WR_DIS	Page Program Wait Write Disable Disables writes to the flash memory after the first write transaction by holding AHB HREADY low. 0b - Enables subsequent writes 1b - Disables subsequent writes
14 PPW_RD_DIS	Page Program Wait Read Disable Disables reads (in the event of a buffer miss) after the first write transaction to flash memory by holding AHB HREADY low. Disables reads is only permitted if: • AWRCR[PPW_WR_DIS] = 1 0b - Enables subsequent reads 1b - Disables subsequent reads

Table continues on the next page...

Table continued from the previous page...

Field	Function
13-8 —	Reserved
7-0 —	Reserved

41.7.2.13 DLL Flash Memory A Configuration (DLLCRA)

Offset

Register	Offset
DLLCRA	60h

Function

Configures DLL and the subordinate delay chain for flash memory A.

See DLL and delay chain usage for the programming sequence.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLLEN	FREQEN	0		DLL_REFCNTR				DLLRES				SLV_FINE_OFFSET			
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	SLV_DLY_OFFSET			SLV_DLY_COARSE				SLV_DLY_FINE			DLL_CDL8	SLAVE_A...	SLV_EN	SLV_DLL...	SLV_UPD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DLLEN	DLL Enable Enables the DLL logic. Write 1 to this field after you configure the DLL reference settings (FREQEN, DLL_REFCNTR, DLLRES, SLAVE_AUTO_UPDT). 0b - Disables. DLL reference logic remains in reset and should be 0 for at least three flash memory clock cycles for reset. 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 FREQEN	Frequency Enable Specifies the frequency of delay chain operations. 0b - Low frequency 1b - High frequency
29-28 —	Reserved
27-24 DLL_REFCNTR	DLL Reference Counter Specifies the "n + 1" interval for DLL phase detection and reference delay updating interval (minimum recommended value is 1).
23-20 DLLRES	DLL Resolution Specifies the minimum resolution for the DLL phase detector to remain locked or unlocked based on flash memory clock jitter. The minimum value is 2 but you should specify a more suitable value like 6.
19-16 SLV_FINE_OFFSET	Fine Offset Delay Elements Specifies the number of fine offset delay elements, up to 16, in the incoming DQS. You must specify a minimum of 1 element.
15 —	Reserved
14-12 SLV_DLY_OFFSET	T/16 Offset Delay Elements Specifies the number of T/16 offset delay elements in the incoming DQS. The default is 0.
11-8 SLV_DLY_COARSE	Delay Elements in Delay Tap Specifies the number of delay elements in each delay tap. This value overwrites the DLL-generated delay values and applies when SLV_DLL_BYPASS is 1. See XSPI datasheet for more details.
7-5 SLV_DLY_FINE	Fine Adjustment Delay Elements Specifies the number of fine adjustment delay elements in the total subordinate delay chain. This field is valid only when you enable subordinate DLL bypass (write 1 to SLV_DLL_BYPASS).
4 DLL_CDL8	DLL Course Delay Line 8 Enable Enables CDL8 and specifies the variation range. 0b - DLL is implemented to support up to 2x variation 1b - DLL is implemented to support up to 3x variation (BCS –> WCS)
3	Subordinate Chain Auto-Update Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
SLAVE_AUTO_UPDT	Enables automatic update of the subordinate chain upon DLL lock. 0b - Disables 1b - Enables
2 SLV_EN	Subordinate Enable Enables the DQS subordinate delay chain. You should write 1 to this field before configuring any subordinate settings. 0b - Disables. The DLL subordinate logic remains in reset, and its value should be 0 for at least three flash memory clock cycles for reset. 1b - Enables
1 SLV_DLL_BYPASS	Subordinate DLL Bypass Enables manual selection of the number of delays in each subordinate delay tap. You specify the number of delays in SLV_DLY_COARSE. 0b - Disables 1b - Enables
0 SLV_UPD	Subordinate Update Updates the DQS subordinate delay chain with either the reference delay or the bypass subordinate delay value. You should select this update in the absence of the DQS clock. Write to this field only after you configure the subordinate delay chain. 0b - Disables any further update on the DQS subordinate delay chain. 1b - Updates

41.7.2.14 Serial Flash Memory Address (SFAR)

Offset

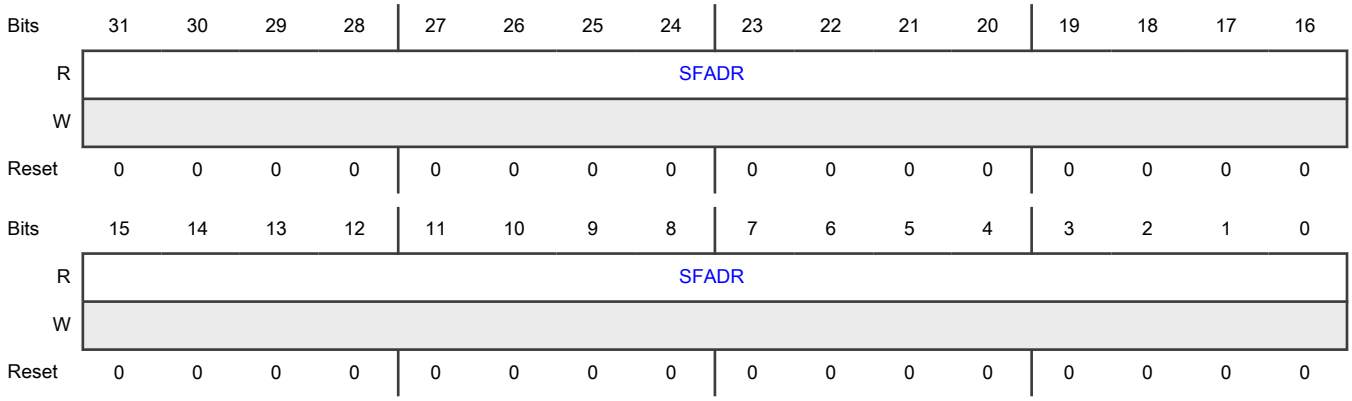
Register	Offset
SFAR	100h

Function

XSPI automatically translates this address in memory to the address in flash memory. When operating in 24-bit mode, only bits 23-0 are sent to the flash memory. In 32-bit mode, bits 27-0 are used with bits 31-28 driven to 0 when the value of SFACR[CAS] is 0. For example, if the value of SFACR[CAS] is 3, then bits 26-3 are sent to the flash memory as its page address in case flash memory is operating in a 24-bit mode. The total number of address bits requested by the flash memory, as its page and column address, must not be more than 32 bits.

XSPI updates this register when SFP triggers an IP-initiated command from any target group queue after it passes required checks and arbitration.

Diagram



Fields

Field	Function
31-0 SFADR	Serial Flash Memory Address Indicates the serial memory address for IP commands. The address is the address of the device without the base address.

41.7.2.15 Serial Flash Memory Address Configuration (SFACR)

Offset

Register	Offset
SFACR	104h

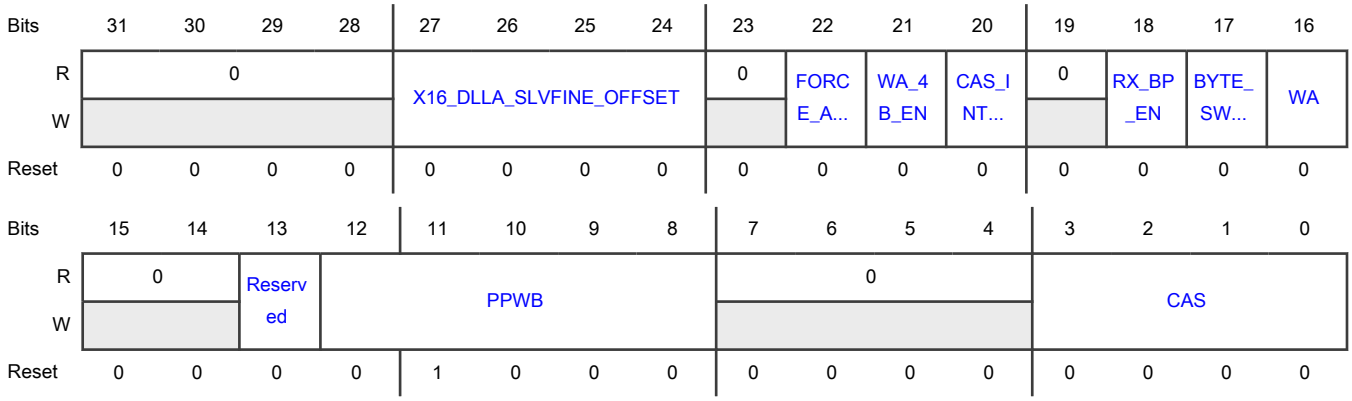
Function

Configures serial flash memory operation. You must program these values for the specific flash memory for XSPI to function correctly. XSPI automatically translates the address in the memory it controls (SFAR) or the incoming address on the AHB bus to the column address of the flash memory. For example, if a flash memory needs 3 bits for its column address, then XSPI sends only the lower three bits of the SFAR or AHB address to the flash memory as its column address. You must ensure that the address in SFAR or the incoming AHB address lies in the valid flash memory address range.

Special write-access is permitted if both:

- IP access is not asserted (SR[IP_ACC] = 0)
- AHB access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-28 —	Reserved
27-24 X16_DLLA_SLVFINE_OFFSET	Fine Offset Delay Elements in Incoming DQS1 Flash Memory A Specifies the number of fine offset delay elements (up to 16) in the incoming DQS1. The default is 1 element. **NOTE** This field is not supported in every instance. The following table includes only supported registers.
23 —	Reserved
22 FORCE_A10	10th Bit Status Specifies whether or not to force bit 10 of the address to 0 during address generation in X16 mode. 0b - Force 1b - Do not force
21 WA_4B_EN	32-Bit Address Mode Enable Enables the 32-bit address mode. Otherwise, XSPI uses the address mode specified in WA.

Instance	Field supported in	Field not supported in
XSPI0	—	SFACR
XSPI1	SFACR	—
XSPI2	SFACR	—

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables. Use WA-specified mode 1b - 32-bit mode
20 CAS_INTRLVD	CAS Interleaving Enables CAS interleaving. 0b - Disables 1b - Enables
19 —	Reserved
18 RX_BP_EN	RX Backpressure Enable Enables the RX backpressure handling mechanism. 0b - Disables 1b - Enables
17 BYTE_SWAP	Byte Swapping In Octal DDR mode, specifies whether a word unit composed of 2 bytes from the posedge and negedge of a single DQS cycle needs to be swapped. See Page program size. For example; if BYTE_SWAP is enabled and we write 4 bytes (byte 4, byte 3, byte 2, and byte 1) then write data is swapped as byte 3, byte 4, byte 1, and byte2 and written to memory device. 0b - One word of two bytes at addresses [n, n + 1] 1b - One word of two bytes at addresses [n + 1, n]
16 WA	Word Addressable Specifies whether the serial flash memory is a byte-addressable or word-addressable. XSPI uses this specification to remap the address to the flash memory interface. See Address schemes. 0b - Byte addressable 1b - Word (2-byte) addressable mode
15-14 —	Reserved
13 —	Reserved
12-8 PPWB	Page Program Boundary Specifies the flash memory-specific page program boundary size in log base 2 format. For example, to specify a 512-byte page program size, log2(512) = 9. The default value is 8 for a 256-byte page program size.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-4 —	Reserved
3-0 CAS	Column Address Space Specifies the width of the column address. For example, if the column address is contained in bits 2–0 of SFAR or the AHB address, then CAS must be 3. If there is no column address separation in any serial flash memory, you must specify this value as 0. See Address schemes.

41.7.2.16 Sampling (SMPR)

Offset

Register	Offset
SMPR	108h

Function

Configures how XSPI samples the incoming data from external serial flash memory devices.

NOTE

See the chip data sheet for how to configure sampling.

Special write access is permitted if XSPI is disabled (MCR[MDIS] = 0).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	Reserved			Reserved	DLLFSMPFA			0				Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FSDLY	FSPHS	0		0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
30-28 —	Reserved
27 —	Reserved
26-24 DLLFSMPFA	DLL Nth Tap Flash Memory A Selects the nth tap provided by a subordinate delay chain for flash memory A. The value of *n* can vary from 0 to 7, with each tap delay based on the DLL configuration (DLLCR*x*).
23-19 —	Reserved
18-16 —	Reserved
15-7 —	Reserved
6 FSDLY	Full-Speed Delay For Internal/Pad Loopback Selects the delay for full-speed commands. Select the delay with respect to the reference edge for the sample point valid for full-speed commands. **NOTE** See the chip-specific XSPI information for the read modes supported by this chip. This field is valid only for MCR[DQS_FA_SEL] SDR sampling. You must write 0 in other sampling modes. 0b - Same DQS 1b - Half cycle early DQS
5 FSPHS	Full-Speed Phase Selects the edge of the sampling clock valid for full-speed commands. **NOTE** See the chip-specific XSPI information for the read modes supported by this chip. This field is valid only for internal non-DQS SDR sampling. You must write 0 in other sampling modes. 0b - Sampling at non-inverted clock 1b - Sampling at inverted clock
4-3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2-0 —	Reserved

41.7.2.17 RX Buffer Status (RBSR)

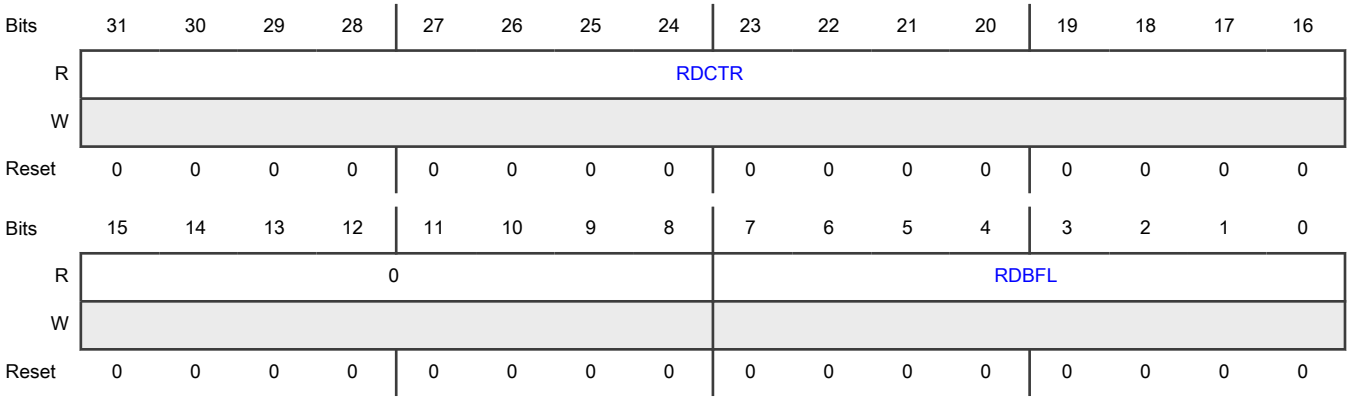
Offset

Register	Offset
RBSR	10Ch

Function

Indicates the status of the receive data buffer.

Diagram



Fields

Field	Function
31-16 RDCTR	Read Counter Indicates the number of 4-byte entries removed from the RX buffer. For example, a value of 2 indicates that 8 bytes have been removed. XPSI increments RDCTR by the value of RBCT[WMRK] + 1 for each RX buffer pop event. You can pop the RX buffer using DMA or by writing 1 to FR[RBDF]; you specify which method to use in RSER[RBDDE]. See RX Buffer Data (RBDR0 - RBDR63) and Data transfer from the XSPI module internal buffers.
15-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 RDBFL	RX Buffer Fill Level Indicates the number of 4-byte entries available in the RX buffer. For example, a value of 2 indicates 8 bytes are available.

41.7.2.18 RX Buffer Control (RBCT)

Offset

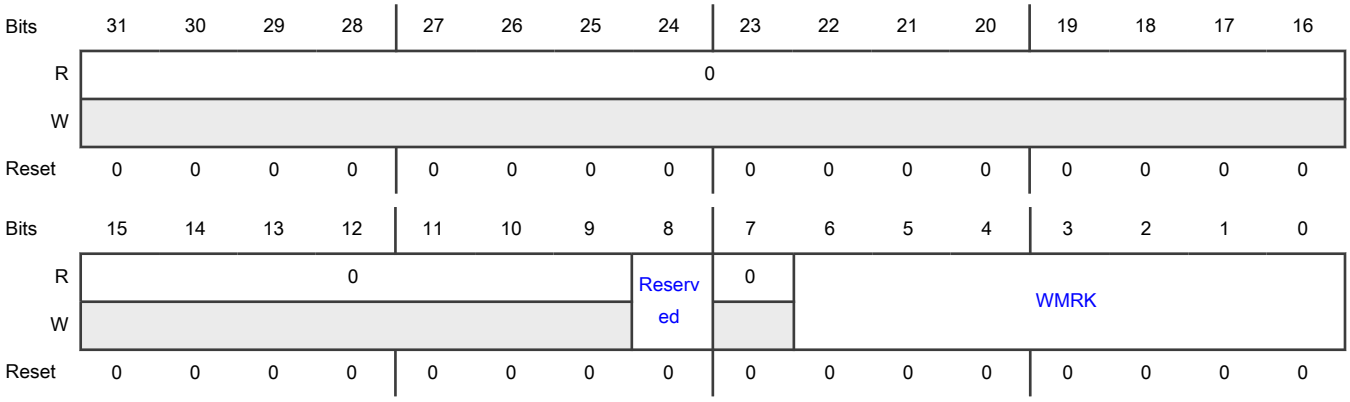
Register	Offset
RBCT	110h

Function

Controls the RX data buffer.

Special write access is permitted if IP access is not asserted (SR[IP_ACC] = 0).

Diagram



Fields

Field	Function
31-9 —	Reserved
8 —	Reserved
7 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
6-0 WMRK	RX Buffer Watermark Specifies the number of valid 4-byte entries minus one in the RX buffer which causes XSPI to assert the watermark exceeded flag (SR[RXWE]). For example, a value of 0 sets the watermark to 4 bytes, 1 to 8 bytes, 2 to 12 bytes, and so on. See RX buffer POP event. **NOTE** Because there are only 64 memory-mapped RX data buffer (RBDR*n*) registers, do not write a value greater than 63 to this field. If you do, you will lose data above 64 words.

41.7.2.19 AHB Write Status (AWRSR)

Offset

Register	Offset
AWRSR	120h

Function

Indicates sequence auto join.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													SEQAUJO...	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-3 —	Reserved
2	Sequence Auto Join

Table continues on the next page...

Table continued from the previous page...

Field	Function
SEQAUJOIN	Indicates that the LUT sequences are automatically joined as a result of the JMP_TO_SEQ command. This field remains asserted as long as you do not encounter the STOP/JMP_ON_CS command. 0b - Not auto-joined 1b - Auto-joined
1 —	Reserved
0 —	Reserved

41.7.2.20 DLL Status (DLLSR)

Offset

Register	Offset
DLLSR	12Ch

Function

Indicates DLL operation status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				0				Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLLA_LO...	SLVA_LO...	DLLA_RA...	DLLA_FI...	0				DLLA_SLV_FINE_VAL				DLLA_SLV_COARSE_VAL			
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-24 —	Reserved
23-16 —	Reserved
15 DLLA_LOCK	DLL A Lock Status Indicates that the DDL for flash memory A is locked. 0b - Not locked 1b - Locked
14 SLVA_LOCK	Subordinate High Lock Status - Flash Memory A Indicates that the high-frequency subordinate delay chain is locked. The field is configured when the subordinate decoder is updated with DLLCRA[SLV_UPD] and is reset after you configure SLV_UPD to 0. 0b - Not locked 1b - Locked
13 DLLA_RANGE_ERR	DLL Initiator Delay Chain Range Error - Flash Memory A Indicates that the DLL initiator delay chain for flash memory A is working out of delay range because of incorrect DLL configuration. This is a DLL characterization error and it creates a functional issue related to DLL training. DLL cells need to be re-evaluated with respect to clock period. 0b - In range 1b - Out of range
12 DLLA_FINE_UNDERFLOW	Fine Delay Chain Underflow - Flash Memory A Indicates that fine delay chain underflow has occurred for flash memory A. 0b - No overflow 1b - Overflow
11-8 —	Reserved
7-4 DLLA_SLV_FINE_VAL	Fine Delay Cells in Subordinate Delay Chain Indicates the total number of fine delay cells (1 delay unit) selected in the subordinate delay chain for flash memory A.
3-0 DLLA_SLV_COARSE_VAL	Coarse Delay Cells in Subordinate Delay Chain - Flash Memory A Indicates the total number of coarse delay cells (16 delay units) selected in the subordinate delay chain for flash memory A.

41.7.2.21 Data Learning Configuration (DLCR)

Offset

Register	Offset
DLCR	130h

Function

Configures data learning operation.

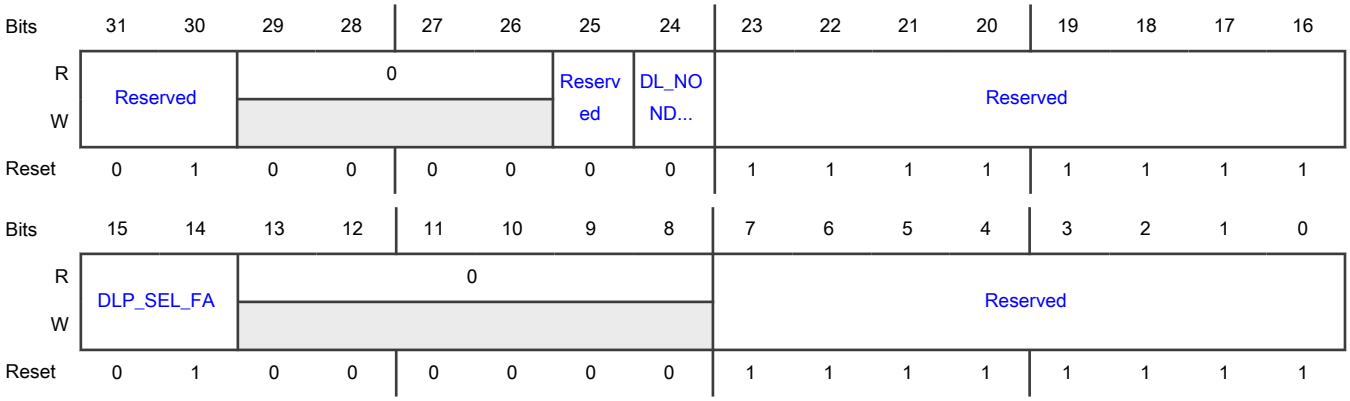
To support data learning operation, a flash memory device must provide more than 10 bits of data learn pattern; otherwise, XSPI considers it to be non-DLP Flash. For non-DLP flash devices, you must read at least 16 bits of data learn pattern from a known location. See Data learning.

To accommodate pessimistic data skew between the different IO lines, you must program a different learning pattern on each of the data pins IO1 and IO3.

Special write-access is permitted if both:

- IP access is not asserted (SR[IP_ACC] = 0)
- AHB access is not asserted (SR[AHB_ACC] = 0)

Diagram



Fields

Field	Function
31-30 —	Reserved
29-26 —	Reserved
25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 DL_NONDLP_FLSH	DLP Non-DLP Flash Memory Enable Enables data learning for flash memories that do not provide a data learning pattern. For a non-DLP flash memory, execute a one-time data learning through the IPS to select a tap before an AHB read operation. 0b - Disables 1b - Enables
23-16 —	Reserved
15-14 DLP_SEL_FA	DLP Selection Flash Memory A Selects which pad to use for pattern matching IO for flash memory A. 00b - Ignore pattern matching. Use this value only for debugging. 01b - IO1 10b - IO3. Use this value only for debugging. 11b - Both IO1 and IO3
13-8 —	Reserved
7-0 —	Reserved

41.7.2.22 Data Learning Status Flash Memory (DLSR_FA)

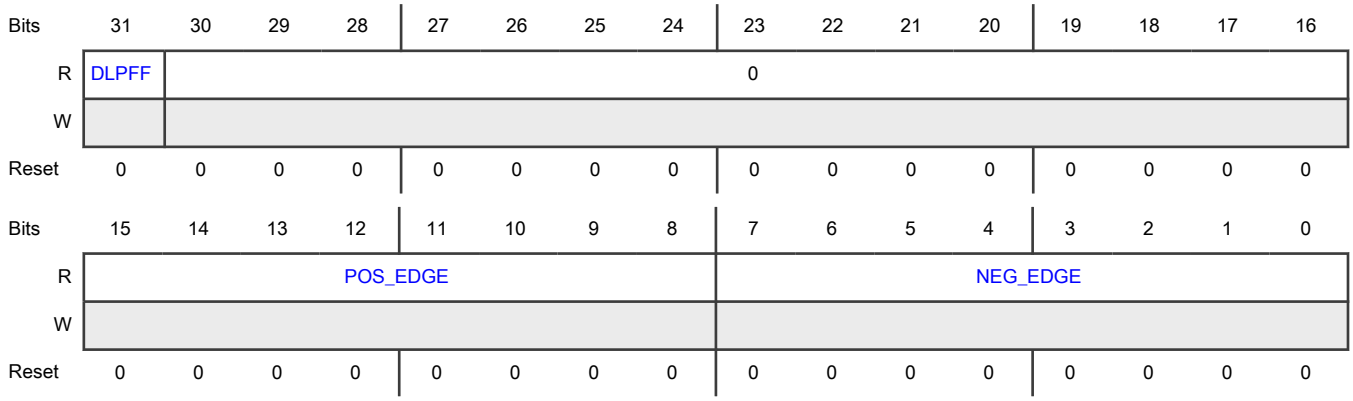
Offset

Register	Offset
DLSR_FA	134h

Function

Indicates the sampling point selected by the data learning algorithm for the corresponding flash memory when data matching is successful (DLPFF = 0). Otherwise, indicates the pattern matching outline.

Diagram



Fields

Field	Function
31 DLPFF	DLP Failure Flag Indicates that data learning has failed. 0b - No failure 1b - Failure
30-16 —	Reserved
15-8 POS_EDGE	DLP Positive Edge Match Indicates the DLP positive edge match signature, which contains the tap number selection (in between 0-7), based on auto data learning scheme.
7-0 NEG_EDGE	DLP Negative Edge Match Indicates the DLP negative edge match signature, which contains the tap number selection (in between 0-7), based on auto data learning scheme.

41.7.2.23 TX Buffer Status (TBSR)

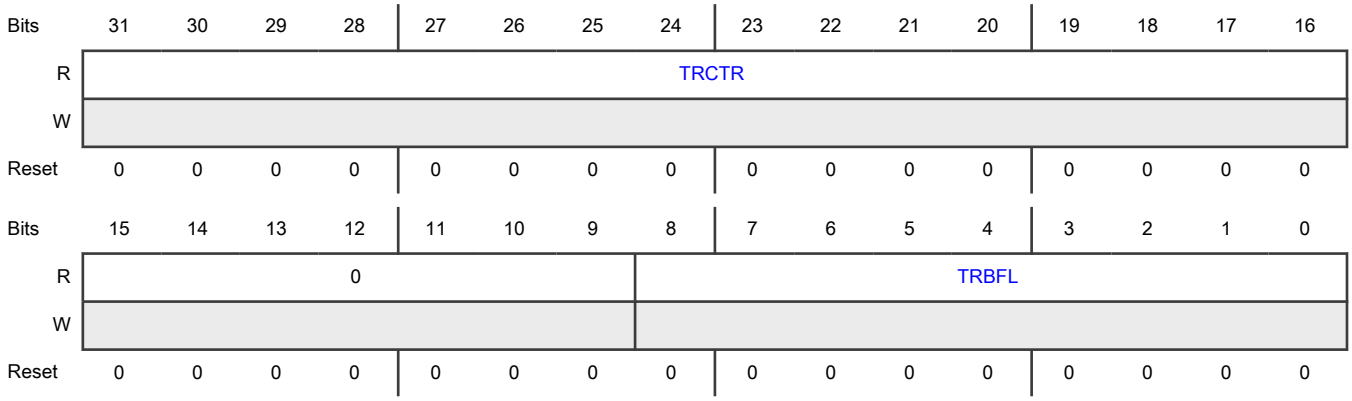
Offset

Register	Offset
TBSR	150h

Function

Indicates the status of the TX data buffer.

Diagram



Fields

Field	Function
31-16 TRCTR	Transmit Counter Indicates how many 4-byte entries have been written to the TX buffer by host accesses. XSPI increments this count on each write to TBDR that results in a push to the TX buffer. To reset this count to 0, write 1 to MCR[CLR_TXF]. If you do not reset the count, it wraps from FFFFh to 0.
15-9 —	Reserved
8-0 TRBFL	TX Buffer Fill Level Indicates the number of 4-byte entries written to the TX buffer for transmitting to the serial flash memory device.

41.7.2.24 TX Buffer Data (TBDR)

Offset

Register	Offset
TBDR	154h

Function

Writes data to the circular TX buffer. The data in the buffer is in turn written to the serial flash memory by the page programming commands.

See Byte ordering – endianness for the byte ordering scheme. A write transaction on the flash memory with data size of less than 32 bits leads to the removal of one data entry from the TX buffer. The valid bits are used and the rest of the bits are discarded.

Special write-access is permitted if the TX buffer is not full (TXFULL = 0).

NOTE

You can write to this register only when write access is granted by the SFP; otherwise, attempting to write to it generates a bus transfer error. See TBDR register write lock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TXDATA	TX Data On write, writes the data to the next available entry of the TX buffer. XSPI updates the buffer fill level (TBSR[TRBFL]). On read, returns the last data written to the register.

41.7.2.25 TX Buffer Control (TBCT)

Offset

Register	Offset
TBCT	158h

Function

Controls operation of the TX data buffer.

Special write-access is permitted if:

- IP access is not asserted (SR[IP_ACC] = 0)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								WMRK							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 WMRK	TX Buffer Watermark Specifies the minimum number of available 4-byte entries in the TX buffer before XSPI deasserts TX watermark available (writes 0 to SR[TXWA]). You must specify a value greater than 0. See TX buffer operation and DMA.

41.7.2.26 Status (SR)

Offset

Register	Offset
SR	15Ch

Function

Indicates the status of SFM command execution and arbitration, as well as the RX, TX, and AHB buffers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0		TXFULL	TXDMA	TXWA	TXNE	RXDMA	ARB_STATE			RXFULL	0	0	RXWE
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARB_LCK	AHB3FUL	AHB2FUL	AHB1FUL	AHB0FUL	AHB3NE	AHB2NE	AHB1NE	AHB0NE	AHBTRN	Reserved	AWRACC	Reserved	AHB_ACC	IP_ACC	BUSY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
29-28 —	Reserved
27 TXFULL	TX Buffer Full Indicates that the TX buffer has the maximum allowed number (256) of 4-byte entries. 0b - Not full 1b - Full
26 TXDMA	TX DMA Indicates that TX buffer fill by DMA is enabled (RSER[TBFDE] = 1) and XSPI has received or is processing a DMA request. 0b - Disabled or not processing the DMA request 1b - Enabled and processing DMA request
25 TXWA	TX Buffer Watermark Available Indicates when the number of available spaces in the TX buffer is greater than or equal to the specified watermark (TBCT[WMRK]). For example, when TBCT[WMRK] = 1, XSPI writes 0 to this field when the TX buffer is full (256 entries). **NOTE** • In case of IPED transaction, once this flag is de-asserted, it will remain low unless application clears buffer pointers after winning arbitration using environment-specific MCR[CLR_TXF]. • In case of timeout (MTO) error, only Main environment-specific MCR[CLR_TXF] is allowed to reset it independent of having arbitration. • In case of arbitration lock timeout, wait for transaction to complete and then release arbitration lock ERRSTAT[ARB_LOCK_TO] by W1C. TX Buffer pointers will retain its previous value and will be cleared using environment-specific MCR[CLR_TXF] after winning arbitration for a new request. 0b - Not available 1b - Available
24 TXNE	TX Buffer Not Empty Indicates when the TX buffer: • Contains data ready for transfer. • Data has been left in the buffer after early termination of a write transaction, which includes the data of the first transfer of an interrupted burst activity when HSIZE is 1 (1 byte) or 2 (2 bytes). 0b - Empty 1b - Not empty
23	RX Buffer DMA

Table continues on the next page...

Table continued from the previous page...

Field	Function
RXDMA	Indicates when RX buffer read out via DMA is active; that is, when DMA is requested or running. 0b - Not active 1b - Active
22-20 ARB_STATE	XSPI Arbitration State Indicates the state of arbitration. 000b - Idle 001b - ARB - Arbitration 010b - AHBWR_PROC - AHB write processing 011b - AHBRD_PROC - AHB read processing 101b - TG_PROC - IP command processing 110b - PROC_FIN - XSPI processing finish wait 111b - RDSR_PROC - RDSR read processing All other values are reserved.
19 RXFULL	RX Buffer Full Indicates that the RX buffer is full; that is, RBSR[RDBFL] equals 128. 0b - Not full 1b - Full
18 —	Reserved
17 —	Reserved
16 RXWE	RX Buffer Watermark Exceeded Indicates that the number of valid entries in the RX buffer exceeds the watermark value (RBCT[WMRK]). 0b - Not exceeded 1b - Exceeded
15 ARB_LCK	Arbitration Lock Status Indicates that arbitration is locked by the AHB page program wait flag (FR[PPWF]) or by the IPS TG queue. 0b - Not locked 1b - Locked
14-11	AHB Buffer n Full

Table continues on the next page...

Table continued from the previous page...

Field	Function
AHBnFUL	Indicates when the corresponding AHB buffer is full. If you enable sub-buffer division (write 1 to BUF*n*CR[SUB_DIV_EN]), indicates that at least one sub-buffer is full. 0b - Not full 1b - Full
10-7 AHBnNE	AHB Buffer n Not Empty Indicates when the corresponding AHB buffer contains data. 0b - Empty 1b - Not empty
6 AHBTRN	AHB Access Transaction Pending Indicates when there is a pending request on the AHB interface. See Flash memory mapped AMBA bus. 0b - No pending request 1b - Request is pending
5 —	Reserved
4 AWRACC	AHB Write Access Indicates when an AHB write access has been requested or is ongoing. 0b - No AHB write access 1b - AHB write access
3 —	Reserved
2 AHB_ACC	AHB Read Access Indicates when an AHB read access has been requested or is ongoing. 0b - Not initiated by AHB bus 1b - Initiated by AHB bus
1 IP_ACC	IP Access Indicates when the currently executing transaction has been initiated by the IP bus. 0b - Not initiated by IP bus 1b - Initiated by IP bus
0 BUSY	Module Busy Indicates when the module is currently busy handling a transaction to an external flash memory device 0b - Not busy

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Busy

41.7.2.27 Flag (FR)

Offset

Register	Offset
FR	160h

Function

Provides all available flags about SFM command execution and arbitration that can serve as sources for the generation of interrupt service requests. The error flags in this register do not relate directly to the execution of the transaction in the serial flash memory device itself but only to the behavior and conditions visible in the XSPI module.

Special write access is permitted in normal mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPFF	0	Reserved	DLLABRT	TBFF	TBUF	0	DLLUNLCK	ILLINE	0	0	0	0		RBOF	RBDF
W	W1C			W1C	W1C	W1C		W1C	W1C						W1C	W1C
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AAEF	AITEF	AIBSEF	ABOF	Reserved	0	0	PPWF	0	IPIEF	IPEDERR	0	0	PERFOVF	RDADDR	TFF
W	W1C	W1C	W1C	W1C	W1C			W1C		W1C	W1C			W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DLPFF	Data Learning Pattern Failure Flag Indicates that XSPI encountered a DATA_LEARN instruction in a sequence, but no sampling point is found for the data learning pattern. While this flag is set, XSPI returns an AHB bus error for all AHB read transactions. If HRESP_ERR_MASK is 1, then XSPI does not generate the AHB HRESP error. Instead, it returns the AHB error payload for the current AHB burst (AHB_ERR_PAYLOAD_LO, AHB_ERR_PAYLOAD_HI). It then treats the next AHB read burst as a normal AHB read and processes it without regard for the clearing of this flag.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No failure 1b - Failure When writing 0b - No action 1b - Clears flag
30 —	Reserved
29 —	Reserved
28 DLLABRT	DLL Terminate Indicates that the DLL has unlocked while reading data from the flash memory. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No lock has occurred 1b - DLL unlock occurred When writing 0b - No action 1b - Clears flag
27 TBFF	TX Buffer Fill Flag Indicates that the TX buffer is filling (not yet full). Before writing to the TX buffer, clear this field and then read it. If it is set, the TX buffer has room for more data. If the field remains cleared, the TX buffer is full. See TX buffer operation. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No room in the TX buffer 1b - TX buffer has room When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No action 1b - Clears flag
26 TBUF	TX Buffer Underrun Flag Indicates that XSPI has attempted to pull data when the TX buffer is empty during IPS write transaction. When a TX-buffer underrun occurs (TBUF = 1), the memory interface generates a sequence termination event. In response to an underrun, you must clear the TX buffer (write 1 to CLR_TXF) and re-execute the entire transaction. XSPI does not service pending AHB write transactions (HREADY low) until you clear this flag. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underrun 1b - Underrun When writing 0b - No action 1b - Clears flag
25 —	Reserved
24 DLLUNLCK	DLL Unlock Indicates the occurrence of a DLL unlock event, regardless of flash memory access. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No unlock event 1b - Unlock event has occurred When writing 0b - No action 1b - Clears the flag
23 ILLINE	Illegal Instruction Error Flag Indicates that XSPI encountered an illegal instruction in any sequence. When this error occurs, you must reconfigure the correct sequence instruction and assert software reset for both the serial flash memory and AHB domains (write 1 to both MCR[SWRSTSD]] and MCR[SWRSTHD]). For a list of legal instructions, see Programmable sequence engine.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears error flag
22 —	Reserved
21 —	Reserved
20 —	Reserved
19-18 —	Reserved
17 RBOF	RX Buffer Overflow Flag Indicates when the RX buffer cannot receive any more data from the serial flash memory device. XSPI continues processing the IP command that triggered the overflow until it reads the number of bytes specified by the IP data size (IPCR[IDATSZ]) from the serial flash memory device. The content of the RX buffer remains unchanged. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No action 1b - Clears flag
16 RBDF	RX Buffer Drain Flag Indicates the RX buffer watermark has been exceeded (SR[RXWE] = 1). If RX buffer drain DMA is enabled (RSER[RBDDE] = 1), XSPI triggers a POP.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If RX buffer drain DMA is not enabled (RSER[RBDDE] = 0), XSPI does not trigger an RX buffer POP event. After the POP event, when you write 1 to this field: • If the RX buffer has less than or up to RBCT[WMRK] valid entries, XSPI clears the flag. • If the RX buffer has more than RBCT[WMRK] valid entries after the POP, XSPI does not clear the flag. For more information, see the "Receive Buffer Drain Interrupt or DMA Request" section in Normal mode interrupt and DMA requests. **NOTE** This field behaves differently for register reads and writes. When reading 0b - RX buffer is not over the watermark 1b - RX buffer is over the watermark When writing 0b - No action 1b - Clears flag if RX buffer is not over the watermark after POP
15 AAEF	AHB Abort Error Flag Indicates that an AHB transaction is in process and any of the following conditions occur that may result in an AHB error response (HRESP): • Data learning pattern failure (FR[DLPFF]) • AHB illegal burst size error during the AHB transfer (FR[AIBSEF]) You should clear this flag and any of these related conditions before initiating any new AHB transfer. If SOCCR[MASK_ECC] is 1, then XSPI does not generate the AHB HRESP error. Instead, it returns the AHB error payload for the current AHB burst. It then treats the next AHB read burst as a normal AHB read and processes it without regard for the clearing of this flag. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears error flag
14 AITEF	AHB Illegal Transaction Error Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates that XSPI has not generated a response to the AHB bus due to an illegal transaction and expiration of the watchdog timer. You specify the timer value in AHB HREADY Timeout (AHRDYTO). For an AHB read transaction, if AHB timeout occurs then XSPI returns the AHB error payload as read data and does not generate an HRESP error. If this flag is set, XSPI returns an HRESP error for every AHB write transaction. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears error flag
13 AIBSEF	AHB Illegal Burst Size Error Flag Indicates that the total burst size (size x beat) of an AHB transaction is greater than the prefetch data size, as specified by ADATSZ or, if ADATSZ is 1, the data size specified in the sequence pointed to by SEQID. For information about HBURST, see HBURST support with AHB read. Until you clear this flag, XSPI returns AHB read transfers with the HRESP error. If HRESP_ERR_MASK is 1, then XSPI does not generate the AHB HRESP error. Instead, it returns the AHB error payload for the current AHB burst (AHB_ERR_PAYLOAD_LO, AHB_ERR_PAYLOAD_HI). It then treats the next AHB read burst as a normal AHB read and processes it without regard for the clearing of this flag. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears error flag
12 ABOF	AHB Buffer Overflow Flag Indicates when the size of the AHB access exceeds the size of the AHB buffer. This condition can occur only if BUF*n*CR[ADATSZ] has the incorrect value. XSPI continues processing the AHB command that triggers this condition until the number of entries specified in BUFxCR[ADATSZ] have been read from the serial flash memory device. The content of the AHB buffer is changed.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No action 1b - Clears overflow flag
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 PPWF	Page-Program Wait Flag Indicates assertion of the page-program wait flag after writing to flash memory. When asserted, this flag blocks all SFP or IP transactions. If you disable page program wait writes (write 1 to AWRCR[PPW_WR_DIS]), it also blocks AHB writes. If you disable page program wait reads (write 1 to AWRCR[PPW_RD_DIS]), it also blocks AHB reads. See Page program wait handling. If any AHB transfer is kept waiting by HREADY being held low when this flag is set, it may cause assertion of AITEF. On assertion of PPWF, XSPI starts the SFP arbitration lock timeout counter (SFP_ARB_TIMEOUT). If the counter expires, it triggers an arbitration lock timeout error (writes 1 to ERRSTAT[ARB_LOCK_TO]). If ERRSTAT[ARB_LOCK_TO] is set, you must clear it before you clear PPWF. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No page program wait 1b - Page program wait in effect When writing 0b - No action 1b - Clears flag
7	Reserved

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>6
IPIEF</td><td>IP Command Trigger Fail Error Flag
If XSPI is currently executing an IP-triggered command (SR[IP_ACC] = 1), indicates when the following condition occurs:
• Write access to RBCT
XSPI ignores any command that causes assertion of the IPIEF flag.
NOTE
This field behaves differently for register reads and writes.
When reading
0b - No error
1b - Error
When writing
0b - No action
1b - Clears error flag</td></tr>
<tr><td>5
IPEDERR</td><td>IPED RX Decryption Error Flag
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>XSPI0</td><td>FR</td><td>—</td></tr><tr><td>XSPI1</td><td>FR</td><td>—</td></tr><tr><td>XSPI2</td><td>—</td><td>FR</td></tr></table>NOTE
This field behaves differently for register reads and writes.
When reading
0b - No error
1b - Error
When writing
0b - No action
1b - Clears error flag</td></tr>
<tr><td>4</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
3 —	Reserved
2 PERFOVF	AHB Performance Monitor Overflow Flag Indicates when AHB_PERF_CTRL reports any buffer hit or miss counter overflows. The overflow bits in AHB_PERF_CTRL remain asserted until you clear this flag by writing 1 to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No action 1b - Clears flags
1 RDADDR	AHB Read Address Error Flag When an AHB buffer is divided into four sub-buffers, indicates one of the following: • The initiator sent an AHB read address to that buffer that is not within the address range of any of its sub-buffers. • The initiator sent an AHB read address to that buffer that is within multiple sub-buffer address ranges because of incorrect programming. The offending transaction causes XSPI to return the AHB error payload (AHB_ERR_PAYLOAD_LO, AHB_ERR_PAYLOAD_HI). See Buffer function with sub-buffer division enabled. This flag is only applicable for a legal AHB read. Clear this flag by writing 1 to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 TFF	IP Command Transaction Finished Flag Indicates that XSPI has completed a running IP command. If an error occurs, and the related error flags are valid in the same clock cycle, XSPI asserts TFF. Clear this flag by writing 1 to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not completed 1b - Completed When writing 0b - No action 1b - Clears flag

41.7.2.28 Interrupt and DMA Request Enable (RSER)

Offset

Register	Offset
RSER	164h

Function

Enables the XSPI interrupts and DMA requests.

NOTE

Every enabled interrupt or DMA request prevents the XSPI module from entering stop mode or module disable mode when the flag for that source is set in FR. See Stop mode and Module Disable mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPFIE	Reserved		0	TBFIE	TBUIE	TBFDE	DLLULIE	ILLINIE	Reserved	RBDDE	Reserved	0	0	RBOIE	RBDIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AAIE	AITIE	AIBSIE	ABOIE	Reserved	Reserved	Reserved	PPWIE	0	IPIEIE	IPEDIE	Reserved	0	PERFOVIE	RDADDRIE	TFIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DLPFIE	Data Learning Pattern Failure Interrupt Enable Enables generation of an interrupt on data learning pattern failure. See FR[DLPFF]. 0b - Disables 1b - Enables
30-29 —	Reserved
28 —	Reserved
27 TBFIE	TX Buffer Fill Interrupt Enable Enables generation of an interrupt on TX buffer fill. See FR[TBFF]. Do not enable this interrupt when using SFP functionality because it causes XSPI BUSY SFM to go high and blocks SFP from generating any transaction over to XSPI. 0b - Disables 1b - Enables
26 TBUIE	TX Buffer Underrun Interrupt Enable Enables generation of an interrupt on TX buffer underrun. See FR[TBUF]. 0b - Disables 1b - Enables
25 TBFDE	TX Buffer Fill DMA Enable Enables generation of a DMA request for TX buffer fill. When enabled, XSPI generates DMA requests as long as the available space in the TX buffer is equal to or greater than the TX watermark (TBCT[WMRK]). 0b - Disables 1b - Enables
24 DLLULIE	DLL unlock interrupt enable Enables generation of an interrupt on a DLL unlock event. See FR[DLLUNLCK]. 0b - Disables 1b - Enables
23 ILLINIE	Illegal Instruction Error Interrupt Enable Enables generation of an interrupt on occurrence of an illegal instruction error. See FR[ILLINE]. 0b - Disables 1b - Enables
22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
21 RBDDE	RX Buffer Drain DMA Enable Enables generation of a DMA request for RX buffer drain. When enabled, XSPI generates a DMA request when the RX buffer exceeds the watermark (SR[RXWE]). **NOTE** After you enable DMA transfer requests (write 1 to this field). To disable DMA transfers, you must perform a software reset for the AHB domain (write 1 to SWRSTHD). 0b - Disabled 1b - Enables
20 —	Reserved
19 —	Reserved
18 —	Reserved
17 RBOIE	RX Buffer Overflow Interrupt Enable Enables generation of an interrupt on occurrence of RX buffer overflow. See FR[RBOF]. This field indicates the RX buffer overflow interrupt enable flag. 0b - Disables 1b - Enables
16 RBDIE	RX Buffer Drain Interrupt Enable Enables generation of an interrupt on occurrence of RX buffer drain. See FR[RBDF]. 0b - Disables 1b - Enables
15 AAIE	AHB Abort Error Interrupt Enable Enables generation of an interrupt on occurrence of an AHB abort error. See FR[AAEF]. 0b - Disables 1b - Enables
14 AITIE	AHB Illegal Transaction Error Interrupt Enable flag Enables generation of an interrupt on occurrence of an AHB illegal transaction error. See FR[AITEF]. This field indicates the AHB illegal transaction interrupt enable flag.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
13 AIBSIE	AHB Illegal Burst Size Error Interrupt Enable Enables generation of an interrupt on occurrence of an AHB illegal burst size error. See FR[AIBSEF]. 0b - Disables 1b - Enables
12 ABOIE	AHB Buffer Overflow Interrupt Enable Enables generation of an interrupt on occurrence of an AHB buffer overflow. See FR[ABOF]. 0b - Disables 1b - Enables
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 PPWIE	Page-Program Wait Interrupt Enable Enables generation of an interrupt on page-program wait. See FR[PPWF]. 0b - Disables 1b - Enables
7 —	Reserved
6 IPIEIE	IP Command Trigger Fail Error Interrupt Enable Enables generation of an interrupt on occurrence of an IP command trigger fail error. See FR[IPIEF]. 0b - Disables 1b - Enables
5 IPEDIE	IPED RX Decryption Error Interrupt Enable Enables generation of an interrupt on occurrence of an IPED RX decryption error. See FR[IPEDERR]. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>XSPI0</td><td>RSER</td><td>—</td></tr><tr><td>XSPI1</td><td>RSER</td><td>—</td></tr><tr><td>XSPI2</td><td>—</td><td>RSER</td></tr></table> 0b - Disables 1b - Enables
4 —	Reserved
3 —	Reserved
2 PERFOVIE	AHB Performance Monitor Overflow Interrupt Enable Enables generation of an interrupt on a buffer hit or miss counter overflow. See FR[PERFOVF]. 0b - Disables 1b - Enables
1 RDADDRIE	AHB Read Address Error Interrupt Enable Enables generation of an interrupt on occurrence of an AHB read address error. See FR[RDADDR]. 0b - Disables 1b - Enables
0 TFIE	Transaction Finished Interrupt Enable Enables generation of an interrupt when an IP command transaction completes. See FR[TFF]. 0b - Disables 1b - Enables

41.7.2.29 Sequence Suspend Status (SPNDST)

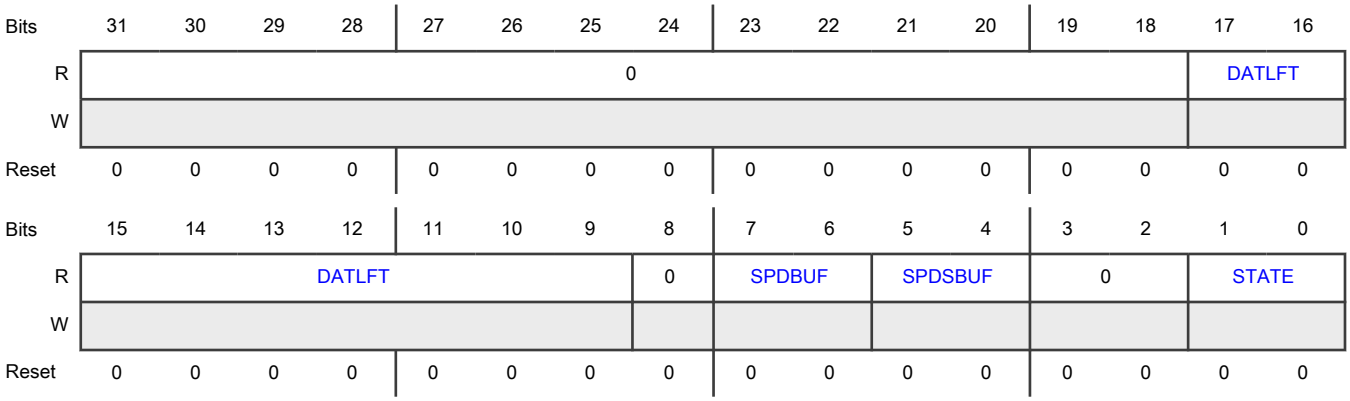
Offset

Register	Offset
SPNDST	168h

Function

Provides information specific to any suspended sequence. An AHB sequence may be suspended when a high priority AHB initiator makes an access before the AHB sequence completes the requested data transfer.

Diagram



Fields

Field	Function
31-18 —	Reserved
17-9 DATLFT	Data Left Indicates the number of 8-byte (64 bits) units of data left to be read in the suspended sequence. This field is valid only when STATE is 01 or 11.
8 —	Reserved
7-6 SPDBUF	Suspended Buffer Indicates the number of the suspended buffer. This field is valid only when STATE is 01 or 11.
5-4 SPDSBUF	Suspended Sub-Buffer Indicates the number of the suspended sub-buffer. This field is valid only when STATE is 01 or 11.
3-2 —	Reserved
1-0 STATE	Suspended Resume State Indicates the state of the suspended transaction. 00b - No suspended transaction pending 01b - Suspended transaction waiting to be resumed 10b - Not applicable 11b - Suspended transaction is resumed

41.7.2.30 Sequence Pointer Clear (SPTRCLR)

Offset

Register	Offset
SPTRCLR	16Ch

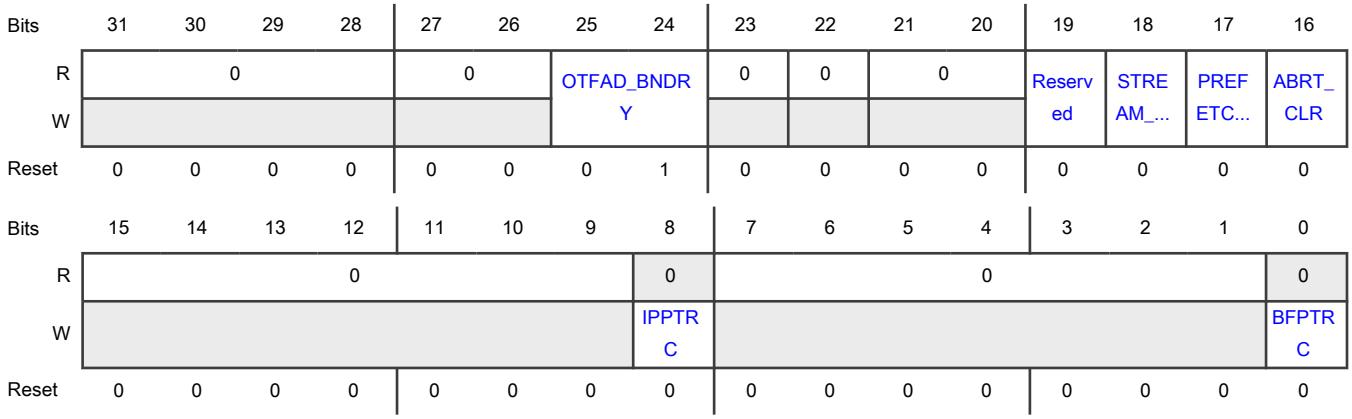
Function

Supports the following controls:

- Clears the IP and buffer sequence pointers.
- Disables prefetch.
- Terminates a flash memory transaction and clears the AHB buffer.
- Enables AHB read as a streaming port.
- Specifies the OTFAD prefetch address boundary.

Sequence pointers specify the index of instructions within the LUT entry that is to be executed next. For example, if the LUT entry ends on a JMP_ON_CS value of 2, the stored index is 2. You should reset the sequence pointers whenever you update the sequence ID (SEQID, SFP_TG_IPCR[SEQID], BFGENCR[SEQID_WR] , SEQID_WR). .

Diagram



Fields

Field	Function
31-28 —	Reserved
27-26 —	Reserved
25-24	OTFAD Prefetch Address Boundary Specifies the OTFAD prefetch boundary.

Table continues on the next page...

Table continued from the previous page...

Field	Function
OTFAD_BNDRY	00b - No prefetch address boundary 01b - Prefetch address boundary is 1K. All other values are reserved.
23 —	Reserved
22 —	Reserved
21-20 —	Reserved
19 —	Reserved
18 STREAM_EN	Enable Streaming Port Enables AHB read as a streaming port with 64K prefetch size. 0b - Enables 1b - Disables
17 PREFETCH_DIS	Prefetch Disable Disables the prefetch mechanism for AHB reads. It is not based on dynamic programming; you should program this field before any other XSPI action. If you enable high priority for either AHB buffer 0 or 1 (HP_EN or HP_EN is 1) then you must write 0 to this field. 0b - Enables 1b - Disables
16 ABRT_CLR	Flash Memory Abort—AHB Buffer Clear Clears the AHB buffer pointers regardless of prefetch disable. It also terminates any current flash memory transaction and rejects any ongoing AHB read with an error. This is a dynamic field; the action takes place immediately. After clearing the AHB buffer pointers, XSPI writes 0 to this field. 0b - No action 1b - Clears pointers and terminates transaction
15-9 —	Reserved
8	IP Pointer Clear

Table continues on the next page...

Table continued from the previous page...

Field	Function
IPPTRC	Clears the sequence pointer for IP accesses (IPCR[SEQID]). This is a self-clearing field; it always reads 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears the sequence pointer
7-1 —	Reserved
0 BFPTRC	Buffer Pointer Clear Clears the sequence pointer for AHB accesses (BFGENCR[SEQID]). This is a self-clearing field; it always reads 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears the sequence pointer

41.7.2.31 Serial Flash Memory Top Address (SFA1AD)

Offset

Register	Offset
SFA1AD	180h

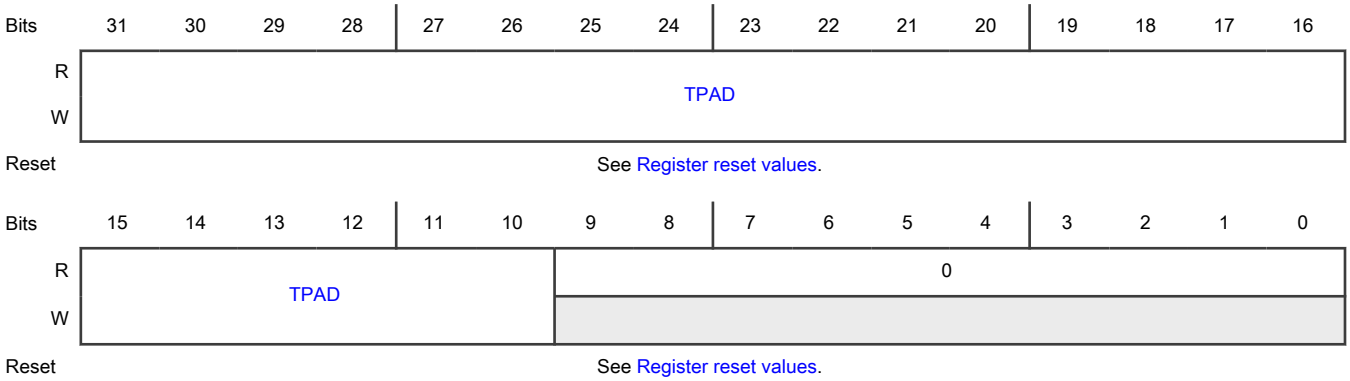
Function

Specifies the top address of the corresponding serial flash memory. See Flash memory device selection.

Special write access is permitted if all these conditions apply:

- IP access is not asserted (SR[IP_ACC] = 0)
- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Register reset values

Register	Reset value
SFA1AD	XSPI0: 2C00_0000h XSPI1: 0C00_0000h XSPI2: 6800_0000h

Fields

Field	Function
31-10 TPAD	Top Address In effect, TPADxn is the first location of the next memory.
9-0 —	Reserved

41.7.2.32 Serial Flash Memory Top Address (SFA2AD)

Offset

Register	Offset
SFA2AD	184h

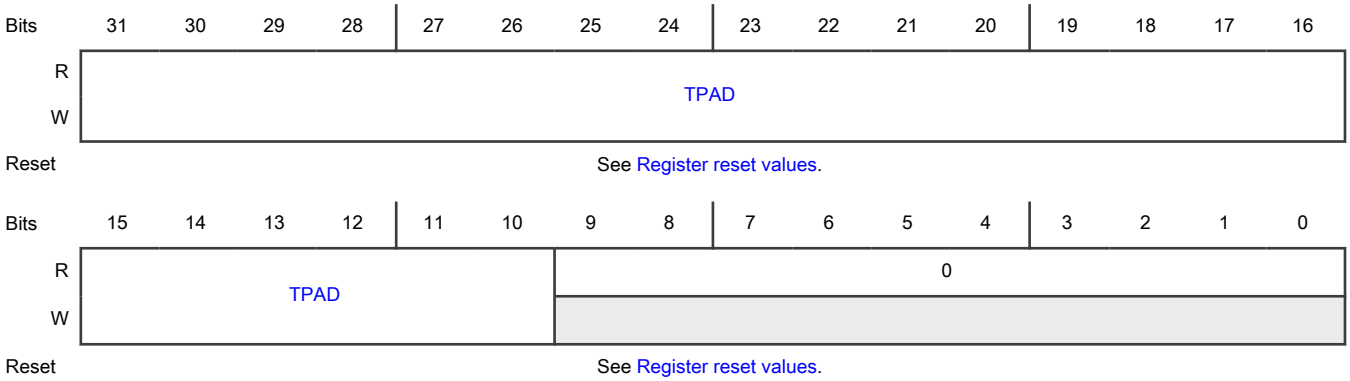
Function

Specifies the top address of the corresponding serial flash memory. See Flash memory device selection.

Special write access is permitted if all these conditions apply:

- IP access is not asserted (SR[IP_ACC] = 0)
- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Register reset values

Register	Reset value
SFA2AD	XSPI0: 3000_0000h XSPI1: 1000_0000h XSPI2: 7000_0000h

Fields

Field	Function
31-10 TPAD	Top Address In effect, TPADxn is the first location of the next memory.
9-0 —	Reserved

41.7.2.33 Data Learning Pattern (DLPR)

Offset

Register	Offset
DLPR	190h

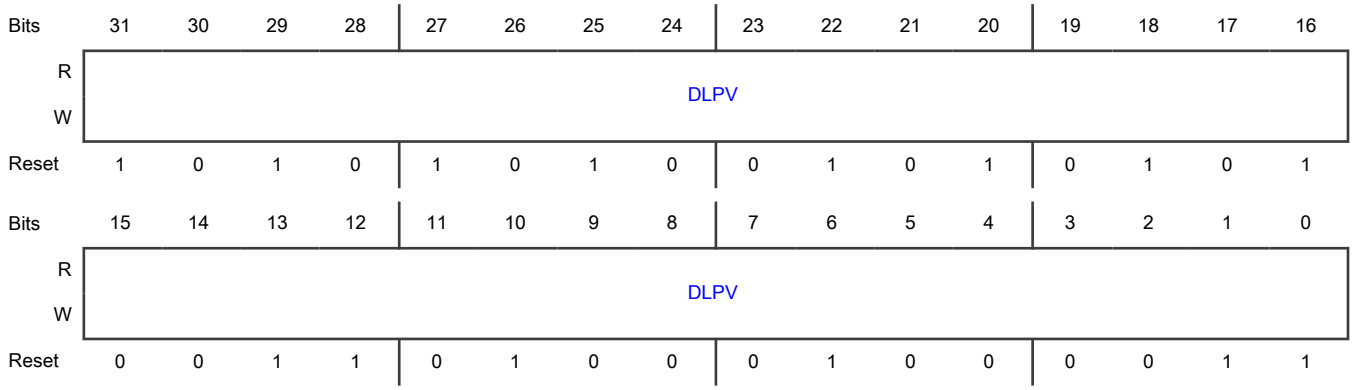
Function

Specifies the data learning pattern for data learning in the DQS modes.

Special write access is permitted if both:

- IP access is not asserted (SR[IP_ACC] = 0)
- AHB read access is not asserted (SR[AHB_ACC] = 0)

Diagram



Fields

Field	Function
31-0 DLPV	Data Learning Pattern Specifies the data learning pattern. XSPI uses this value in the SDR/DDR modes along with the DATA_LEARN instruction. You can specify different patterns for selected IOs. That is, if you select IO1 or IO3 or both (using DLCR[DLP_SEL_FA] , XSPI provides bits 31–16 for the IO3 pattern and 15–0 for the IO1 pattern. You should store the pattern bit-wise in little endian format. See Data learning.

41.7.2.34 Flash Memory A Failing Address Status (FAILA_ADDR)

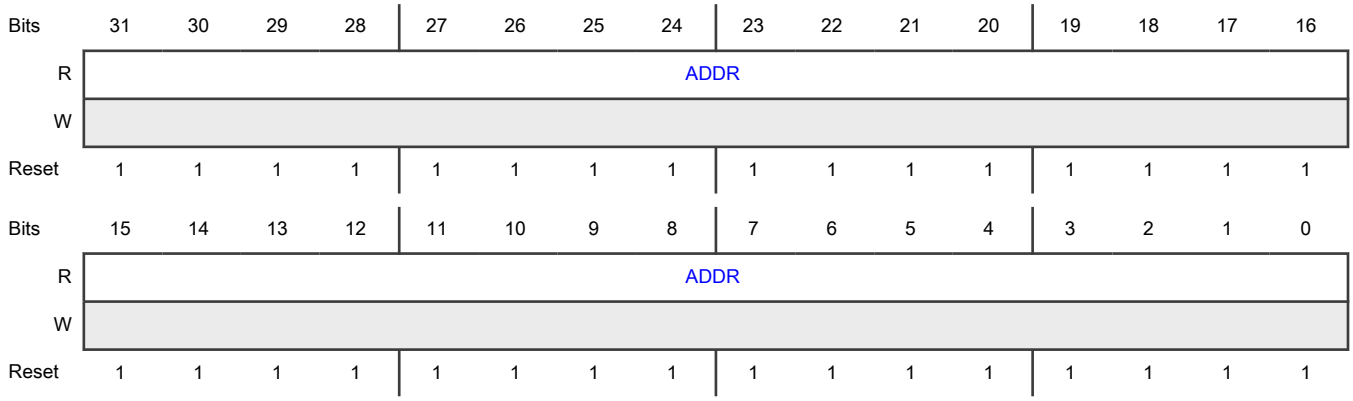
Offset

Register	Offset
FAILA_ADDR	194h

Function

Indicates the flash memory address where a data learning error has occurred.

Diagram



Fields

Field	Function
31-0 ADDR	Failing Address

41.7.2.35 RX Buffer Data (RBDR0 - RBDR63)

Offset

For a = 0 to 63:

Register	Offset
RBDRa	200h + (a × 4h)

Function

Contains the individual entries in the RX buffer. See Byte ordering of serial flash memory read data for the byte ordering scheme.

RBDR0 corresponds to the position of the read pointer within the RX buffer. RBSR[RDBFL] indicates the number of available valid entries that have been written to the RX buffer.

Example 1: The RX buffer is filled completely with 128 words. In this example, the address range for valid read access ranges from RBDR0 to RBDR127.

Example 2: The RX buffer contains five valid words (RBSR[RDBFL] = 5). In this example, RBDR4 points to the last valid entry.

Any access beyond the range of valid RX buffer entries returns undefined results.

NOTE

If the number of valid entries is greater than 64, accessing the buffer data beyond RBDR[63] requires an RX buffer pop event. See RBDF and RX buffer POP event.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 RXDATA	RX Data Indicates the data in corresponding RX buffer entry.

41.7.2.36 LUT Key (LUTKEY)

Offset

Register	Offset
LUTKEY	300h

Function

Specifies the key to lock and unlock the LUT. See Lookup table (LUT).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Fields

Field	Function
31-0 KEY	Key The key is 5AF05AF0h and the read value is always 5AF05AF0h.

41.7.2.37 LUT Lock Configuration (LCKCR)

Offset

Register	Offset
LCKCR	304h

Function

In conjunction with LUTKEY, locks or unlocks the LUT. For a successful lock operation, you must write to LCKCR immediately after writing to LUTKEY. See Locking the LUT and Unlocking the LUT. You cannot write the same value to both LOCK and UNLOCK. In other words, LOCK and UNLOCK cannot both be 00b or both be 11b.

Special write access is permitted after writing to LUTKEY.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														UNLOCK	LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-2 —	Reserved
1 UNLOCK	Unlock LUT Unlocks the LUT. You can write to this field only immediately after writing 5AF05AF0h to LUTKEY. 0b - No action 1b - Unlocks

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 LOCK	Lock LUT Locks or unlocks the LUT. You can write to this field only immediately after writing 5AF05AF0h to LUTKEY. 0b - No action 1b - Locks

41.7.2.38 Lookup Table (LUT0 - LUT79)

Offset

For a = 0 to 79:

Register	Offset
LUTa	310h + (a × 4h)

Function

Specifies instruction-operand pairs. Each LUT register can contain two 16-bit (2-byte) pairs. See Lookup table (LUT).

Special write access is permitted if the LUT is unlocked and the global configuration lock is disabled (MGC[GCLCK] = 00b). See Unlocking the LUT.

Only privilege initiators can access this register. A write attempt by any other initiator generates a bus transfer error. See Programmable sequence engine.

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	See Register reset values.															

Register reset values

Register	Reset value
LUT0	XSPI0–XSPI2: 0818_0403h

Table continues on the next page...

Table continued from the previous page...

Register	Reset value
LUT1	XSPI0–XSPI2: 2400_1C08h
LUT2–LUT79	XSPI0–XSPI2: 0000_0000h

Fields

Field	Function
31-26 INSTR1	Instruction 1 Specifies instruction 1.
25-24 PAD1	Pads for INSTR1 Specifies the number of pads for instruction 1. 00b - 1 01b - 2 10b - 4 11b - 8
23-16 OPRND1	Operand for INSTR1 Specifies the operand for instruction 1.
15-10 INSTR0	Instruction 0 Specifies instruction 0.
9-8 PAD0	Pads for INSTR0 Specifies the number of pads for instruction 0. 00b - 1 01b - 2 10b - 4 11b - 8
7-0 OPRND0	Operand for INSTR0 Specifies the operand for instruction 0.

41.7.2.39 AHB Buffer n Sub Buffer m Start and End Address Range (BUF0_0_ADDR_RANGE - BUF3_3_ADDR_RANGE)

Offset

For n = 0 to 3; m = 0 to 3:

Register	Offset
BUFn_m_ADDR_RANGE	480h + (n × 10h) + (m × 4h)

Function

Specifies the address range (start and end) for the corresponding AHB buffer when you enable sub-buffer division (write 1 to BUF*n*CR[SUB_DIV_EN]). If the AHB address falls within this address range and results in a buffer miss, then XSPI starts data prefetch in this buffer.

Special write access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ENDADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STARTADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 ENDADR	End Address Specifies the upper 16-bits of the end address range for the associated sub-buffer. Lower 16-address bits are treated as FFFF.
15-0 STARTADR	Start Address Specifies the upper 16-bits of the start address range for the associated sub-buffer. Lower 16-address bits are treated as 0000.

41.7.2.40 AHB Buffer Status (AHB_BUF_STATUS)

Offset

Register	Offset
AHB_BUF_STATUS	4C0h

Function

Indicates the status of pre-fetched data in the AHB buffer.

NOTE

- The AHB buffer asserts a Partial Full status when the difference between the End Address and the Read Address is less than the size of the sub-buffer.
- When sub-buffer division is enabled, the BUFnCR[ADATSZ] field is ignored.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF3_3		BUF3_2		BUF3_1		BUF3_0		BUF2_3		BUF2_2		BUF2_1		BUF2_0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF1_3		BUF1_2		BUF1_1		BUF1_0		BUF0_3		BUF0_2		BUF0_1		BUF0_0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30: BUF3_3 29-28: BUF3_2 27-26: BUF3_1 25-24: BUF3_0	AHB Buffer 3 Sub Buffer n Status 00b - Empty 01b - Full 10b - Partially full All other values are reserved.
23-22: BUF2_3 21-20: BUF2_2 19-18: BUF2_1 17-16: BUF2_0	AHB Buffer 2 Sub Buffer n Status 00b - Empty 01b - Full 10b - Partially full All other values are reserved.
15-14: BUF1_3 13-12: BUF1_2 11-10: BUF1_1 9-8: BUF1_0	AHB Buffer 1 Sub Buffer n Status 00b - Empty 01b - Full 10b - Partially full All other values are reserved.
7-6: BUF0_3 5-4: BUF0_2	AHB Buffer 0 Sub Buffer n Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
3-2: BUF0_1 1-0: BUF0_0	**NOTE** When BUF0CR[SUB_DIV_EN] = 0, BUF0_0 indicates the status of buffer 0. 00b - Empty 01b - Full 10b - Partially full All other values are reserved.

41.7.2.41 AHB Buffer Hit/Miss Performance Monitor Control (AHB_PERF_CTRL)

Offset

Register	Offset
AHB_PERF_CTRL	4C4h

Function

Specifies the buffer performance and counter operation. See AHB read performance monitors.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SUB_BUF_SEL 3		SUB_BUF_SEL 2		SUB_BUF_SEL 1		SUB_BUF_SEL 0		0			TCNTO	BUF3_HI...	BUF2_HI...	BUF1_HI...	BUF0_HI...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF3_MI...	BUF2_MI...	BUF1_MI...	BUF0_MI...	0						CNTSTART	0	BUF3_EN	BUF2_EN	BUF1_EN	BUF0_EN
W												CNTSTP				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30: SUB_BUF_SEL 3	Sub-buffer Selection for Buffer n Performance Monitoring Specifies the sub-buffer to use for performance monitoring of the corresponding buffer. 00b - 0

Table continues on the next page...

Table continued from the previous page...

Field	Function
29-28: SUB_BUF_SEL 2 27-26: SUB_BUF_SEL 1 25-24: SUB_BUF_SEL 0	01b - 1 10b - 2 11b - 3
23-21 —	Reserved
20 TCNTO	Time Counter Overflow Indicates an overflow of the time counter. To clear this flag, write 1 to PERFOVF. 0b - No overflow 1b - Overflow
19-16 BUFn_HIT_OVF	Buffer n Hit Counter Overflow Indicates an overflow of the corresponding buffer hit counter. To clear this flag, write 1 to PERFOVF. 0b - No overflow 1b - Overflow
15-12 BUFn_MISS_OVF	Buffer n Miss Counter Overflow Indicates an overflow of the corresponding buffer miss counter. To clear this flag, write 1 to PERFOVF. 0b - No overflow 1b - Overflow
11-6 —	Reserved
5 CNTSTART	Performance Counter Start Starts the performance counter. Stopping the counter (writing 1 to CNTSTP) automatically clears this field. XSPI automatically clears this field in the event of any overflow. 0b - No effect 1b - Starts
4 CNTSTP	Performance Counter Stop Stops the performance counter. Stopping the counter automatically clears CNTSTART. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Stops counter
3-0 BUFn_EN	Buffer n Counter Enable Enables the corresponding buffer counter. 0b - Disables 1b - Enables

41.7.2.42 AHB Performance Monitor Time Counter (AHB_PERF_TIME_CNT)

Offset

Register	Offset
AHB_PERF_TIME_CNT	4C8h

Function

Indicates, in AHB clock cycles, how long the performance monitor was running.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PERF_TIME_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PERF_TIME_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PERF_TIME_COUNT	Time Count Indicates, in AHB clock cycles, how long the performance monitor was running. Updated when you stop the performance monitoring or when this counter overflows.

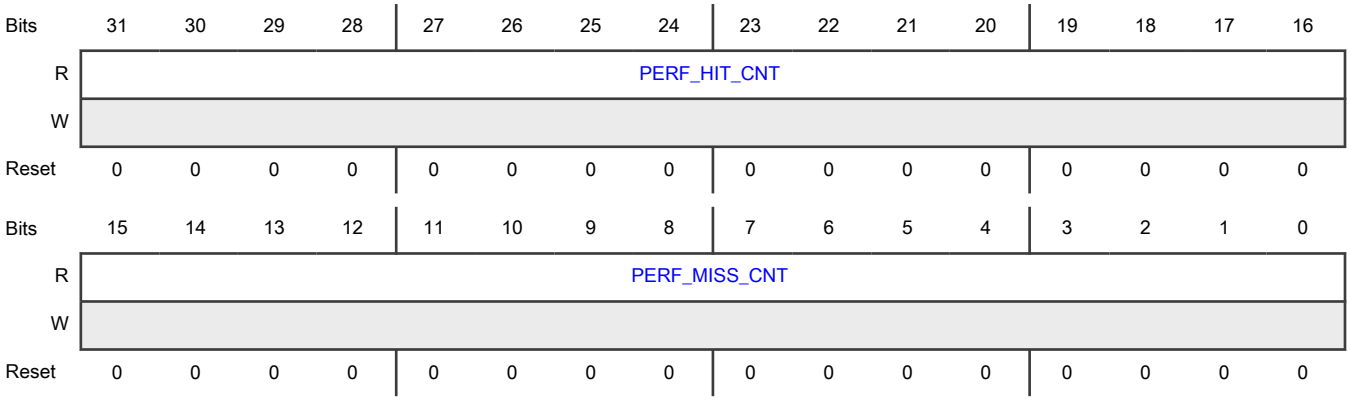
41.7.2.43 AHB Buffer n Performance Monitor (AHB_PERF_BUF0 - AHB_PERF_BUF3)

Offset

Register	Offset
AHB_PERF_BUF0	4CCh
AHB_PERF_BUF1	4D0h
AHB_PERF_BUF2	4D4h
AHB_PERF_BUF3	4D8h

Function
Indicates the hit and miss counts for the corresponding AHB buffer.

Diagram



Fields

Field	Function
31-16 PERF_HIT_CNT	Performance Hit Count Indicates the number of hits that have occurred on the corresponding AHB buffer.
15-0 PERF_MISS_CNT	Performance Miss Count Indicates the number of misses that have occurred on the corresponding AHB buffer and the sub-buffer selected by SUB_BUF_SEL0.

41.7.2.44 AHB HREADY Timeout (AHRDYTO)

Offset

Register	Offset
AHRDYTO	4DCh

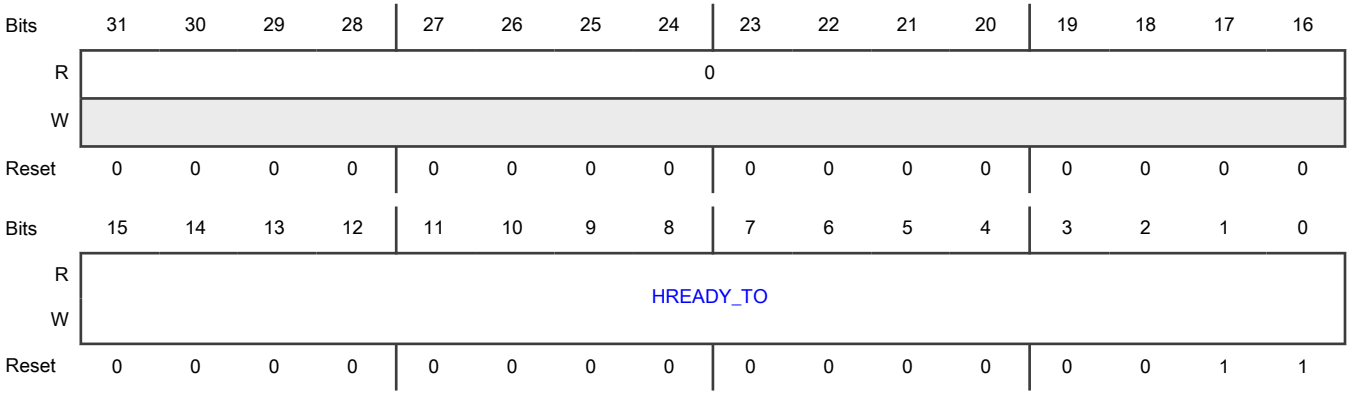
Function

Specifies how long XSPI can hold HREADY low while waiting for response to an AHB transfer.

Special write access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 HREADY_TO	AHB HREADY Low Timeout Counter Value Specifies, in multiples of 50000 AHB clock cycles, how long XSPI holds HREADY low before triggering an AHB illegal transaction error (FR[AITEF]).

41.7.2.45 AHB Error Payload High (AHB_ERR_PAYLOAD_HI)

Offset

Register	Offset
AHB_ERR_PAYLOAD_HI	4E0h

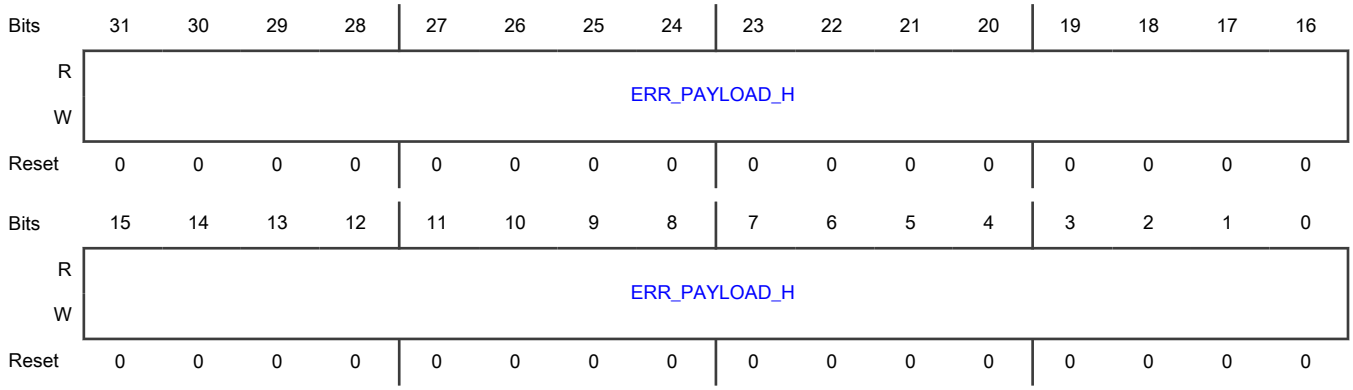
Function

Specifies the data to send on AHB HRDATA[63:32] in the event of certain error situations. Depending on the Error situation, HRESP can be OK or controlled by SOCCR[HRESP_ERR_MASK]). For a list of the error situations, see FR error flags overview.

Special write access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-0 ERR_PAYLOAD_H	Error Payload High Specifies the data to send on AHB HRDATA[63:32] in the event of certain error situations.

41.7.2.46 AHB Error Payload Low (AHB_ERR_PAYLOAD_LO)

Offset

Register	Offset
AHB_ERR_PAYLOAD_LO	4E4h

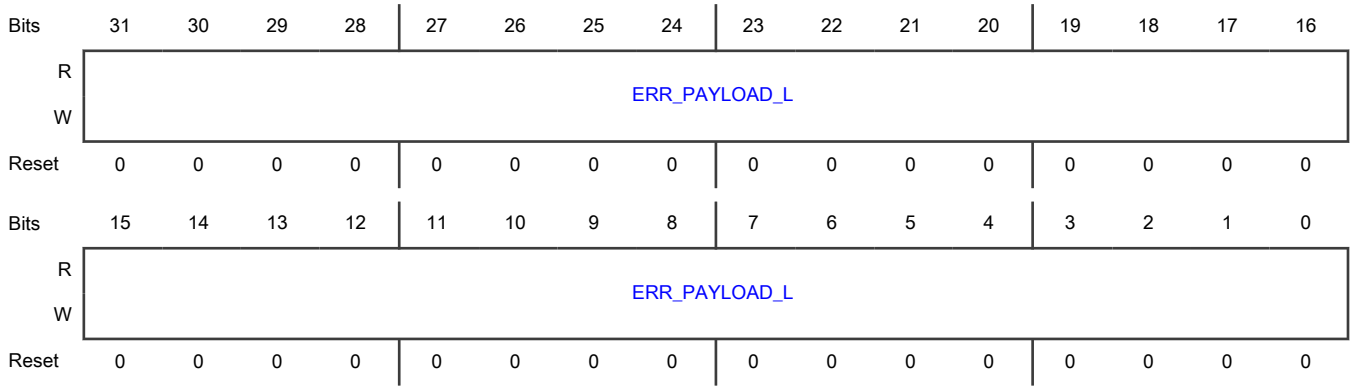
Function

Specifies the data to send on AHB HRDATA[31:0] in the event of certain error situations. HRESP remains OK in such situations. For a list of the error situations, see FR error flags overview.

Special write access is permitted if both:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)

Diagram



Fields

Field	Function
31-0 ERR_PAYLOAD_L	Error Payload Low Specifies the data to send on AHB HRDATA[31:0] in the event of certain error situations.

41.7.2.47 AHB Read Error Address (AHB_RD_ERR_ADDR)

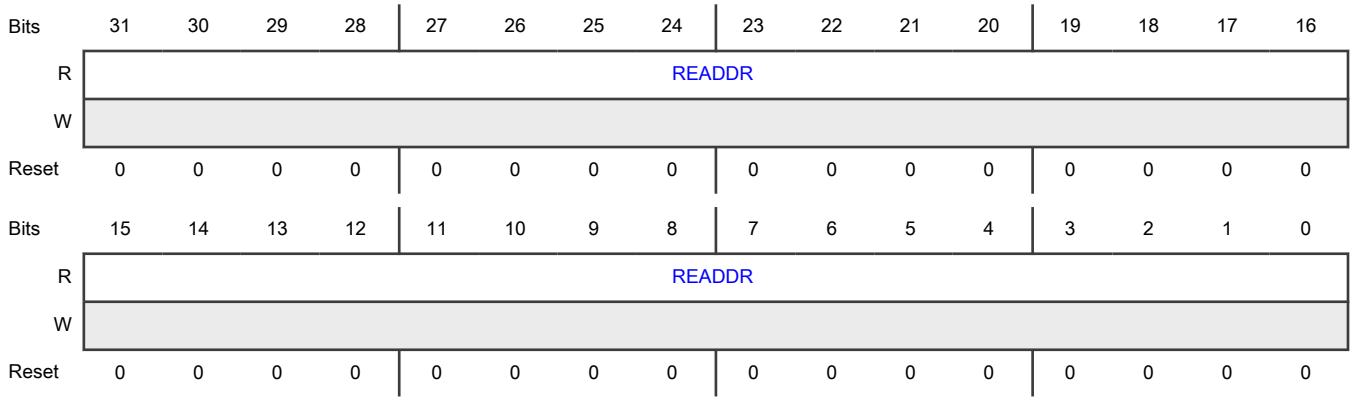
Offset

Register	Offset
AHB_RD_ERR_ADDR	4E8h

Function

Indicates the AHB read error address.

Diagram



Fields

Field	Function
31-0 READDR	Read Error Address Indicates the address of the AHB read transaction that triggered the error flagged in FR[RDADDR]. To clear this field, clear the read error flag (write 1 to FR[RDADDR]).

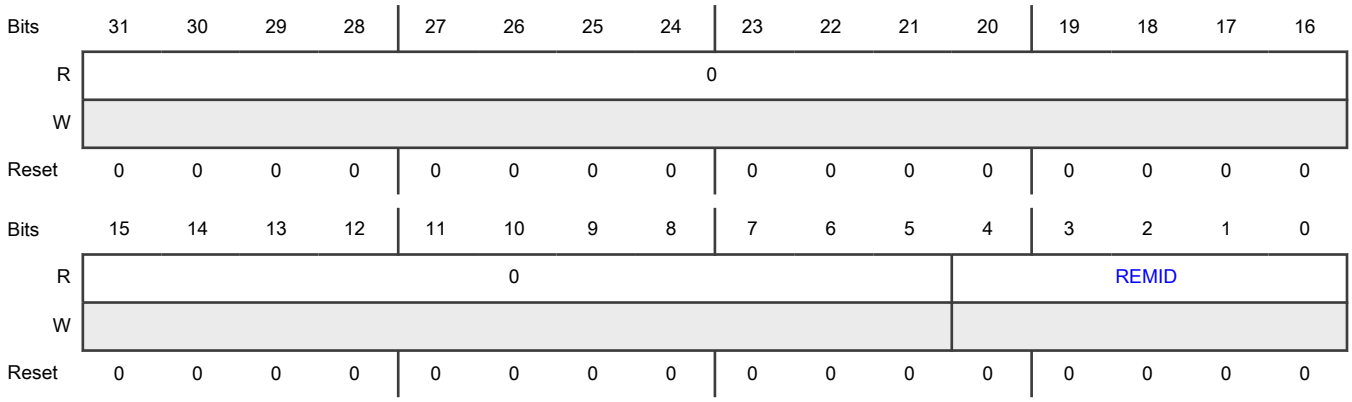
41.7.2.48 AHB Read Error Initiator ID (AHB_RD_ERR_MID)

Offset

Register	Offset
AHB_RD_ERR_MID	4ECh

Function
Indicates the AHB read error initiator ID.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 REMID	Read Error Initiator ID Indicates the initiator ID of the AHB read transaction that triggered the error flagged in RDADDR. To clear this field, clear the read error flag (write 1 to FR[RDADDR]). See the chip-specific information for the supported initiator IDs.

41.7.2.49 Suspend Transaction Address (SPNDST_ADDR)

Offset

Register	Offset
SPNDST_ADDR	4F0h

Function

Indicates the suspend transaction address.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ASAHBR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ASAHBR													0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-3 ASAHBR	Address of Suspended AHB Read Transaction Indicates the address of the suspended AHB read transaction. This field is valid only when there is a suspended AHB read transaction waiting to be resumed (SPNDST[STATE] = 01b).
2-0 —	Reserved

41.7.2.50 Page Program Wait Time Counter (PPWF_TCNT)

Offset

Register	Offset
PPWF_TCNT	4F4h

Function

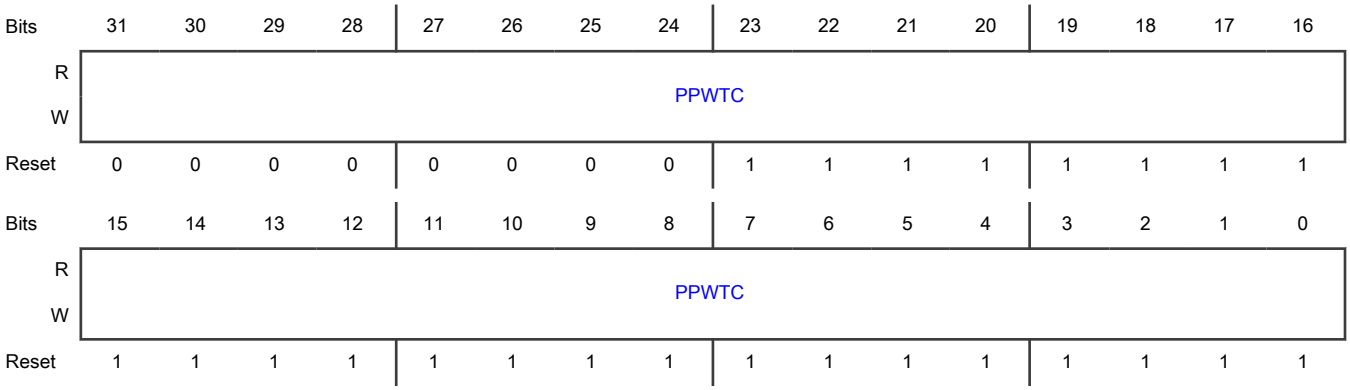
Specifies the page program wait time.

Special write access is permitted if all these conditions apply:

- AHB read access is not asserted (SR[AHB_ACC] = 0)

- AHB write access is not asserted (SR[AWRACC] = 0)
- Page program wait flag is not set (FR[PPWF] = 0)

Diagram



Fields

Field	Function
31-0 PPWTC	Page Program Wait Time Counter Value Specifies, in AHB clock cycles, how long XSPI waits before triggering a read of the flash status register, which is reflected in PPW_RDSR[RDSR]. See Page program wait handling.

41.7.2.51 Page Program Wait Read Status (PPW_RDSR)

Offset

Register	Offset
PPW_RDSR	4F8h

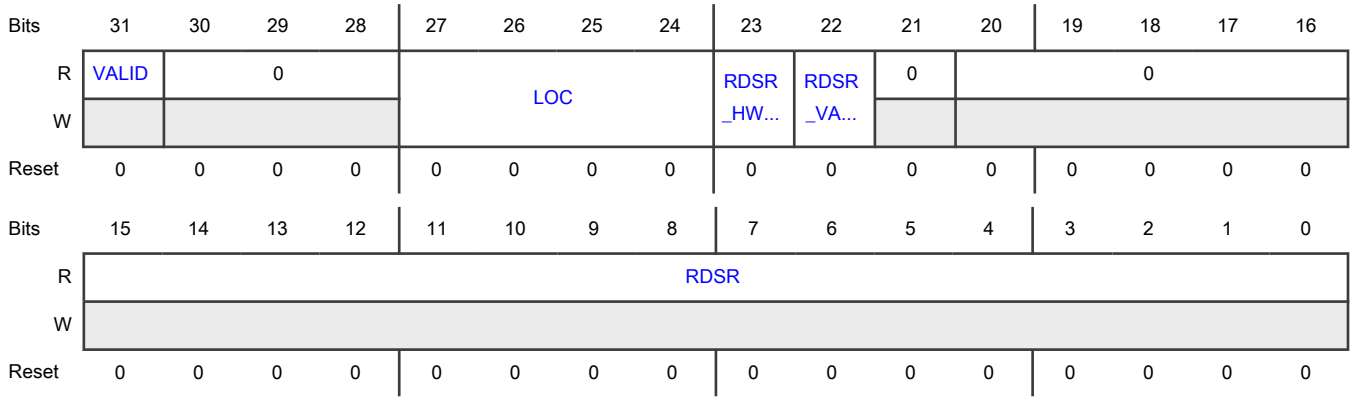
Function

Indicates the page program wait read status.

Special write access is permitted if all these conditions apply:

- AHB read access is not asserted (SR[AHB_ACC] = 0)
- AHB write access is not asserted (SR[AWRACC] = 0)
- Page program wait flag is not set (FR[PPWF] = 0)

Diagram



Fields

Field	Function
31 VALID	Valid Indicates whether the value of RDSR is valid. 0b - Not valid 1b - Valid
30-28 —	Reserved
27-24 LOC	Location of RDSR Specifies the bit in RDSR that indicates the page program wait has expired.
23 RDSR_HWORD_SEL	RDSR Half Word Select Specifies which word (upper or lower) is selected from flash. 0b - RDSR contains 16 least significant bits (15:0) of RDSR read from flash. 1b - RDSR contains 16 most significant bits (31:16) of RDSR read from flash.
22 RDSR_VAL_CHK	RDSR Value Check Specifies the value to be checked on the selected bit location for determining if the page program wait period is expired. 0b - Check for 0 1b - Check for 1
21 —	Reserved
20-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-0 RDSR	Value of Flash Status Register Read Indicates the data returned by a flash memory status read.

41.7.2.52 IPED Context Initial Vector (IPEDCTX0IV0 - IPEDCTX15IV1)

Offset

For m = 0 to 15; n = 0 to 1:

Register	Offset
IPEDCTXmIVn	520h + (m × 20h) + (n × 4h)

Function

Specifies the lowest or highest 32 bits of the initial vector for the context region:

- IV 0: lowest 32 bits
- IV 1: highest 32 bits

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTX0IV0–IPEDCTX15IV1	—
XSPI1	IPEDCTX0IV0–IPEDCTX15IV1	—
XSPI2	—	IPEDCTX0IV0–IPEDCTX15IV1

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTX_IV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTX_IV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CTX_IV	32 bits of IV for the context region

41.7.2.53 IPED Start Address Region (IPEDCTX0START - IPEDCTX15START)

Offset

For m = 0 to 15:

Register	Offset
IPEDCTXmSTART	528h + (m × 20h)

Function

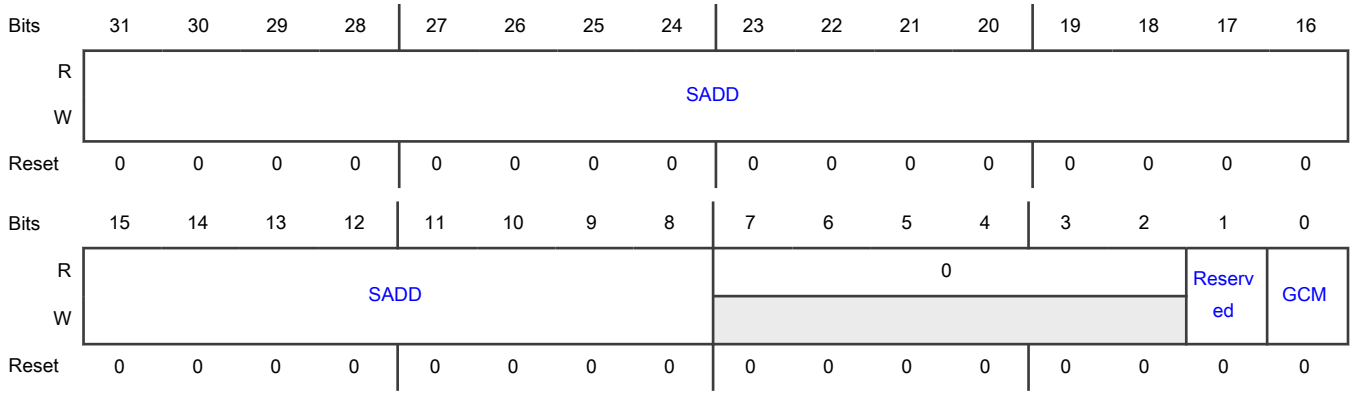
Specifies the lowest address of the IPED region and region operation. If IPEDCTXCTRL0[CTX*n*_FREEZE] is 10b and IPEDCTXCTRL1[CTX*n*_FREEZE] is 10b, then IPEDCTX*n*START is writable.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTX0START–IPEDCTX15START	—
XSPI1	IPEDCTX0START–IPEDCTX15START	—
XSPI2	—	IPEDCTX0START–IPEDCTX15START

Diagram



Fields

Field	Function
31-8 SADD	Start Address Specifies the start address of the corresponding AHB or IPS memory region. This value must be 256-byte aligned.
7-2 —	Reserved
1 —	Reserved
0 GCM	GCM Mode Region Specifies that the corresponding region is a GCM region. 0b - CTR 1b - GCM

41.7.2.54 IPED End Address Region (IPEDCTX0END - IPEDCTX15END)

Offset

For m = 0 to 15:

Register	Offset
IPEDCTXmEND	52Ch + (m × 20h)

Function

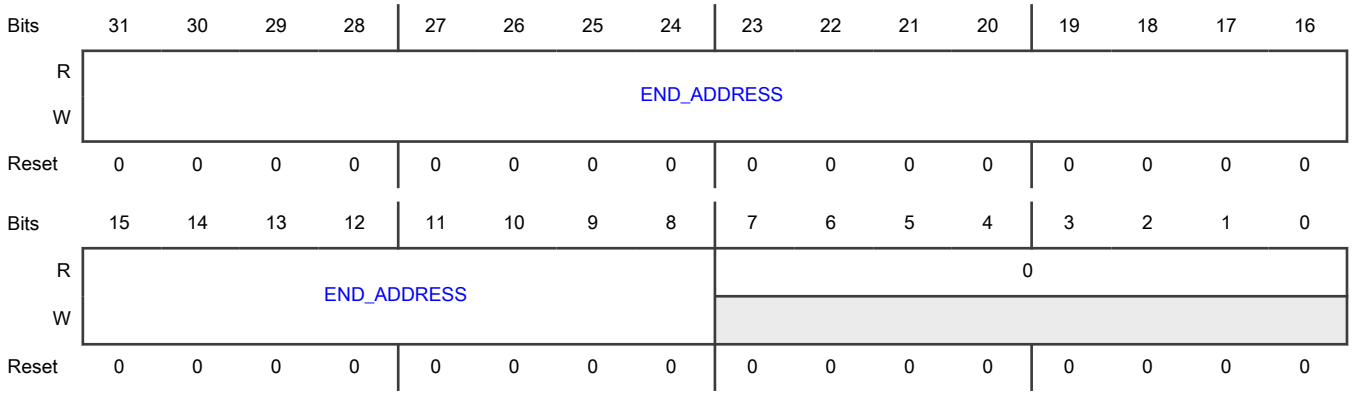
Specifies the highest address of IPED region. If IPEDCTXCTRL0[CTX*n*_FREEZE] is 10b and IPEDCTXCTRL1[CTX*n*_FREEZE] is 10b, then IPEDCTX*n*END is writable.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTX0END–IPEDCTX15END	—
XSPI1	IPEDCTX0END–IPEDCTX15END	—
XSPI2	—	IPEDCTX0END–IPEDCTX15END

Diagram



Fields

Field	Function
31-8 END_ADDRESS	End Address Specifies the end address of the IPED region. This value must be 256-byte aligned. XSPI compares this AHB address with the system bus address. **NOTE** The end address of the context region is defining the upper limit of that region through (END_ADDRESS - 1) value.
7-0 —	Reserved

41.7.2.55 IPED Context Additional Authentication Data (IPEDCTX0AAD0 - IPEDCTX15AAD1)

Offset

For m = 0 to 15; n = 0 to 1:

Register	Offset
IPEDCTXmAADn	530h + (m × 20h) + (n × 4h)

Function

Specifies lowest or highest 32 bits of additional authentication data for the context region:

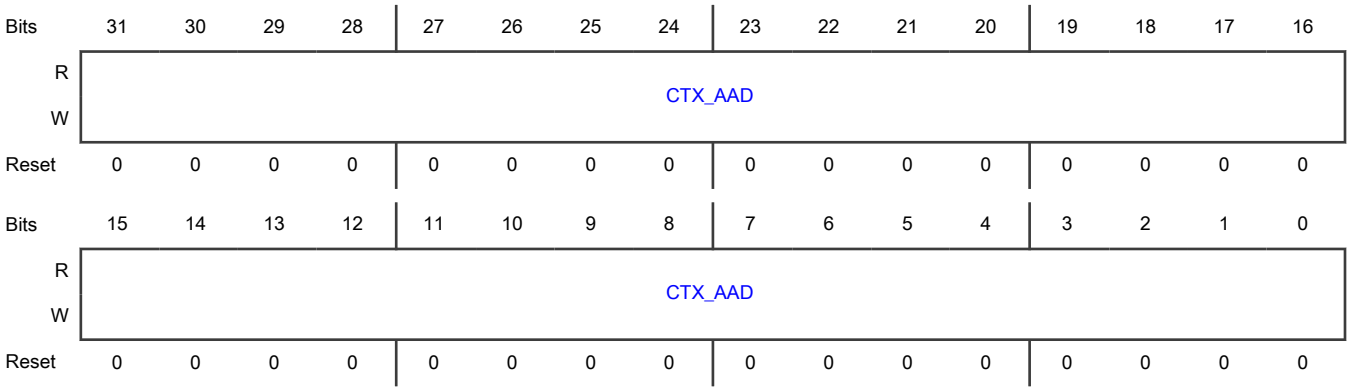
- ADD 0: lowest 32 bits
- ADD 1: highest 32 bits

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTX0AAD0–IPEDCTX15AAD1	—
XSPI1	IPEDCTX0AAD0–IPEDCTX15AAD1	—
XSPI2	—	IPEDCTX0AAD0–IPEDCTX15AAD1

Diagram



Fields

Field	Function
31-0 CTX_AAD	32 bits of AAD for the region

41.7.2.56 IPED Function Control (IPEDCTRL)

Offset

Register	Offset
IPEDCTRL	720h

Function

Specifies the IPED function.

Special write access is permitted if all of these conditions apply:

- IP access is not asserted (IP_ACC = 0)
- AHB read access is not asserted (AHB_ACC = 0)
- AHB write access is not asserted (AWRACC = 0)

NOTE

The following fields should be programmed in isolation: IPEDCTRL[AHBXEXRE], IPEDCTRL[AHBXEXWE], IPEDCTRL[IPSXEXWE].

Additionally, the following fields must also be configured independently: IPEDCTRL[AHBGCMRD], IPEDCTRL[AHBGCMWR], IPEDCTRL[IPSGCMWR], IPEDCTRL[AHBRD_EN], IPEDCTRL[AHBWR_EN], IPEDCTRL[IPSWR_EN].

Dynamic switching between XEX mode and other IPED modes is not supported. XEX mode must operate in isolation.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTRL	—
XSPI1	IPEDCTRL	—
XSPI2	—	IPEDCTRL

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															IPS_GCM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0			IPED_XE...	IPED_SW...	IPED_PR...	AHBGCMRD	AHGCMWR	IPGCMWR	Reserved	AHBRD_EN	AHBWR_EN	IPWR_EN	IPED_EN	CONFIG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-17 —	Reserved
16 IPS_GCM	IPS Transaction Region Status Indicates whether an IPS transaction occurs in a GCM region. You use this information to program the watermark in IPS GCM mode. 0b - Non-GCM region, either CTR or bypass 1b - GCM region

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>15
—</td><td>Reserved</td></tr>
<tr><td>14-12
—</td><td>Reserved</td></tr>
<tr><td>11
IPED_XEX_EN</td><td>IPED XEX Interface Enable
Enables the IPED XEX interface.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>XSPI0</td><td>—</td><td>IPEDCTRL</td></tr><tr><td>XSPI1</td><td>IPEDCTRL</td><td>—</td></tr></table>
0b - Bypasses IPED logic and writes unencrypted or undecrypted data
1b - Enables</td></tr>
<tr><td>10
IPED_SWRESET</td><td>IPED CTR/GCM Software Reset
Resets IPED CTR/GCM operation.
NOTE
Set this bit when software reset for AHB domain (MCR[SWRSTHD]), serial flash memory domain (MCR[SWRSTSD]), and IPS TG queue (MCR[IPS_TG_RST]) are already set.
0b - No reset
1b - Resets</td></tr>
<tr><td>9
IPED_PROTECT</td><td>IPED Protection
Enables IPED protection. If enabled, only privileged controllers can write to the IPED configuration registers. Otherwise, there are no restrictions.
0b - Disables
1b - Enables</td></tr>
<tr><td>8
AHBGCMRD</td><td>AHB Read IPED GCM Decryption Enable
Enables GCM encryption for AHB reads.
0b - Bypasses AHB data and reads unencrypted data
1b - Enables</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 AHGCMWR	AHB Write IPED GCM Encryption Enable Enables GCM encryption for AHB writes. 0b - Bypasses AHB data and writes unencrypted data 1b - Enables
6 IPGCMWR	IP Write GCM Enable Enables GCM encryption for IP writes. 0b - Bypasses IPS-based data and writes unencrypted data 1b - Enables
5 —	Reserved
4 AHBRD_EN	AHB Read IPED CTR Mode Decryption Enable Enables AHB CTR mode encryption for read transactions. 0b - Bypasses AHB-based data and reads unencrypted 1b - Enables
3 AHBWR_EN	AHB Write IPED CTR Mode Encryption Enable Enables AHB CTR mode encryption for write transactions. 0b - Bypasses AHB-based data and writes unencrypted 1b - Enables
2 IPWR_EN	IP Write IPED CTR Mode Enable Enables IPS CTR mode encryption. 0b - Bypasses IPS-based data and writes unencrypted data 1b - Enables
1 IPED_EN	IPED CTR-GCM Interface Enable Enables CTR/GCM operation. 0b - Bypasses IPED logic and allows writing of unencrypted or undecrypted data 1b - Enables
0 CONFIG	IPED CTR/GCM Interface Configure Specifies the CTR/GCM configuration. **NOTE** This field does not apply to XEX based encryption.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Prince 12 rounds 1b - Prince 22 rounds

41.7.2.57 IPED Context Control 0 (IPEDCTXCTRL0)

Offset

Register	Offset
IPEDCTXCTRL0	728h

Function

Freezes the IPEDCTX*n*START and IPEDCTX*n*END. If the bit1 and bit0 CTXn_FREEZE field are different and CTXn_FREEZE1 field of IPEDCTXCTRL1 is 2'b10, then the CTXn_FREEZE field is writable.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTXCTRL0	—
XSPI1	IPEDCTXCTRL0	—
XSPI2	—	IPEDCTXCTRL0

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTX15_FREEZE		CTX14_FREEZE		CTX13_FREEZE		CTX12_FREEZE		CTX11_FREEZE		CTX10_FREEZE		CTX9_FREEZE		CTX8_FREEZE	
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTX7_FREEZE		CTX6_FREEZE		CTX5_FREEZE		CTX4_FREEZE		CTX3_FREEZE		CTX2_FREEZE		CTX1_FREEZE		CTX0_FREEZE	
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Fields

Field	Function
31-30: CTX15_FREEZE 29-28: CTX14_FREEZE 27-26: CTX13_FREEZE 25-24: CTX12_FREEZE 23-22: CTX11_FREEZE 21-20: CTX10_FREEZE 19-18: CTX9_FREEZE 17-16: CTX8_FREEZE 15-14: CTX7_FREEZE 13-12: CTX6_FREEZE 11-10: CTX5_FREEZE 9-8: CTX4_FREEZE 7-6: CTX3_FREEZE 5-4: CTX2_FREEZE 3-2: CTX1_FREEZE 1-0: CTX0_FREEZE	CTX Freeze Controls the RW properties of this field and the n region context registers (CTXn_xxxx).

41.7.2.58 IPED Context Control 1 (IPEDCTXCTRL1)

Offset

Register	Offset
IPEDCTXCTRL1	72Ch

Function

Freezes the IPEDCTX*n*START and IPEDCTX*n*END. If the bit1 and bit0 CTXn_FREEZE field are different and CTXn_FREEZE1 field of IPEDCTXCTRL1 is 2'b10, then the CTXn_FREEZE field is writable.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
XSPI0	IPEDCTXCTRL1	—
XSPI1	IPEDCTXCTRL1	—
XSPI2	—	IPEDCTXCTRL1

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTX15_FREEZE		CTX14_FREEZE		CTX13_FREEZE		CTX12_FREEZE		CTX11_FREEZE		CTX10_FREEZE		CTX9_FREEZE		CTX8_FREEZE	
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTX7_FREEZE		CTX6_FREEZE		CTX5_FREEZE		CTX4_FREEZE		CTX3_FREEZE		CTX2_FREEZE		CTX1_FREEZE		CTX0_FREEZE	
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Fields

Field	Function
31-30: CTX15_FREEZE 29-28: CTX14_FREEZE	CTX Freeze Controls the RW properties of this field and the n region context registers (CTXn_xxxx).

Table continues on the next page...

Field	Function
27-26: CTX13_FREEZE 25-24: CTX12_FREEZE 23-22: CTX11_FREEZE 21-20: CTX10_FREEZE 19-18: CTX9_FREEZE 17-16: CTX8_FREEZE 15-14: CTX7_FREEZE 13-12: CTX6_FREEZE 11-10: CTX5_FREEZE 9-8: CTX4_FREEZE 7-6: CTX3_FREEZE 5-4: CTX2_FREEZE 3-2: CTX1_FREEZE 1-0: CTX0_FREEZE	

41.7.2.59 Flash Region Word 0 – Start Address (FRAD0_WORD0 - FRAD7_WORD0)

Offset

For n = 0 to 7:

Register	Offset
FRADn_WORD0	800h + (n × 20h)

Function

Specifies the starting address of the flash memory region. Only privilege initiators can write to this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STARTADR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 STARTADR	Start Address Specifies the most-significant 16 bits of the starting address (64-KB alignment) of the corresponding memory region.
15-0 —	Reserved

41.7.2.60 Flash Region Word 1 – End Address (FRAD0_WORD1 - FRAD7_WORD1)

Offset

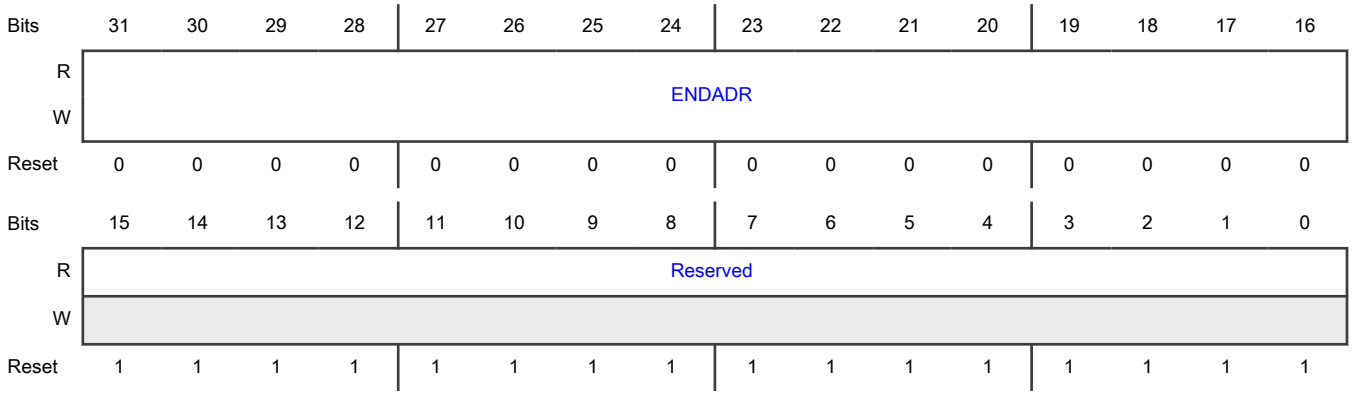
For n = 0 to 7:

Register	Offset
FRADn_WORD1	804h + (n × 20h)

Function

Specifies the ending address of the flash memory region. Only privilege initiators can write to this register.

Diagram



Fields

Field	Function
31-16 ENDADR	End Address Specifies the most-significant 16 bits of the inclusive ending address (64-KB alignment) of the corresponding memory region.
15-0 —	Reserved

41.7.2.61 Flash Region Word 2 – Privileges (FRAD0_WORD2 - FRAD7_WORD2)

Offset

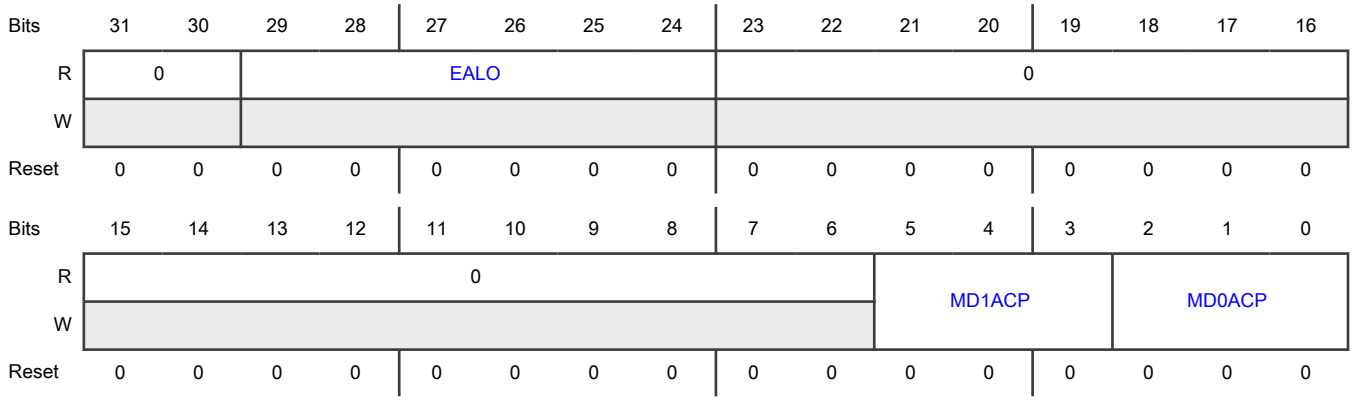
For n = 0 to 7:

Register	Offset
FRADn_WORD2	808h + (n × 20h)

Function

Specifies the access restrictions for the initiator domain and the exclusive access lock owner for the flash memory region. Only privilege initiators can write to this register.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 EALO	Exclusive Access Lock Owner If FRAD*n*_WORD3[EAL] = 11b, indicates the initiator ID that owns the exclusive access lock.
23-6 —	Reserved
5-3: MD1ACP 2-0: MD0ACP	Initiator Domain Access Control Policy Specifies the access restrictions for the respective initiator domain/TGn corresponding to this FRAD region. The secure and privilege attributes of the current transaction help to determine the access permissions. Read access is not restricted. You cannot write to this field in any of the following conditions exist: • The MDAD registers are not valid. • FRAD*n* is locked (FRAD*n*_WORD3[LOCK] = 01b). If you lock the descriptor except for initiator write (write 10b to FRADn_WORD3[LOCK]), then you can write to this field only if the initiator ID matches the ID check of the respective MDAD queue. If the MDAD is not valid, then the respective MD*n*ACP register is read-only.

Table 404. Field value mapping

Policy	Secure privilege	Secure user	Non secure privilege	Non secure user
0xxb	Read	Read	Read	Read
100b	Read/Write	Read/Write	Read	Read
101b	Read/Write	Read	Read	Read
110b	Read/Write	Read/Write	Read/Write	Read/Write
111b	Read/Write	Read	Read/Write	Read

Field	Function
	NOTE You must write 101b for a FRAD that has LUT access enabled for all atomic instructions. See LUT access enable.

41.7.2.62 Flash Region Word 3 – Lock Control (FRAD0_WORD3 - FRAD7_WORD3)

Offset

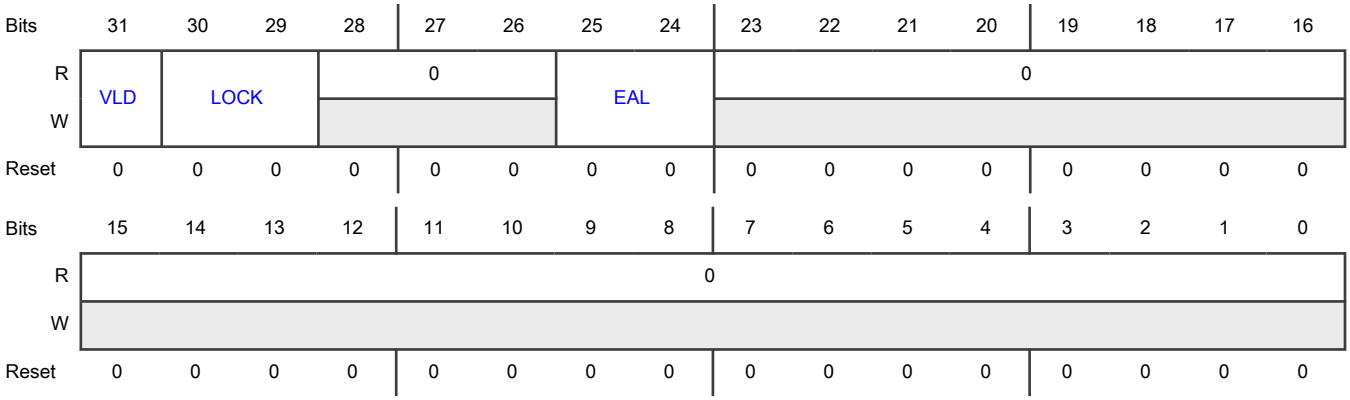
For n = 0 to 7:

Register	Offset
FRADn_WORD3	80Ch + (n × 20h)

Function

Configures exclusive access lock operation. Only privilege initiators can write to this register.

Diagram



Fields

Field	Function
31 VLD	Valid Specifies whether the FRAD descriptor for the flash region is valid. You cannot write this field in the same cycle that you write 10b or 11b to EAL. **NOTE** Write to this field only after you program all of the following: • FRAD*n*_WORD0 • FRAD*n*_WORD1 • FRAD*n*_WORD2

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Invalid 1b - Valid
30-29 LOCK	Descriptor Lock Locks the flash region descriptors (FRAD registers) from accidental writes. Only secure or privileged initiators can lock descriptors. Lock functionality does not affect EAL. 00b - Unlocks. Any initiator can write to the descriptors. 01b - Locks until hard reset. Descriptors are read-only. 10b - Locks except for initiator. Descriptors are read-only. The initiator whose ID matches MID check of the respective target group MDAD can write to the MDnACP fields. If the target group MDAD is not valid, then MDnACP fields are also read-only. 11b - Locks. Descriptors are read-only.
28-26 —	Reserved
25-24 EAL	Exclusive Access Lock Specifies the write lock behavior for the FRAD region based on the initiator domain access control policy (FRAD0_WORD2[MD*n*ACP]). 00b - No lock. Write permission is available for all initiators based on their MD*n*ACP evaluation. 01b - No action 10b - Enables lock. Write permissions revoked for all domains. You can change this field only to 11b; you cannot change it directly to 00b. 11b - Enables exclusive access lock. The initiator with the initiator ID specified as the exclusive access lock owner (FRAD*n*_WORD3[EAL]) has exclusive [read-]write permission based on its MD*n*ACP evaluation. Write access is locked for all other initiators. An initiator can write 11b to this field only if the above condition is met. When it does write 11b to this field, XSPI stores its ID as the exclusive access lock owner (EAL). If the initiator does not satisfy this check, XSPI does not generate a transfer error.
23-0 —	Reserved

41.7.2.63 Flash Region Word 4 – Compare Address Status (FRAD0_WORD4 - FRAD7_WORD4)

Offset

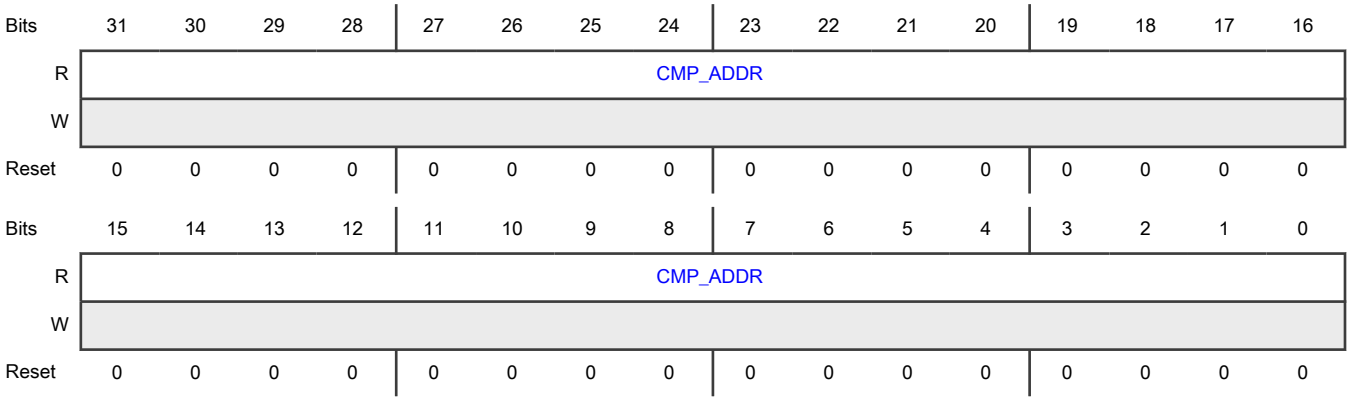
For n = 0 to 7:

Register	Offset
FRADn_WORD4	810h + (n × 20h)

Function

Indicates the 32-bit start address of the last transaction that lies inside the address range of FRAD*n*. Any write to this register generates a bus transfer error.

Diagram



Fields

Field	Function
31-0 CMP_ADDR	Capture Address Indicates the 32-bit start address of the last transaction that lies inside the address range of the FRAD.

41.7.2.64 Flash Region Word 5 – Compare Status Data (FRAD0_WORD5 - FRAD7_WORD5)

Offset

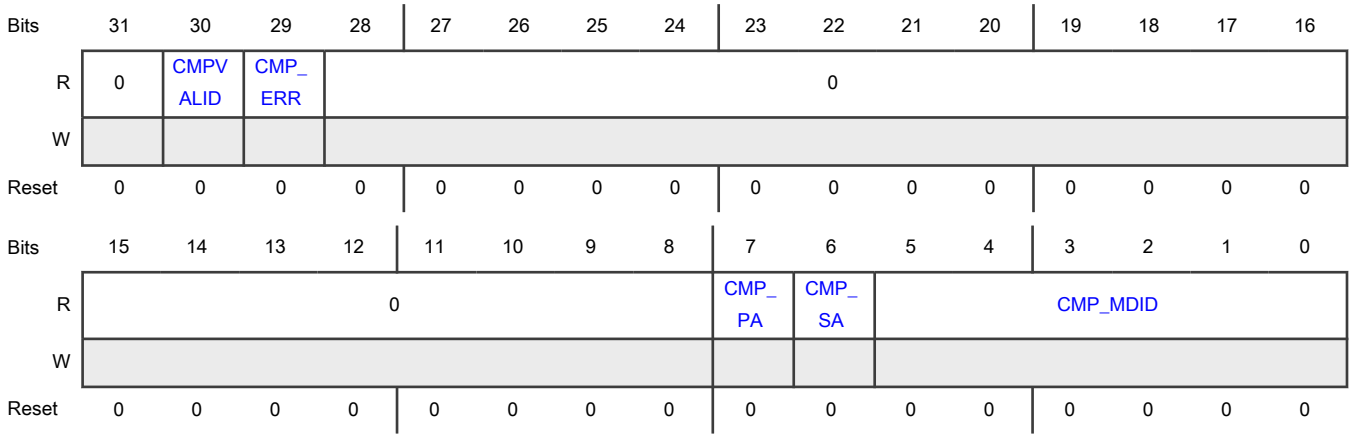
For n = 0 to 7:

Register	Offset
FRADn_WORD5	814h + (n × 20h)

Function

Indicates various aspects of FRAD operation. Any write to this register generates a bus transfer error.

Diagram



Fields

Field	Function
31 —	Reserved
30 CMPVALID	Comparison Valid Indicates availability of the result or status of a flash region-specific comparison check. 0b - Not available 1b - Available
29 CMP_ERR	Comparison Error Indicates the error status (based on MDxACP permissions or exclusive access lock as programmed in FRAD) of the flash region specific comparison check for the last transaction. You clear this error by writing 1 to the corresponding error status field (ERRSTAT[FRAD*n*ACC]). This field updates only after you clear the corresponding error field. 0b - No error 1b - Error
28-8 —	Reserved
7 CMP_PA	Captured Privilege Attribute Indicates the capture Privilege attribute of last transaction that passed the address check for the FRAD. 0b - Non-privileged 1b - Privileged
6 CMP_SA	Captured Secure Attribute Indicates the captured Secure attribute of the last transaction that passed the address check for the FRAD.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Non-secure 1b - Secure
5-0 CMP_MDID	Captured Initiator Value Indicates the captured initiator ID value of the last transaction that passed the address check for the FRAD.

41.7.2.65 SFP Arbitration Lock Timeout Counter (SFP_ARB_TIMEOUT)

Offset

Register	Offset
SFP_ARB_TIMEOUT	8FCh

Function

Specifies the time in clock cycles before an unreleased arbitration triggers a timeout error. Only the initiator with ID 011111b can access this register.

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFP_ARB_TOC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFP_ARB_TOC															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-0 SFP_ARB_TOC	SFP Arbitration Timeout Value Specifies the time in IPS clock cycles before an unreleased arbitration triggers a timeout error (ARB_LOCK_TO = 1). See Arbitration lock.

41.7.2.66 Target Group Initiator Domain Access Descriptor (TG0MDAD - TG1MDAD)

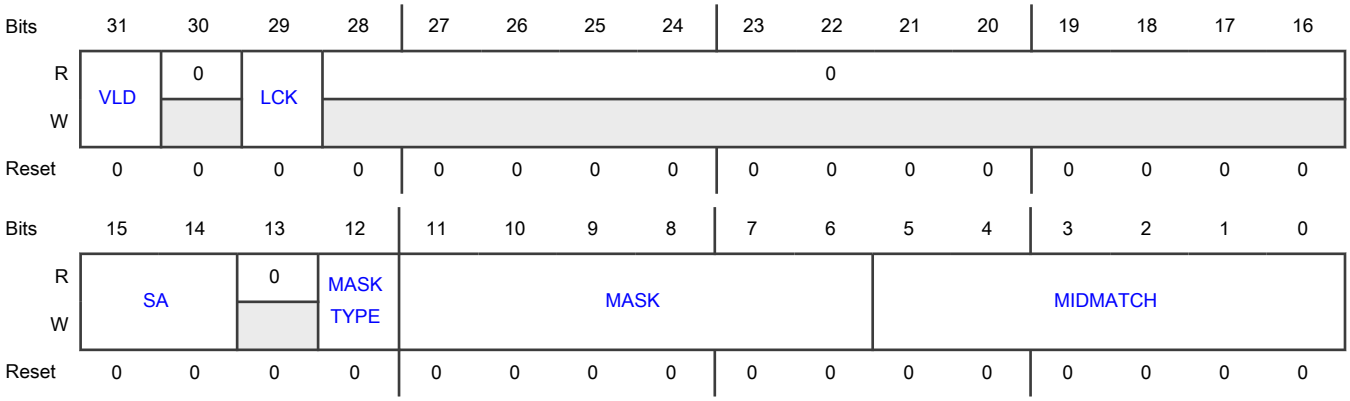
Offset

Register	Offset
TG0MDAD	900h
TG1MDAD	910h

Function

Specifies characteristics of initiator domain access. See MDAD checks. Only privilege initiators can write to this register.

Diagram



Fields

Field	Function
31 VLD	Valid Specifies whether the MDAD for the target group is valid. 0b - Not valid 1b - Valid
30 —	Reserved
29 LCK	Descriptor Lock Locks the MDAD. When locked, the descriptor is read-only. When you lock the descriptor, it remains locked until hard reset. 0b - No action 1b - Locks
28-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-14 SA	Secure Attribute Specifies the secure attribute selection criteria for entry into the descriptor queue. 00b - Reserved 01b - Non-secure only 10b - Secure only 11b - Both secure and non-secure
13 —	Reserved
12 MASKTYPE	Mask Type Specifies the mask type. 0b - AND 1b - OR
11-6 MASK	Mask Specifies the 6-bit mask value for the ID Match comparison.
5-0 MIDMATCH	Initiator ID Reference Specifies the reference value of the initiator ID for ID match comparison.

41.7.2.67 Target Group SFAR Address (TGSFAR)

Offset

Register	Offset
TGSFAR	904h

Function

Indicates the flash memory start address of the transaction in the target group queue.

NOTE

Only an initiator that passes the security checks in TG*n*MDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFARADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SFARADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SFARADDR	SFAR Address Indicates the flash memory start address of the transaction in the target group queue.

41.7.2.68 Target Group SFAR Status (TGSFARS)

Offset

Register	Offset
TGSFARS	908h

Function

Indicates the status of the target group SFAR.

NOTE

Only an initiator that passes the security checks in TGnMDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	ERR	CLR	0												
W			W1C													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PA	0	SA	0						TG_MID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VLD	Valid Indicates whether the SFAR address for the target group is valid and the queue is busy. XSPI does not set this bit if ERR is 1. 0b - Not valid 1b - Valid
30 ERR	Error Indicates whether the SFAR address stored by an initiator has the required access attributes for this target group descriptor. 0b - With required attributes 1b - Without required attributes
29 CLR	Clear Clears TGSFAR and TGSFARS. When you write 1 to this field, if there is a valid transaction present in the MDAD queue, XSPI updates TGSFAR and TGSFARS based on that transaction. Otherwise, XSPI clears both registers. Only a secure initiator (TG*n*MDAD[SA] = 10b or 11b) can write to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears
28-13 —	Reserved
12 PA	Privileged Attribute Indicates the privilege attribute of the initiator that wrote the SFAR register. 0b - Not privileged 1b - Privileged
11 —	Reserved
10 SA	Secure Attribute Indicates the secure attribute of the initiator that wrote to the SFAR register.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Non-secure 1b - Secure
9-6 —	Reserved
5-0 TG_MID	Transaction Initiator ID Indicates the initiator-ID of a transaction that programmed the SFAR registers with required attributes.

41.7.2.69 Target Group IP Configuration Status (TGIPCRS)

Offset

Register	Offset
TGIPCRS	90Ch

Function

Indicates the status of the target group IPCR.

NOTE

Only an initiator that passes the security checks in TG*n*MDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	ERR		CLR	0			0	0	ARB_UNL...	ARB_LOCK	0	SEQID			
W				W1C												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDATSZ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Valid

Table continues on the next page...

Table continued from the previous page...

Field	Function
VLD	Indicates whether the IPCR IDATZ, and SEQID stored in this target group queue are valid and the queue is locked. XSPI does not set this bit if ERR is 1. 0b - Not valid 1b - Valid and queue is locked
30-29 ERR	Error Indicates whether the initiator storing the IPCR has the required access attributes with respect to the target group MDAD. 00b - Required attributes 01b - IPCR DATZ without required attributes 10b - IPCR SEQID without required attributes 11b - IPCR IDATSZ and SEQID without required attributes
28 CLR	Clear Clears this register. Only a secure initiator (TG*n*MDAD[SA] = 10b or 11b) can write to this field. When you write 1 to this field, if there is a valid transaction present in the MDAD queue, XSPI updates IPCRS based on that transaction. Otherwise, XSPI clears the register (writes 0000_0000h to it). It does not clear the register if the secure attribute is non-secure (TG*n*MDAD[SA] = 00b). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears
27-25 —	Reserved
24 —	Reserved
23 —	Reserved
22 ARB_UNLOCK	Arbitration Unlock Indicates if the target queue has requested arbitration unlock. 0b - Not requested

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Requested
21 ARB_LOCK	Arbitration Lock Indicates if the target queue has requested arbitration lock. 0b - Not requested 1b - Requested
20 —	Reserved
19-16 SEQID	SEQID Value Indicates the sequence ID for the target group queue.
15-0 IDATSZ	IP Data Transfer Size Indicates the IP data transfer size in bytes of the target group queue.

41.7.2.70 Initiator Global Configuration (MGC)

Offset

Register	Offset
MGC	920h

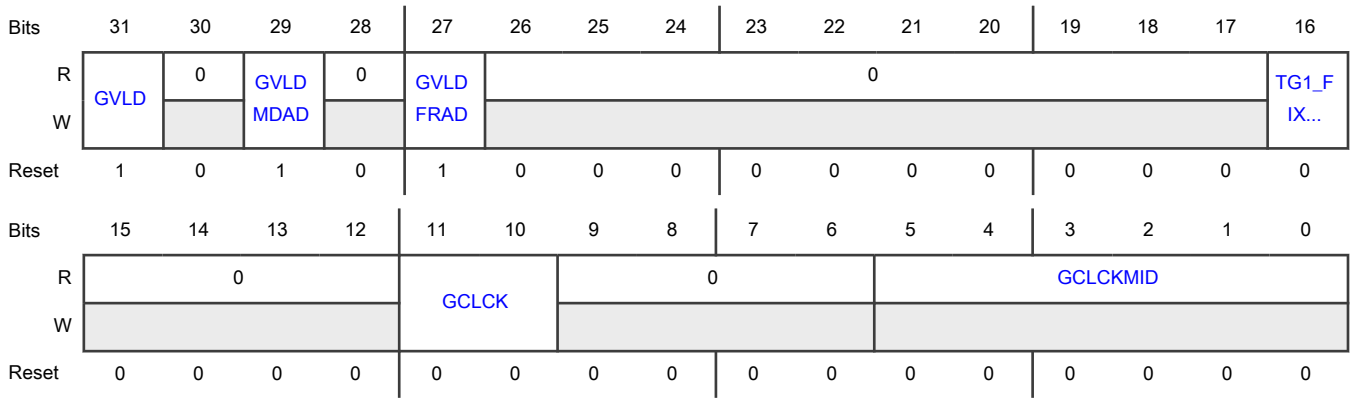
Function

Specifies several aspects of initiator domain behavior, and indicates the global configuration lock owner.

NOTE

Only an initiator with an ID equal to 000000b can access this register. For MGC, MRC, MTO, and SFP_ARB_TIMEOUT, XSPI uses the initiator ID for the access check. For transactions inside MDAD and FRAD, it uses the domain ID for the access check.

Diagram



Fields

Field	Function
31 GVLD	Global Valid Access Control Enables MDADs globally. 0b - Disables. XSPI does not perform a descriptor comparison check, and allows all transactions. 1b - Enables
30 —	Reserved
29 GVLDMDAD	Global Valid MDAD Enables MDADs globally. 0b - Disables 1b - Enables
28 —	Reserved
27 GVLDFRAD	Global Valid FRAD Enables FRADs globally. 0b - Disables 1b - Enables
26-17 —	Reserved
16 TG1_FIX_PRIO	Target Group Queue 1 Fixed Priority Enable Enables fixed priority for the target group 1 queue. All other values - TG1 always has fixed priority regardless of this value.
15-12 —	Reserved
11-10 GCLCK	Global Configuration Lock Locks write access to the following registers: • MGC • MRC • MTO • SFP_ARB_TIMEOUT • SFP_LUT_EN*n*

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• LUT*n* • BFGENCR 00b - Unlocks. You can write the registers based on the individual register access attribute. 01b - Locks. The registers remain read-only until the next hard reset. 10b - Locks. Only the global configuration lock owner can write to the registers. 11b - Locks. The registers remain read-only until the next hard reset.
9-6 —	Reserved
5-0 GCLCKMID	Global Configuration Lock Owner Indicates the 6-bit ID of the initiator that owns the global configuration lock. See GCLCK. Sideband signals provide the input value of the global initiator ID.

41.7.2.71 Initiator Read Command (MRC)

Offset

Register	Offset
MRC	924h

Function

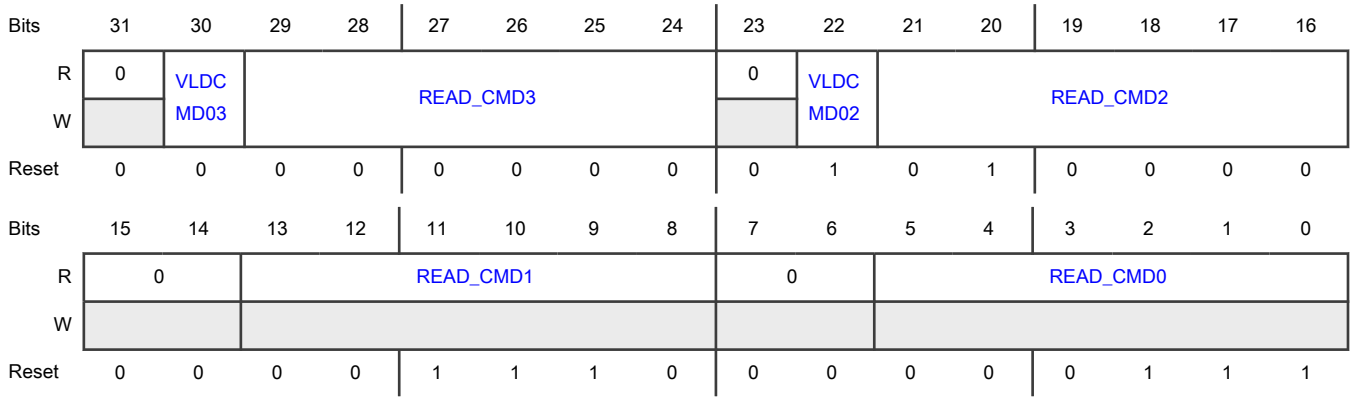
Indicates the encoding for the existing read instruction.

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

NOTE

- Only an initiator with an ID equal to 000000b can access this register.
- In the event of a JMP_TO_SEQ instruction in LUT*n*, SFP parses only the first sequence; it does not parse the second sequence for Read commands.

Diagram



Fields

Field	Function
31 —	Reserved
30 VLDCMD03	Valid Command 3 Indicates if READ_CMD3 is valid. 0b - Not valid 1b - Valid
29-24 READ_CMD3	Read Command 3 Programs the additional READ Instruction/Command encoding information, if applicable
23 —	Reserved
22 VLDCMD02	Valid Command 2 Indicates if READ_CMD2 is valid. 0b - Not valid 1b - Valid
21-16 READ_CMD2	Read Command 2 Specifies additional encoding for the READ instruction or command, if applicable.
15-14 —	Reserved
13-8 READ_CMD1	Read Command 1 Indicates the encoding for the existing instruction - READ_DDR command.
7-6	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
5-0 READ_CMD0	Read Command 0 Indicates the encoding for the existing instruction-READ command.

41.7.2.72 Initiator Timeout (MTO)

Offset

Register	Offset
MTO	928h

Function

Specifies the maximum time in clock cycles for XSPI to wait for an ongoing write or read command to complete before terminating the command.

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

NOTE

Only an initiator with an ID equal to 000000b can access this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFP_ACC_TO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFP_ACC_TO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-0 SFP_ACC_TO	SFP Access Timeout Specifies the maximum time in IPS clock cycles for XSPI to wait for an ongoing write or read command to complete before terminating the command. The timeout counter begins when any target group queue

Table continues on the next page...

Field	Function
	wins the access arbitration and XSPI is idle (FSMSTAT[STATE] != 00b). If the timeout counter expires and generates an interrupt (if enabled), XSPI clears the SFP queue. See ERRSTAT[TO_ERR]. **NOTE** MTO should be programmed with a large value such that MTO cannot occur when the data is getting transmitted to flash.

41.7.2.73 Flash Sequence Request (FLSEQREQ)

Offset

Register	Offset
FLSEQREQ	92Ch

Function

Indicates information about the last executed flash memory sequence request.

NOTE

This register retains its value if a timeout error occurs (FLSEQREQ[TIMEOUT]. XSPI updates it with a new value when you clear the timeout error (write 1 to ERRSTAT[TO_ERR]).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0		0	TIMEOUT	0		0	0	CMD	0		SEQID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	FRAD			0	ARB_LOCK	PA	SA	0	REQ_TG	REQ_MID					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VLD	Valid Indicates whether the flash memory sequence request status is valid. 0b - Not valid 1b - Valid

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-29 —	Reserved
28 —	Reserved
27 TIMEOUT	Timeout Error Status Indicates the timeout error status of the last executed flash memory sequence request. 0b - No error 1b - Error
26-25 —	Reserved
24 —	Reserved
23 —	Reserved
22 CMD	Instruction Type Indicates the instruction type of the last executed flash memory sequence request. 0b - Read 1b - Non-read
21-20 —	Reserved
19-16 SEQID	Sequence ID Indicates the sequence ID of the last executed flash memory sequence request.
15 —	Reserved
14-12 FRAD	Flash Region Descriptor Number Indicates the number of the flash region (FRAD) for the last executed flash memory sequence request. XSPI does not update this value for read instructions.
11 —	Reserved
10	Arbitration Lock

Table continues on the next page...

Table continued from the previous page...

Field	Function
ARB_LOCK	Indicates the arbitration lock status of the last executed flash memory sequence request. 0b - Not locked 1b - Locked
9 PA	Privilege Attribute Indicates the privilege attribute of the last executed flash memory sequence request. 0b - Non-privilege 1b - Privilege
8 SA	Secure Attribute Indicates the secure attribute of the last executed flash memory sequence request. 0b - Non-secure 1b - Secure
7 —	Reserved
6 REQ_TG	FlashSeq Request Target Group Indicates the target group queue that passed the last executed transaction. 0b - TG0 1b - TG1
5-0 REQ_MID	Flash Sequence Request Initiator ID Indicates the initiator ID of the last executed flash memory sequence request.

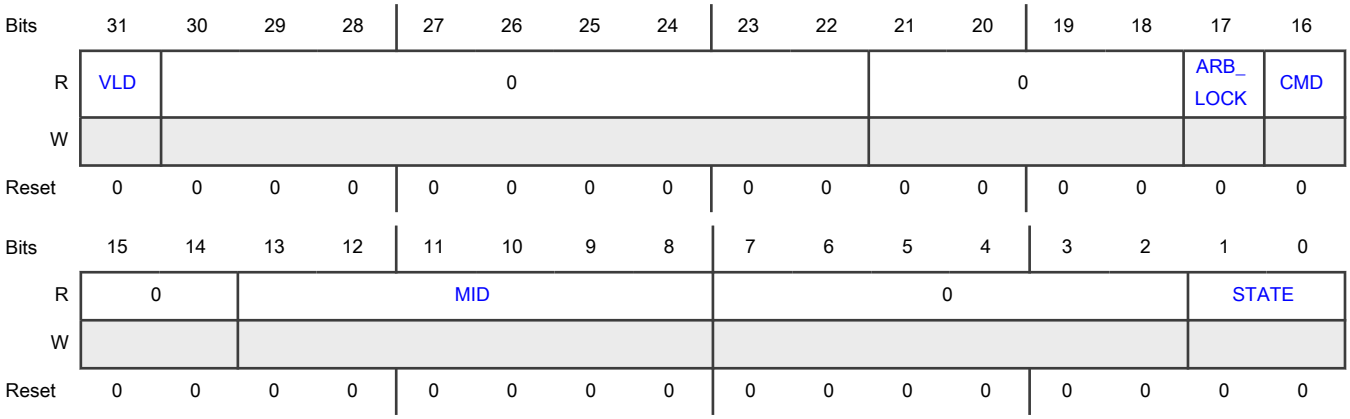
41.7.2.74 FSM Status (FSMSTAT)

Offset

Register	Offset
FSMSTAT	930h

Function
Indicates the finite state machine status.

Diagram



Fields

Field	Function
31 VLD	Valid Indicates whether the FSM status is valid. 0b - Not valid. No IPS transfer is queued. 1b - Valid. The IPS transfer is granted arbitration or execution.
30-22 —	Reserved
21-18 —	Reserved
17 ARB_LOCK	Arbitration Lock Indicates the status of arbitration lock for the current request. 0b - Not locked 1b - Locked
16 CMD	Command Indicates the instruction type of the currently initiated flash memory sequence transaction. 0b - Read instruction sequence 1b - Non-read instruction sequence
15-14 —	Reserved
13-8 MID	Initiator ID Indicates the initiator ID of the currently initiated flash memory sequence transaction.
7-2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
1-0 STATE	FSM State Indicates the current state of the FSM. 00b - Transaction granted. Transaction is granted but XSPI is busy with one of the following: an ongoing previous DMA transaction; RDBFL contains residue data; or an interrupt is pending. 01b - TBDR lock open. IPS initiator can write to TBDR. 10b - Write transfer triggered. XSPI has received the SEQID. 11b - Read transfer triggered. XSPI has received the SEQID.

41.7.2.75 IPS Error (IPSERROR)

Offset

Register	Offset
IPSERROR	934h

Function

Indicates the error status of an IPS write on the SFAR or IPCR registers if they are not qualified for any of the target group queues .

Only privilege initiators can write to this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		CLR		0				0				0			
W			W1C													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FRADPROG	MDADPROG	TG1MID	TG0MID	TG1SEC	TG0SEC	TG1LCK	TG0LCK	0		MID					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
29 CLR	Clear Clears the status of this register and the IPS_ERR flag. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears
28-25 —	Reserved
24-22 —	Reserved
21-19 —	Reserved
18-16 —	Reserved
15 FRADPROG	FRAD Program Status Indicates the FRAD programming status. 0b - Some or all FRADs programmed 1b - No FRADs programmed
14 MDADPROG	TG/MDA Descriptor Program Status Indicates the target group or MDA descriptor programming status. 0b - One or both target group descriptors programmed 1b - None of the target group descriptors are programmed or valid
13-12 TGnMID	TGn Initiator-ID Status Indicates the status of the initiator-ID check. 0b - Passed 1b - Failed

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-10 TGnSEC	TG Security Status Indicates the security status of the target group. 0b - Security attribute check passed 1b - Security attribute check failed
9-8 TGnLCK	TGn Lock Indicates the lock status of the TG. 0b - SEQID not written yet 1b - SEQID written and queue locked
7-6 —	Reserved
5-0 MID	IPS Initiator ID Indicates the IPS initiator ID for the transaction that generated this IPS transfer error.

41.7.2.76 Error Status (ERRSTAT)

Offset

Register	Offset
ERRSTAT	938h

Function

Provides error and status information about the current transaction.

Only privilege initiators can write to this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	LOCK_ERR	ARB_LOC...	ARB_WIN	0				0							
W		W1C	W1C	W1C												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TO_ERR	TG1IPCR	TG0IPCR	TG1SFAR	TG0SFAR	IPS_ERR	FRAD7ACC	FRAD6ACC	FRAD5ACC	FRAD4ACC	FRAD3ACC	FRAD2ACC	FRAD1ACC	FRAD0ACC	FRADMTCH
W		W1C						W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 LOCK_ERR	Lock Register Error Indicates an attempt to write a locked register. See Register lock access error. XSPI captures the new error address only after you clear the error by writing 1 to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears
29 ARB_LOCK_TO	Arbitration Lock Timeout Error Indicates that arbitration lock has timed out. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears flag
28 ARB_WIN	Arbitration Win Event Status Indicates that the target group queue request was granted access (it won arbitration) and XSPI has generated an interrupt. Clear this flag by writing 1 to it. **NOTE** If the corresponding interrupt is not enabled (INT_EN[ARB_WIN_IE] = 0), this field has no meaning. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Request not granted or the interrupt is already cleared 1b - Request granted and interrupt generated When writing 0b - No action 1b - Clears interrupt
27-24 —	Reserved
23-16 —	Reserved
15 —	Reserved
14 TO_ERR	Timeout Error Indicates that a flash memory transaction has triggered a timeout error and has been terminated by SFP. Clear this error by writing 1 to the field. XSPI keeps the error indication latched in FLSEQREQ[TIMEOUT] until you clear this error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears
13-12 TGnIPCR	TG IPCR Error Indicates that an IPCR write has triggered an error while being written to the corresponding target group queue. Clear this field by writing 1 to TGIPCRS[CLR]. 0b - No error 1b - Error
11-10 TGnSFAR	TG SFAR Error Indicates that an SFAR write triggered an error while being written to the corresponding target group queue. Clear this field by writing 1 to TGSFARS[CLR]. 0b - No error 1b - Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 IPS_ERR	IPS Error Indicates that a common error has occurred and XSPI has generated an IPS bus transfer error. Find the error details in IPSERROR. Clear this field by writing 1 to IPSERROR[CLR]. 0b - No error 1b - Error
8-1 FRADnACC	FRAD Access Error Indicates that the transaction address lies within the address range of the corresponding FRAD but it does not pass the access permission checks for the FRAD or the FRAD is under exclusive lock (FRAD*n*_WORD3[EAL] = 10b or 11b) by another initiator. It may also indicate that the transaction address qualifies for multiple FRAD regions. Clear this flag and CMP_ERR by writing 1 to this field. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error. Transaction triggered an error and cleared the target queue. When writing 0b - No action 1b - Clears
0 FRADMTCH	No FRAD Match Error Indicates that the transaction address does not lie within the address range of any FRAD. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No action 1b - Clears

41.7.2.77 Interrupt Enable (INT_EN)

Offset

Register	Offset
INT_EN	93Ch

Function

Enables the various SFP interrupts.

Only privilege initiators can write to this register.

NOTE

If you enable the queue-specific error interrupts (TG0SFAR, TG1SFAR, TG0IPCR, or TG1IPCR), ensure that the core executing the corresponding interrupt handler can clear TGSFARS and TGIPCRS (the core's security attribute matches the queue). If there is a mismatch in the security attribute (TG*n*MDAD[SA]), you must change the security attribute of that queue to match the core security attribute. The core can clear the error and then you can revert the change to TG*n*MDAD[SA].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LCK	LCK_ERR...	ARB_TO_...	ARB_WIN...	0	0										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TO_ERR	TG1IPCR	TG0IPCR	TG1SFAR	TG0SFAR	IPS_ERR	FRAD7ACC	FRAD6ACC	FRAD5ACC	FRAD4ACC	FRAD3ACC	FRAD2ACC	FRAD1ACC	FRAD0ACC	FRADMTCH
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LCK	Lock Disables write access to INT_EN. When you write 1 to this field, it remains 1 until the next hard reset. 0b - Unlocks 1b - Locks
30 LCK_ERR_IE	Lock Register Write Error Interrupt Enable Enables the interrupt generated by an attempt to write a locked register. See ERRSTAT[LOCK_ERR]. 0b - Disables 1b - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Function
29 ARB_TO_IE	Arbitration Lock Timeout Error Interrupt Enable Enables the interrupt generated when a flash memory transaction triggers a timeout error and has been terminated by SFP. See ERRSTAT[ARB_LOCK_TO]. 0b - Disables 1b - Enables
28 ARB_WIN_IE	Arbitration Win Interrupt Enable Enables the interrupt generated when the target group queue request is granted access (it won arbitration). See ERRSTAT[ARB_WIN]. 0b - Disables 1b - Enables
27 —	Reserved
26-15 —	Reserved
14 TO_ERR	Timeout Error Interrupt Enable Enables the interrupt generated when a flash memory transaction triggers a timeout error and has been terminated by SFP. See ERRSTAT[TO_ERR]. 0b - Disables 1b - Enables
13-12 TGnIPCR	TGn IPCR Error Interrupt Enable Enables the interrupt generated when an IPCR write triggers an error while being written to the corresponding target group queue. See ERRSTAT[TG*n*IPCR]. 0b - Disables 1b - Enables
11-10 TGnSFAR	TG SFAR Error Interrupt Enable Enables the interrupt generated when an SFAR write triggers an error while being written to the corresponding target group queue. See ERRSTAT[TG*n*SFAR]. 0b - Disables 1b - Enables
9 IPS_ERR	IPS Error Interrupt Enable Enables the interrupt generated when a common error has occurred and XSPI has generated an IPS bus transfer error. See ERRSTAT[IPS_ERR]. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
8-1 FRADnACC	FRAD Access Error Interrupt Enable Enables the interrupt generated when the transaction address lies within the address range of the corresponding FRAD but it does not pass the access permission checks for the FRAD or the FRAD is under exclusive lock. See ERRSTAT[FRAD*n*ACC]. 0b - Disables 1b - Enables
0 FRADMTCH	No FRAD Match Error Interrupt Enable Enables the interrupt generated when the transaction address does not lie within the address range of any FRAD. See ERRSTAT[FRADMTCH]. 0b - Disables 1b - Enables

41.7.2.78 IP Configuration (SFP_TG_IPCR)

Offset

Register	Offset
SFP_TG_IPCR	958h

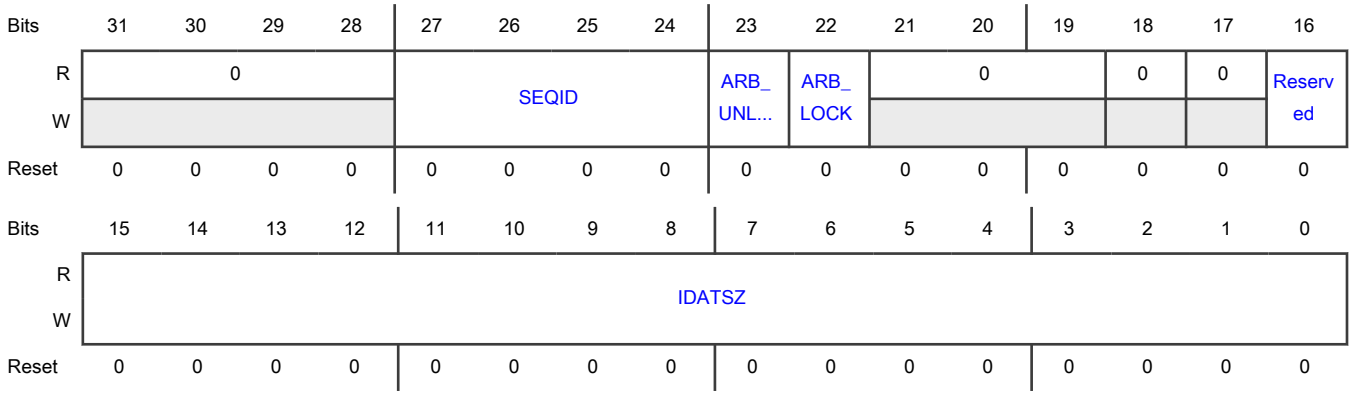
Function

Configures XSPI operation for an IP-initiated command, triggered by writing to SEQID in this register. If the SEQID write is successful, including passing the required SFP checks, XSPI begins the command sequence pointed to by SEQID. This register applies to transactions initiated for TG0.

NOTE

- You should write 1 to only one of ARB_LOCK or ARB_UNLOCK at a time. Writing 1 to both fields simultaneously inverts the lock status. In other words, if the arbitration is already locked, it becomes unlocked and vice-versa.
- If you enable MDAD and FRAD checks (see MGC), but none of the MDADs or FRADs are valid, then any write to this register generates a bus transfer error.
- Only an initiator that passes the security checks (TG*n*MDAD[SA]) can write this register, unless the MDAD is not valid, in which case any initiator can write to it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] is 00b.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-24 SEQID	Sequence ID Specifies the sequence index of the LUT and triggers a transaction on the serial flash memory interface. See Lookup table (LUT).
23 ARB_UNLOCK	Arbitration Unlock Unlocks arbitration. Writing 0 has no effect. XSPI unlocks the arbitration only if the transaction passes both MDAD and FRAD checks. To lock the arbitration, write 1 to ARB_LOCK. See Arbitration lock. 0b - No effect. Always reads 0. 1b - Unlocks
22 ARB_LOCK	Arbitration Lock Locks arbitration for the target queue receiving the transaction. Writing 0 has no effect. XSPI grants the arbitration lock only if the transaction passes both MDAD and FRAD checks. To unlock the arbitration, write 1 to ARB_UNLOCK. See Arbitration lock. 0b - No effect. Always reads 0. 1b - Locks
21-19 —	Reserved
18 —	Reserved
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 —	Reserved
15-0 IDATSZ	IP Data Transfer Size Specifies the data transfer size, in bytes, of the IP command.

41.7.2.79 Serial Flash Memory Address (SFP_TG_SFAR)

Offset

Register	Offset
SFP_TG_SFAR	95Ch

Function

Specifies the address for a memory transaction. XSPI automatically translates this address on the memory map to the address on the memory device. When XSPI is in 24-bit mode, it sends only bits 23–0 to memory. In 32-bit mode, it sends bits 27–0 and drives 31–28 to 0 when you specify 0 for the column address space (SFACR[CAS]). For example, in 24-bit mode, if SFACR[CAS] is 3 (0011b), then XSPI sends bits 26-3 to the memory as the page address. The total number of address bits requested by memory, for page and column address, must not exceed 32 bits.

For the mapping between the access mode and the SFAR content, see Serial flash memory address assignment. For information about command triggering and command execution, see Normal mode overview. You must ensure that the address you provide is in the valid memory address range, as described in Serial flash memory address assignment.

Use this register for transactions on TG0.

NOTE

- If you enable MDAD and FRAD checks (MGC) but none of the descriptors are valid, then any write on this register generates a bus transfer error.
- Only an initiator that passes the security checks (TG*n*MDAD[SA]) can write this register, unless the MDAD is not valid, in which case any initiator can write to it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] is 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SFADR	Serial Flash Memory Address Indicates the serial memory address for IP commands. The address is the address of the device without the base address.

41.7.2.80 LUT Access Enable (SFP_LUT_EN0 - SFP_LUT_EN1)

Offset

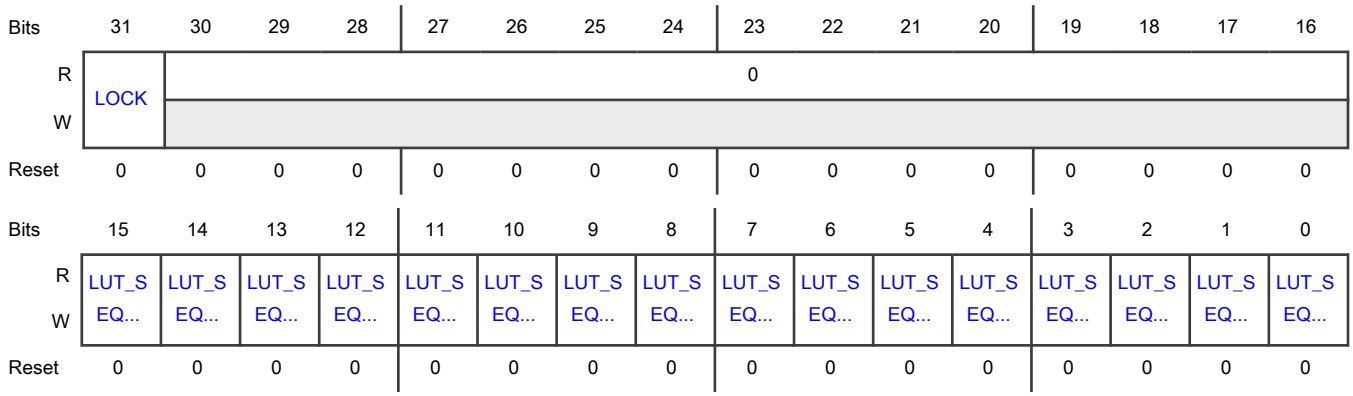
Register	Offset
SFP_LUT_EN0	960h
SFP_LUT_EN1	964h

Function

Controls access to the LUT sequence for the corresponding target queue. Only an initiator with ID equal to 6'd0 can access this register.

You can write to this register only if the global configuration lock is disabled (GCLCK = 00b).

Diagram



Fields

Field	Function
31 LOCK	Locks access to this register. When you write 1 to this field, it remains 1 until hard reset. 0b - No effect 1b - Locks
30-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-0 LUT_SEQn_EN	LUT Sequence Enable Enables valid LUT SEQID values to be used by the corresponding TG. See LUT access enable. **NOTE** This LUT sequence access check applies only to write transactions. 0b - Enables 1b - Disables

41.7.2.81 SFP Lock Error Address (SFP_LOCK_ERR_ADDR)

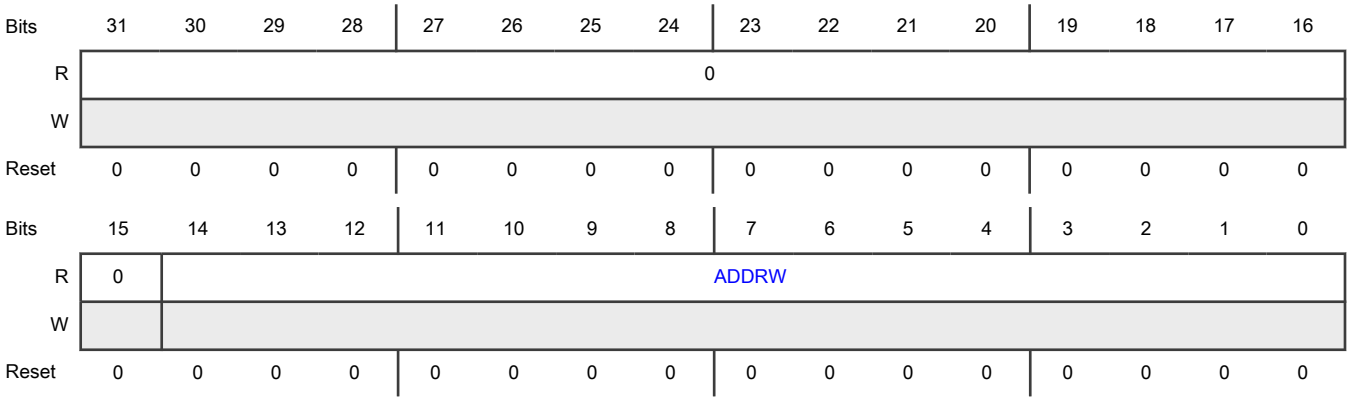
Offset

Register	Offset
SFP_LOCK_ERR_ADDR	A7Ch

Function

Latches the address of the register on which write access is attempted when the register write access is locked by any of the configuration bits.

Diagram



Fields

Field	Function
31-15 —	Reserved
14-0 ADDRW	Address Indicates the address that triggered the lock error. Only valid when there is a lock error (LOCK_ERR = 1).

41.7.2.82 IP Configuration (SFP_TG_SUB_1_IPCR)

Offset

Register	Offset
SFP_TG_SUB_1_IPCR	A80h

Function

Configures an IP-initiated command for the corresponding TG that can be triggered by writing to SEQID. If the write to SEQID is successful and it passes the required SFP checks, it initiates a new command to the external serial flash memory per the sequence pointed to by that field. See Normal mode overview.

NOTE

- You should write 1 to only one of ARB_LOCK and ARB_UNLOCK at a time. Writing 1 to both fields at the same time inverts the lock status. If arbitration was already locked, it unlocks and vice-versa.
- If you enable MDAD and FRAD checks MGC) but none of the descriptors are valid, then any write to this register generates a bus transfer error.
- Only an initiator that passes the security checks in TGnMDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				SEQID				ARB_UNL...	ARB_LOCK	0			0	0	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDATSZ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 SEQID	Points to a sequence in the LUT Specifies the sequence index of the LUT and triggers a transaction on the serial flash memory interface. See Lookup table (LUT).
23	Arbitration Unlock

Table continues on the next page...

Table continued from the previous page...

Field	Function
ARB_UNLOCK	Unlocks arbitration. Writing 0 has no effect. XSPI unlocks the arbitration only if the transaction passes both MDAD and FRAD checks. To lock the arbitration, write 1 to ARB_LOCK. 0b - No effect. Always reads 0. 1b - Unlocks
22 ARB_LOCK	Arbitration Lock Locks arbitration for the target queue receiving the transaction. Writing 0 has no effect. XSPI grants the arbitration lock only if the transaction passes both MDAD and FRAD checks. To unlock the arbitration, write 1 to ARB_UNLOCK. 0b - No effect. Always reads 0. 1b - Locks
21-19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15-0 IDATSZ	IP Data Transfer Size Indicates the IP data transfer size in bytes of the target group queue.

41.7.2.83 Serial Flash Memory Address (SFP_TG_SUB_1_SFAR)

Offset

Register	Offset
SFP_TG_SUB_1_SFAR	A84h

Function

Specifies the address for a memory transaction. XSPI automatically translates this address on the memory map to the address on the memory device. When XSPI is in 24-bit mode, it sends only bits 23–0 to memory. In 32-bit mode, it sends bits 27–0 and drives 31–28 to 0 when you specify 0 for the column address space (SFACR[CAS]). For example, in 24-bit mode, if SFACR[CAS] is 3 (0011b), then XSPI sends bits 26-3 to the memory as the page address. The total number of address bits requested by memory, for page and column address, must not exceed 32 bits.

For the mapping between the access mode and the SFAR content, see Serial flash memory address assignment. For information about command triggering and command execution, see Normal mode overview. You must ensure that the address you provide is in the valid memory address range, as described in Serial flash memory address assignment.

Use this register for transactions on the corresponding TG.

NOTE

- If you enable MDAD and FRAD checks (MGC) but none of the descriptors are valid, then any write on this register generates a bus transfer error.
- Only an initiator that passes the security checks (TG*n*MDAD[SA]) can write this register, unless the MDAD is not valid, in which case any initiator can write to it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] is 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SFADR	Serial flash memory address Indicates the serial memory address for IP commands. The address is the address of the device without the base address.

41.7.2.84 Target Group SFAR Address (TGSFAR_SUB_1)

Offset

Register	Offset
TGSFAR_SUB_1	A88h

Function

Specifies the flash memory start address of the transaction in the target group queue.

NOTE

Only an initiator that passes the security checks in TGnMDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFARADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SFARADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SFARADDR	SFAR Address Indicates the flash memory start address of the transaction in the target group queue.

41.7.2.85 Target Group SFAR Status (TGSFARS_SUB_1)

Offset

Register	Offset
TGSFARS_SUB_1	A8Ch

Function

Indicates the status of the target group SFAR.

NOTE

Only an initiator that passes the security checks in TGnMDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	ERR	CLR	0												
W			W1C													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PA	0	SA	0						TG_MID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VLD	Valid Indicates whether the SFAR Address for the target group is valid and the queue is busy. XSPI does not set this bit if ERR is 1. 0b - Not valid 1b - Valid
30 ERR	Error Indicates whether the SFAR address stored by an initiator has the required access attributes for this target group descriptor. 0b - With required attributes 1b - Without required attributes
29 CLR	Clear Clears TGSFAR_SUB_*n* and TGSFARS_SUB_*1*. **NOTE** Only a secure initiator (TG*n*MDAD[SA] = 10b or 11b) can write to this field. When you write 1 to this field, if there is a valid transaction present in the MDAD queue, XSPI updates TGSFAR and TGSFARS based on that transaction. Otherwise, XSPI clears both registers. It does not clear them if the secure attribute is non-secure (TG*n*MDAD[SA] = 00b). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears
28-13 —	Reserved
12 PA	Privileged Attribute Indicates the privilege attribute of the initiator that wrote the SFAR register. 0b - Non-privileged 1b - Privileged
11 —	Reserved
10	Secure Attribute

Table continues on the next page...

Table continued from the previous page...

Field	Function
SA	Indicates the secure attribute of the initiator that wrote to the SFAR register. 0b - Non-secure 1b - Secure
9-6 —	Reserved
5-0 TG_MID	Transaction Initiator ID Indicates the initiator-ID of a transaction that programmed the SFAR registers with required attributes.

41.7.2.86 Target Group n IP Configuration Status (TGIPCRS_SUB_1)

Offset

Register	Offset
TGIPCRS_SUB_1	A90h

Function

Indicates the status of the target group IPCR.

NOTE

Only an initiator that passes the security checks in TG*n*MDAD[SA] can read this register, unless the MDAD is not valid, in which case any initiator can read it. Otherwise, this register reads 0. It also reads 0 if TG*n*MDAD[SA] = 00b.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	ERR		CLR	0			0	0	ARB_ UNL...	ARB_ LOCK	0	SEQID			
W				W1C												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDATSZ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VLD	Valid Indicates whether the IPCR IDATZ, SEQID, and PAR stored in this target group queue are valid and the queue is locked. XSPI does not set this bit if ERR is 1. 0b - Not valid 1b - Valid and queue is locked
30-29 ERR	Error Indicates whether the initiator storing the IPCR has the required access attributes with respect to the target group MDAD. 00b - Required attributes 01b - IPCR DATZ without required attributes 10b - IPCR SEQID without required attributes 11b - IPCR DATZ and SEQID without required attributes
28 CLR	Clear Clears this register. **NOTE** Only a secure initiator (TG*n*MDAD[SA] = 10b or 11b) can write to this field. When you write 1 to this field, if there is a valid transaction present in the MDAD queue, XSPI updates IPCRS based on that transaction. Otherwise, XSPI clears the register (writes 0000_0000h to it). It does not clear the register if the secure attribute is non-secure (TG*n*MDAD[SA] = 00b). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Conveys no useful information When writing 0b - No action 1b - Clears
27-25 —	Reserved
24 —	Reserved
23 —	Reserved
22	Arbitration Unlock

Table continues on the next page...

Table continued from the previous page...

Field	Function
ARB_UNLOCK	Indicates if the target queue has requested arbitration unlock. 0b - Not requested 1b - Requested
21 ARB_LOCK	Arbitration Lock Indicates if the target queue has requested arbitration lock. 0b - Not requested 1b - Requested
20 —	Reserved
19-16 SEQID	SEQID Value Indicates the sequence ID for the target group queue.
15-0 IDATSZ	IP Data Transfer Size Indicates the IP data transfer size in bytes of the target group queue.

41.7.3 Serial flash memory address assignment

The serial flash memory address assignment can be modified by writing into Serial Flash Memory Top Address (SFA1AD) and Serial Flash Memory Top Address (SFA2AD) for device A.

Table 405. Serial flash memory address assignment

Parameter	Function	Access mode
XSPI_AMBA_BASE ((31:10) - 22 bits)	XSPI AHB base address First address of the serial flash memory device as presented to the XSPI controller. This might be the base of the serial flash memory in the system address map or it may be a remapping, for instance to 0h, performed by the system. (See the system address map file attached to this document)	
TOP_ADDR_MEMA1(TPADA1)	Top address for the external flash memory A1 (first device of the dual-die flash memory A, or the first of the two independent flash memories sharing the IOFA)	Any access to the address space between TOP_ADDR_MEMA1 and XSPI_AMBA_BASE is routed to serial flash memory A1.

Table continues on the next page...

Table 405. Serial flash memory address assignment (continued)

Parameter	Function	Access mode
TOP_ADDR_MEMA2(TPADA2)	Top address for the external flash memory A2 (second device of the dual-die flash memory A, or the second of the two independent flash memories sharing the IOFA)	Any access to the address space between TOP_ADDR_MEMA2 and TOP_ADDR_MEMA1 is routed to serial flash memory A2.

41.8 Flash memory mapped AMBA bus

XSPI_AMBA_BASE defines the address to be used as the start address of the serial flash memory device, as defined by the system memory map. Note that this may be a remapping of the physical address of the serial flash memory in the system. See the system address map for details.

Table 406. XSPI AMBA bus memory map

Address	Register name
XSPI_AMBA_BASE to (TOP_ADDR_MEMA2 - 1h)	• See Memory-mapped serial flash memory data—individual flash memory mode on flash memory A. • For information about byte ordering, see Table 335 and Table 336.

NOTE

Any read access to non-implemented addresses provides undefined results.

In individual flash memory modes, the 3/4 address bytes (as programmed in the instruction/operand in the sequence) available for the flash memory address are determined by SFADR[23:0] or SFADR[31:0] as provided in the table shown above.

41.8.1 AHB bus access read considerations

Note that all logic in the XSPI module implementing the AHB bus access is designed to read the content of an external serial flash memory device. Therefore, the following restrictions apply to the XSPI module with respect to accesses to the AHB bus:

- AHB bus read access types—NONSEQ and BUSY—are fully supported.
- AHB read access type—SEQ—is treated in the same way as NONSEQ. See Flash memory mapped AMBA bus for details.
- Early burst termination is not supported for AHB transactions.
- An AHB error response is provided on occurrence of either of the following:
  - Data learning failure
  - If FR[AIBSEF] flag is set during the AHB transfer

  Also, interrupts (if enabled) are provided along with the AHB error response. See Flag (FR) for complete list of errors.

41.8.2 Memory-mapped serial flash memory data—individual flash memory mode on flash memory A

Starting with address XSPI_AMBA_BASE, the content of the first external serial flash memory device is mapped into the address space of the device containing the XSPI module. Serial flash memory byte address 0h corresponds to bus address, XSPI_AMBA_BASE, in an increasing order. Assuming that a dual-die flash memory is connected to the first set of external pads, the address space is divided into two parts, one for each device of the dual-die package. See the following table for the address mapping. The byte ordering for 32-bit access is provided in Table 335 and for 64-bit read access, the byte ordering is provided in Table 336.

Table 407. Memory-mapped individual flash memory mode—flash memory A address scheme

Memory-mapped address 32-bit access	Memory-mapped address 64-bit access	Serial flash memory byte address	Flash memory device
XSPI_AMBA_BASE + 0h	XSPI_AMBA_BASE + 0h	0h to 3h	A1
XSPI_AMBA_BASE + 4h		4h to 7h	
...	...	...	
TOP_ADDR_MEMA1 - 8h	TOP_ADDR_MEMA1 - 8h	(TOP_ADDR_MEMA1- 8h) to (TOP_ADDR_MEMA1 - 0h4 -0h1)	
TOP_ADDR_MEMA1 - 4h		(TOP_ADDR_MEMA1 - 4h) to (TOP_ADDR_MEMA1 - 0h1)	
TOP_ADDR_MEMA1 + 4h	TOP_ADDR_MEMA1 + 0h	0h to 3h	A2
TOP_ADDR_MEMA1 + 0h4		4h to 7h	
...	...	...	
TOP_ADDR_MEMA2 - 8h	TOP_ADDR_MEMA2 - 8h	(TOP_ADDR_MEMA2 - 8h) to (TOP_ADDR_MEMA2 - 4h - 1h)	
TOP_ADDR_MEMA2 - 4h		(TOP_ADDR_MEMA2 - 4h) to (TOP_ADDR_MEMA2 - 1h)	

The available address range depends on the size of the external serial flash memory device. Any access beyond the size of the external serial flash memory provides undefined results.

For details concerning the read process, see Flash memory read using IP command.

Chapter 42
Cache Controller (CACHE64_CTRL)

42.1 Chip-specific CACHE64_CTRL information

Table 408. Reference links to related information

Topic	Related module	Reference
Full description	Cache controller	CACHE64_CTRL
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Power management	PMC	Power overview PMC
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
CACHE64 policy select	CACHE64_POLSEL	CACHE64_POLSEL
External serial peripheral interface	XSPI	XSPI

NOTE

The CACHE64_CTRL module works in conjunction with the CACHE64_POLSEL module. The CACHE64_POLSEL determines the cache policy to be applied for a given AHB address and passes this information to the CACHE64_CTRL.

42.1.1 Module instances

This device has two instances of the CACHE64_CTRL module, CACHE64_CTRL0 and CACHE64_CTRL1.

Table 409. CACHE64_CTRL instances

	CACHE64_CTRL0	CACHE64_CTRL1
Located in	VDD2_COMP	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
Cache for XSPI	XSPI0	XSPI1
Related CMX_PERFMON	CMX_PERFMON2	CMX_PERFMON3

NOTE

Only accesses to XSPI0 and XSPI1 from COMP_VDD bus initiators, like CPU0, eDMA0, eDMA1, are routed through the CACHE64_CTRL and CACHE64_POLSEL instances. HiFi4 and EZH-V can access the CACHE64_CTRL registers, but their accesses to XSPI0 and XSPI1 do NOT go through the cache.

42.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

42.1.3 CSAR register

For this chip, the CSAR[28] must be configured as 0b.

42.2 Overview

A cache is a block of high-speed memory locations that contains address information (commonly known as a tag) and its associated data. It decreases the average time of a memory access. CACHE64_CTRL includes cache memory controller (CACHE64), which is a general purpose AMBA AHB bus protocol cache.

42.2.1 Block diagram

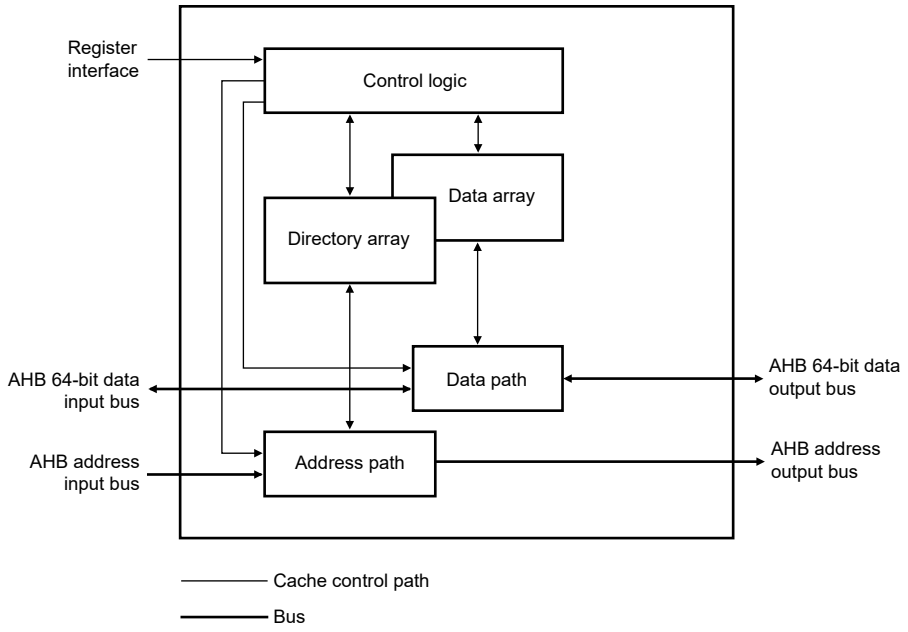


Figure 162. Block diagram

42.2.2 Features

CACHE64_CTRL is designed to work with AMBA AHB bus systems. It includes:

- An input (Controller) AHB bus that you can access using Cache mode attributes for cacheable copyback, cacheable write-through, or noncacheable mode operation and write bufferable mode operation on an access-by-access basis.
- An output (Target) 64-bit data path AHB bus for downstream cache read miss, cache write-through write, and noncacheable accesses.
- A peripheral interface for accessing the cache's program model.

42.3 Functional description

A cache operates on two principles of locality:

- Spatial locality — Access to one location possibly follows an access from adjacent locations (for example, sequential instruction execution or usage of a data structure).
- Temporal locality — Access to an area of memory possibly repeats within a short period of time (for example, execution of a code loop).

To minimize the quantity of control information stored, the spatial locality property groups several locations together, under the same tag. This logical block is known as a cache line.

You achieve temporal locality by retaining recently accessed lines in the cache.

When data loads into a cache, access times for subsequent loads and stores are reduced, resulting in overall performance benefits. An access to information already in a cache is known as a cache hit, and other accesses are known as cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the bus controller wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated, and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

Caches introduce a number of potential problems, mainly because of:

- Memory accesses occurring at times other than when the programmer would normally expect them.
- The existence of multiple physical locations where a data item can be held.

42.3.1 Operations

42.3.1.1 Cache configuration

CACHE64_CTRL receives manager bus requests on the manager (M) bus.

You can access the cache programming model via the cache controller's AMBA AHB bus.

CACHE64_CTRL then processes the cacheable accesses as needed, when bypassing the noncacheable, cache write-through, cache miss, and cache maintenance accesses to its subordinate bus.

The cache on this chip has a 2-way set associative cache structure with a total size of 32 KB. The cache has a 32-bit address, 64-bit data paths, and a 32-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

For this 32 KB cache, each cache TAG function uses two 512 x 20-bit RAM arrays, and the cache DATA function uses two 2048 x 64-bit RAM arrays. The cache TAG entries store 18 bits of the upper address as well as a modified and valid bit per cache line. The cache DATA entries store eight bytes of code or data.

All normal cache accesses use physical addresses. This leads to the use of the following cache address:

CACHE - 32 KB size = (512 sets) x (32-byte lines) x (2-way set associative)

Table 410. Tag cache address use

Tag cache address use	Cache
Tag hit address range	Address[31:14]
Tag set select address range	Address[13:5] used to select 1 of 512 sets
Not used	Address[4:0]

Table 411. Data cache address use

Data cache address use	System cache
Not used	Address[31:14]
Data set select address range	Address[13:5] used to select one of 512 sets

Table continues on the next page...

Table 411. Data cache address use (continued)

Data cache address use	System cache
64-bit word select	Address[4:3] used to select one of four 64-bit words within a set
Byte select	Address[2:0] used to select the byte within the 64-bit word

42.3.1.2 Cache control

The cache is disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable and configure the cache, you must clear the cache commands and initialize the required tag array.

42.3.1.2.1 Cache set commands

The cache set commands may operate on one of the following:

- All of way 0
- All of way 1
- All of both ways (complete cache)

Cache set commands are specified using the upper bits in Cache Control (CCR). Cache set commands perform their operation on the cache independent of CCR[ENCACHE].

You initiate a cache set command by writing 1 to CCR[GO]. This field also acts as a busy bit for set commands. The field remains 1 when the command is active, and the module writes 0 to it after the set command completes.

Supported cache set commands are provided in Table 412. Following are the set commands:

- Invalidate — Unconditionally clears valid and modify bits of a cache entry.
- Push — Pushes a cache entry if it is valid and modified, then clears the modify bit. If the entry is not valid or not modified, leave it as is.
- Clear — Pushes a cache entry if it is valid and modified, then clears the valid and modify bits. If the entry is not valid or not modified, clear the valid bit.

Table 412. Cache set commands

Cache Control (CCR)[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
0	0	0	0	No operation (NOP)
0	0	0	1	Invalidate all of way 0
0	0	1	0	Push all of way 0
0	0	1	1	Clear all of way 0
0	1	0	0	Invalidate all of way 1
0	1	0	1	Invalidate all of way 1 and way 0 (invalidate complete cache)
0	1	1	0	Invalidate all of way 1; push all of way 0
0	1	1	1	Invalidate all of way 1; clear all of way 0
1	0	0	0	Push all of way 1
1	0	0	1	Push all of way 1; invalidate all of way 0

Table continues on the next page...

Table 412. Cache set commands (continued)

Cache Control (CCR)[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
1	0	1	0	Push all of way 1 and way 0 (push complete cache)
1	0	1	1	Push all of way 1; clear all of way 0
1	1	0	0	Clear all of way 1
1	1	0	1	Clear all of way 1; invalidate all of way 0
1	1	1	0	Clear all of way 1; push all of way 0
1	1	1	1	Clear all of way 1 and way 0 (clear complete cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, writing 8500_0001h to Cache Control (CCR) invalidates the cache and enables the cache.

42.3.1.2.2 Cache line commands

Cache line commands operate on a single line in the cache at a time. You can perform these commands using a physical or cache address that consists of a set address and a way select. The line command acts on the specified cache line.

Cache line commands with physical addresses first search both ways of the cache set specified by the following physical address bits. If they hit, the commands perform their action on the hit way: For cache- [13:5].

You specify cache line commands using the upper bits in Cache Line Control (CLCR). Cache line commands perform their operation on the cache independent of CCR[ENCACHE]. Using a cache address, the command can be completely specified using Cache Line Control (CLCR). Using a physical address, the command must also use Cache Search Address (CSAR) to specify the physical address.

You can initiate a line cache command by writing 1 to CLCR[LGO] or CSAR[LGO]. This field also acts as a busy bit for line commands. The field remains 1 when the command is active and CACHE64_CTRL writes 0 to it after the set command completes.

Table 413. Cache line commands

Cache Line Control (CLCR)[27:24]			Command
CLCR[LACC]	CLCR[LADSEL]	CLCR[LCMD]	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	NOP

Table continues on the next page...

Table 413. Cache line commands (continued)

Cache Line Control (CLCR)[27:24]			Command
CLCR[LACC]	CLCR[LADSEL]	CLCR[LCMD]	
1	0	10	NOP
1	0	11	NOP
1	1	*xx*	NOP

42.3.1.2.2.1 Executing a series of line commands using cache addresses

You can perform a series of line commands with incremental cache addresses by writing to Cache Line Control (CLCR):

1. Place the command in Cache Line Control (CLCR)[27:24].
2. Write 1 to CLCR[WSEL] and CLCR[TDSEL] as needed.
3. Place the cache address in CLCR[CACHEADDR].
4. Write 1 to CLCR[LGO].
5. Wait for the command to complete (CLCR[LGO] becomes 0).
6. Increment the cache address (at bit 3 to step through data or at bit 5 to step through lines).
7. Write 1 to CLCR[LGO].
8. Wait for the command to complete (CLCR[LGO] becomes 0).
9. Repeat steps 5 - 8 as needed.

42.3.1.2.2.2 Executing a series of line commands using physical addresses

You can perform a series of line commands with incremental physical addresses using the following steps:

1. Write to Cache Line Control (CLCR):
   a. Place the command in Cache Line Control (CLCR) [27:24].
   b. Write 1 to CLCR[TDSEL].
2. Place the physical address in CSAR[PHYADDR] and write 1 to CSAR[LGO].
3. Wait for the command to complete (CLCR[LGO] becomes 0).
4. Increment the physical address (at bit 3 to step through data or at bit 5 to step through lines).
5. Write 1 to CLCR[LGO].
6. Wait for the command to complete (CLCR[LGO] becomes 0).
7. Repeat steps 3 - 6 as needed.

Cache Line Control (CLCR) and Cache Search Address (CSAR) both contain a field called LGO, and this field has the same functionality. Therefore, you can complete the above steps in a single write to Cache Search Address (CSAR).

42.3.1.2.2.3 Line command results

At completion of a line command, Cache Line Control (CLCR) contains information on the initial state of the line targeted by the command. For line commands with cache addresses, read this information before performing a line command action from the targeted cache line. For line commands with physical addresses, read this information on a hit before the line command action is performed from the hit cache line. In the cache, for line commands with a physical address miss, CLCR[LCIVB] becomes 0, and no line command action is performed. In general, if CLCR[LCIVB] becomes 0, the targeted line is invalid at the start of the line command and no line operation is performed.

Table 414. Line command results

Cache Line Control (CLCR)[22:20]			For cache address commands	For physical address commands
CLCR[LCWAY]	CLCR[LCIMB]	CLCR[LCIVB]		
0	0	0	Way 0 line is invalid	No hit
0	0	1	Way 0 valid and not modified	Way 0 valid and not modified
0	1	0	Way 0 line is invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line is invalid	No hit
1	0	1	Way 1 valid and not modified	Way 1 valid and not modified
1	1	0	Way 1 line is invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

On completion of a line command other than a write, Cache Read/Write Value (CCVR) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command used a physical address and missed, the data is don't care. For write commands, Cache Read/Write Value (CCVR) holds the write data.

The cache does not have line lock capability.

42.3.2 Modes of operation

Table 415. Modes of operation

Mode	Description
Write-Through	Access to address spaces with this cache mode are cacheable: • If all cacheable spaces are read-only spaces, then cache contains read-only data and all writes to the cache cause a fault. See the chip-specific cacheable space information. • A read miss on the input bus causes a line read on the output bus of a 32-byte-aligned memory address that contains the desired address. A read miss data loads into the cache and marks as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus accesses. • A write miss bypasses the cache and writes to the output bus (no allocation on write-miss policy for Write-Through mode spaces). • A write hit updates the cache hit line with the write data and writes to the output bus. • The caches are bus-controller-local and do not support hardware cache coherency. If the bus controller accesses write-through regions and another bus controller that can access the same target memory, without going through the cache (such as DMA), must update these

Table continues on the next page...

Table 415. Modes of operation (continued)

Mode	Description
	regions, you must first perform explicit cache clears to ensure all modified cache lines update their associated memories, before they are modified by external controllers and subsequent cache bus controller accesses get the updated memory.
Write-Back	Access to address spaces with this cache mode are cacheable: • A read miss on the input bus causes a line read on the output bus of a 32-byte-aligned memory address that contains the desired address. This miss data loads into the cache and marks it as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus access. • A write miss performs a read-to-write (executes write-miss policy for Write-Back mode spaces). You perform a line read on the output bus of a 32-byte-aligned memory address containing the desired write address. This miss data is loaded into the cache and marked as valid and modified. Then the write data updates the appropriate cache data locations. • A write hit updates the cache hit line with the write data and marks the line as modified. • Caches do not support hardware cache coherency. If the cache's bus controller has written to write-back regions, and another bus controller that can access the same target memory without going through the cache must see these updates, you must perform explicit cache pushes to any needed cache memory range to ensure that all modified cache lines update their associated memories before being this controller reads them. Likewise, if the cache's bus controller has accessed write-back regions, and another bus controller that can access the same target memory without going through the cache must update these regions, you must first perform explicit cache clears to any needed cache memory range to ensure that all modified cache lines update their associated memories before this controller modifies them, and subsequent cache controller bus accesses get the updated memory.
Noncacheable	Access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

42.3.3 Clocking

Caches use the AHB manager bus clock.

42.3.4 Reset

Caches are disabled at reset. However, cache tag and data arrays are not cleared at reset. See Initialization and Cache control for more information regarding cache initialization and configuration.

42.3.5 Interrupts

CACHE64_CTRL has no interrupts.

42.4 External signals

There are no external signals for CACHE64_CTRL.

42.5 Initialization

Perform this procedure to initialize cache:

1. Write 1 to CCR[INVW1] and CCR[INVW0].
2. Write 1 to CCR[GO].
3. Write 1 to CCR[ENCACHE].

NOTE

Steps 1, 2 and 3 can be done simultaneously. That is, writing 8500_0003h to Cache Control (CCR) invalidates the cache and enables the cache and write buffer.

42.6 Memory map and registers

The cache programmer's model provides a variety of registers for configuring and controlling the cache, as well as indirect access paths to all cache tag and data storage.

NOTE

The cache registers are accessible in Supervisor mode only.

42.6.1 CACHE64 register descriptions

42.6.1.1 CACHE64 memory map

CACHE64_CTRL0 base address: 4003_5000h

CACHE64_CTRL1 base address: 4003_6000h

Offset	Register	Width (In bits)	Access	Reset value
800h	Cache Control (CCR)	32	RW	0000_0000h
804h	Cache Line Control (CLCR)	32	RW	0000_0000h
808h	Cache Search Address (CSAR)	32	RW	0000_0000h
80Ch	Cache Read/Write Value (CCVR)	32	RW	0000_0000h

42.6.1.2 Cache Control (CCR)

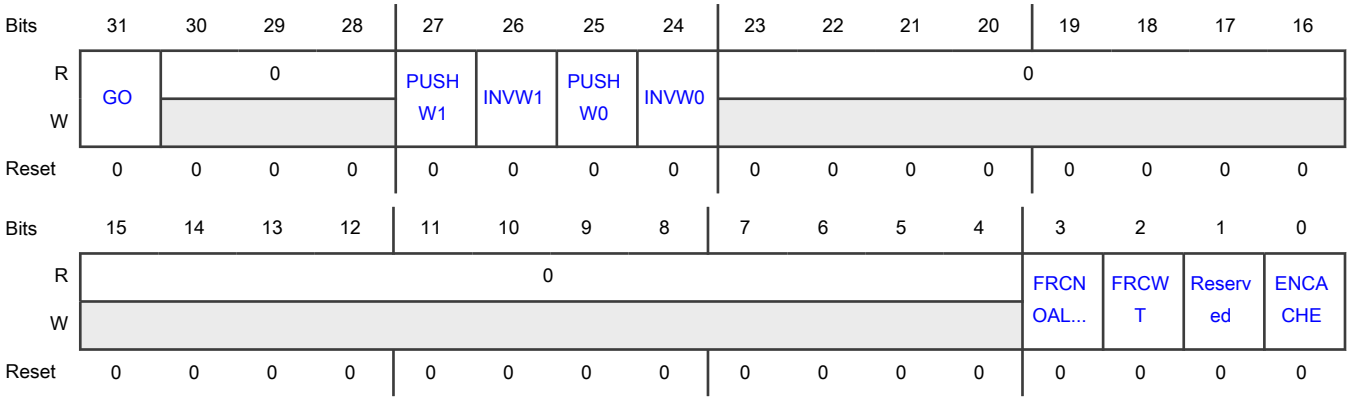
Offset

Register	Offset
CCR	800h

Function

Enables the cache. This register also contains the fields used for commands specified in Cache set commands.

Diagram



Fields

Field	Function
31 GO	Initiate Cache Command Initiates the cache command that CCR[INVW0], CCR[PUSHW0], CCR[INVW1], and CCR[PUSHW1] indicate, if you write 1 to this field. Reading this field indicates whether a cache command is active. **NOTE** This field remains 1 until the cache command completes. Writing 0 has no effect. 0b - Write: no effect; Read: no cache command active 1b - Write: initiates cache command; Read: cache command active
30-28 —	Reserved
27 PUSHW1	Push Way 1 Specifies the cache command to execute the next time CCR[GO] becomes 1. 0b - No operation 1b - Push all modified lines in way 1
26	Invalidate Way 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
INVW1	Specifies the cache command to execute the next time CCR[GO] becomes 1. **NOTE** If CCR[PUSHW1] and CCR[INVW1] are 1, then after you write 1 to CCR[GO], this field pushes all modified lines in way 1 and invalidates all lines in way 1 (clear way 1). 0b - No operation 1b - Invalidates all lines in way 1
25 PUSHW0	Push Way 0 Specifies the cache command to execute the next time CCR[GO] becomes 1. 0b - No operation 1b - Push all modified lines in way 0
24 INVW0	Invalidate Way 0 Specifies the cache command to execute the next time CCR[GO] becomes 1. **NOTE** If CCR[PUSHW0] and CCR[INVW0] are 1, then after you write 1 to CCR[GO], this field pushes all modified lines in way 0 and invalidates all lines in way 0 (clear way 0). 0b - No operation 1b - Invalidates all lines in way 0
23-4 —	Reserved
3 FRCNOALLC	Forces No Allocation On Cache Misses 0b - Allocation on cache misses 1b - Forces no allocation on cache misses (FRCWT must be asserted)
2 FRCWT	Force Write Through Mode Specifies whether all cacheable spaces to write through are forced. 0b - Does not force 1b - Force
1 —	This field is reserved. Software should write 0 to this bit to maintain compatibility.
0 ENCACHE	Cache Enable Enables the cache. 0b - Disables 1b - Enables

Table continues on the next page...

Field	Function

42.6.1.3 Cache Line Control (CLCR)

Offset

Register	Offset
CLCR	804h

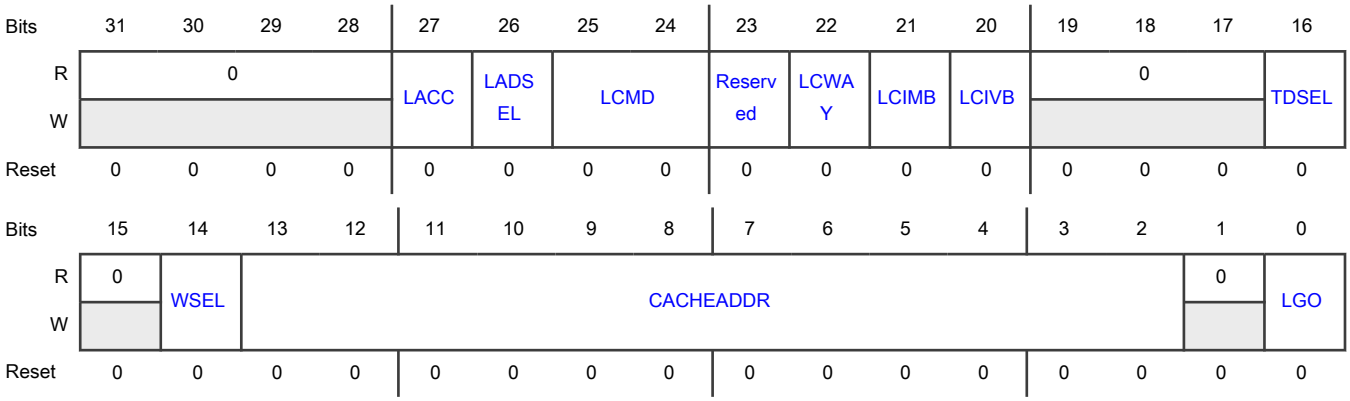
Function

Defines the commands specified in Cache line commands that you must perform using a specific cache line address or physical address.

If you specify the physical address, you can search cache in both the ways, and the command only performs on the way that hits.

You can write to CLCR[LCWAY], CLCR[LCIMB], and CLCR[LCIVB]; however, the values that you write to these fields have no functional purpose.

Diagram



Fields

Field	Function
31-28 —	Reserved
27 LACC	Line Access Type Specifies the operation that you must perform (read or write) after a hit, when performing a search command. **NOTE** Only used for search commands (CLCR[LCMD] = 00). 0b - Read

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Write
26 LADSEL	Line Address Select Specifies the address (cache or physical). When using the cache address (specified in CLCR[WSEL] and CLCR[CACHEADDR]), the way must also be specified in CLCR[WSEL]. When using the physical address (specified in Cache Search Address (CSAR)), both ways are searched and the command is performed only after a hit. 0b - Cache 1b - Physical
25-24 LCMD	Line Command Performs one of the four commands after a cache hit. CLCR[LACC] determines whether the operation is a read or write. 00b - Search and read or write 01b - Invalidate 10b - Push 11b - Clear
23 —	Reserved
22 LCWAY	Line Command Way Indicates the way that the line command uses. Applies only if CLCR[LCIVB] = 1. 0b - Way 0 1b - Way 1
21 LCIMB	Line Command Initial Modified Bit Shows the initial state of the modified bit. If this field = 0, and the command uses cache address and way, this field shows the initial state of the modified bit, which is 0. If the command uses physical address and a hit, this field shows the initial state of the modified bit, which is 0. If the command uses physical address and is a miss, this field reads 0. If this field = 1, and the command uses cache address and way, this field shows the initial state of the modified bit, which is 1. If the command uses physical address and way, this field shows the initial state of the modified bit, which is 1. If a miss, this field reads 0. 0b - Initial state 0 1b - Initial state 1
20 LCIVB	Line Command Initial Valid Bit Shows the initial state of the valid bit.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If this field = 0, and the command uses cache address and way, this field shows the initial state of the valid bit, which is 0. If the command uses physical address and a hit, this field shows the initial state of the valid bit, which is 0. If the command uses physical address and is a miss, this field reads 0. If this field = 1, and the command uses cache address and way, this field shows the initial state of the valid bit, which is 1. If the command uses physical address and way, this field shows the initial state of the valid bit, which is 1. If a miss, this field reads 0. 0b - Initial state 0 1b - Initial state 1
19-17 —	Reserved
16 TDSEL	Tag Or Data Select Selects tag or data for search and read or write commands. 0b - Data 1b - Tag
15 —	Reserved
14 WSEL	Way Select Selects the way for line commands. 0b - Way 0 1b - Way 1
13-2 CACHEADDR	Cache Address Represents bits [13:2] of the cache address: CLCR[13:5] accesses the tag arrays. CLCR[13:3] accesses the data arrays. CLCR[2] specifies which half of the 64 bits of data to place in Cache Read/Write Value (CCVR).
1 —	Reserved
0 LGO	Initiate Cache Line Command Initiates the cache command that CLCR[LCMD], CLCR[LADSEL], and CLCR[LACC] indicate, if you write 1 to this field. Reading this field indicates whether a cache command is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE • This field remains 1 until the command completes. Writing 0 has no effect. • This field is shared with CSAR[LGO]. You can write either register address bit[0] interchangeably. This saves cache writes (helps performance), when performing cache operations. Cache line operations start when you write to Cache Line Control (CLCR) with CLCR[LGO] being 1 or the cache search address can be changed and a search is started by writing to Cache Search Address (CSAR) with CSAR[LGO] being 1. 0b - Write: no effect; Read: no line command active 1b - Write: initiate line command; Read: line command active

42.6.1.4 Cache Search Address (CSAR)

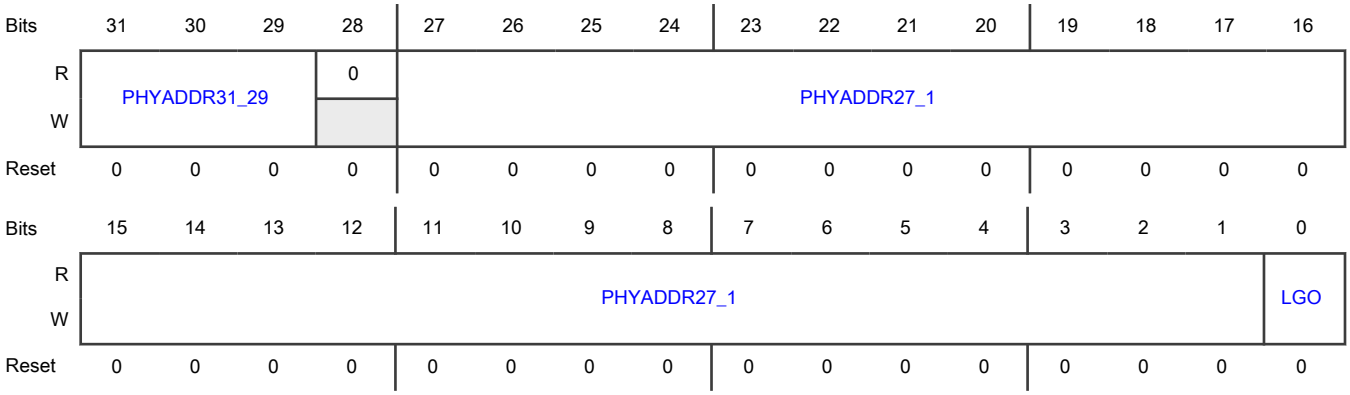
Offset

Register	Offset
CSAR	808h

Function

Defines the physical address for cache line commands, when CLCR[LADSEL] = 1.

Diagram



Fields

Field	Function
31-29	Physical Address PHYADDR31_29 represents bits [31:29] and PHYADDR27_1 represents bits [27:1] of the system address.

Table continues on the next page...

Table continued from the previous page...

Field	Function
PHYADDR31_29	CSAR[31:14] is used for tag compare. CSAR[13:5] accesses the tag arrays. CSAR[13:3] accesses the data arrays. CSAR[2] specifies which half of the 64 bits of data to place in Cache Read/Write Value (CCVR). CSAR[1] is not used. Writes to this bit are allowed, and the read value is the previously written value. This bit is not functionally used.
28 —	Reserved Always read as zero.
27-1 PHYADDR27_1	Physical Address PHYADDR31_29 represents bits [31:29] and PHYADDR27_1 represents bits [27:1] of the system address. CSAR[31:14] is used for tag compare. CSAR[13:5] accesses the tag arrays. CSAR[13:3] accesses the data arrays. CSAR[2] specifies which half of the 64 bits of data to place in Cache Read/Write Value (CCVR). CSAR[1] is not used. Writes to this bit are allowed, and the read value is the previously written value. This bit is not functionally used.
0 LGO	Initiate Cache Line Command Initiates the cache command that CSAR[PHYADDR] indicates, if you write 1 to this field. Reading this field indicates whether a cache command is active. **NOTE** • This field remains 1 until the command completes. Writing 0 has no effect. • This field is shared with CLCR[LGO]. 0b - Write: no effect; Read: no line command active 1b - Write: initiate line command; Read: line command active

42.6.1.5 Cache Read/Write Value (CCVR)

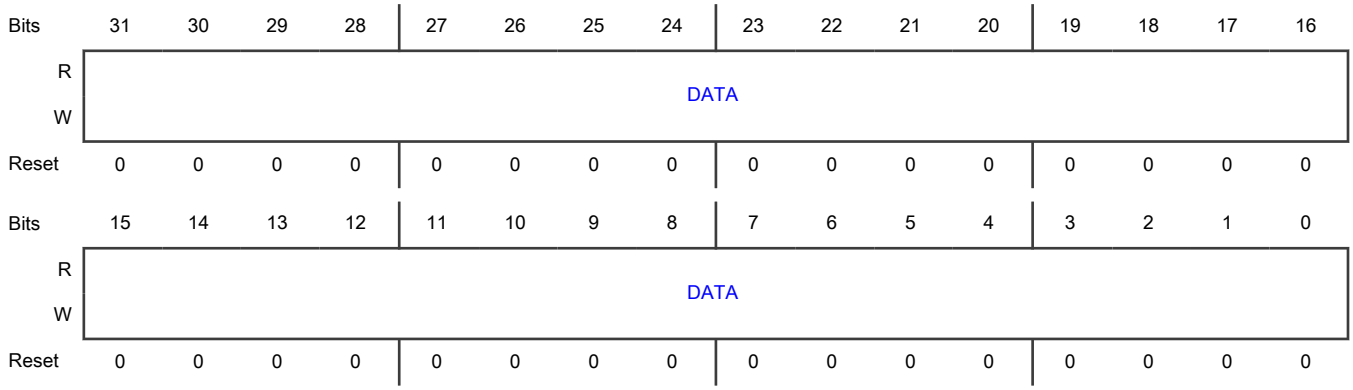
Offset

Register	Offset
CCVR	80Ch

Function

Sources write data or return read data when a search line command is requested (CLCR[LCMD] = 00). CLCR[LACC] determines whether the access is a read or write, and CLCR[TDSEL] determines whether the access is to the data or tag arrays.

Diagram



Fields

Field	Function
31-0 DATA	Cache Read/Write Data Reads or writes data during tag search and data search. For tag search, read or write: • CCVR[31:14] for tag array R/W value. • CCVR[13:5] for tag set address on reads; unused on writes. • CCVR[4:2] are reserved. • CCVR[1] for tag modify bit. • CCVR[0] for tag valid bit. For data search, read or write: • CCVR[31:0] for data array R/W value.

Chapter 43
CACHE64 Policy Select (CACHE64_POLSEL)

43.1 Chip-specific CACHE64_POLSEL information

Table 416. Reference links to related information

Topic	Related module	Reference
Full description	CACHE64_POLSEL	CACHE64_POLSEL
Peripheral memory map		Memory map overview
Clock controller		Clock controller
Power management	PMC	Power overview PMC
CACHE64 controller	CACHE64_CTRL	CACHE64_CTRL
External serial peripheral interface	XSPI	XSPI

NOTE

Each of the CACHE64_CTRL modules interface to a XSPI module for access to off-chip memory. The 128 MB space divided into three policy regions, is allocated for each CACHE64 POLSEL interface. POLICY_SEL register has three 2-bit fields associated with the three defined regions. Each field designates the policy to be applied to addresses falling within that region. The three policies are completely independent so, it is possible to assign the same policy to all three regions.

43.1.1 Module instances

This device has two instances of the CACHE64_POLSEL module, CACHE64_POLSEL0 and CACHE64_POLSEL1.

Table 417. CACHE64_POLSEL instances

	CACHE64_POLSEL0	CACHE64_POLSEL1
Located in	VDD2_COMP	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

NOTE

Only accesses to XSPI0 and XSPI1 from COMP_VDD bus initiators, like CPU0, eDMA0, eDMA1, are routed through the CACHE64_CTRL and CACHE64_POLSEL instances. HiFi4 and EZH-V can access the CACHE64_POLSEL registers, but their accesses to XSPI0 and XSPI1 do NOT go through the cache.

43.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

43.2 Overview

CACHE64_POLSEL monitors incoming AHB addresses to define the policy that the cache controller (CACHE64_CTRL) uses. Both CACHE64_POLSEL and the cache controller work in tandem with each other.

43.2.1 Features

- Provides cache policy information to the cache controller.
- Supports up to three regions, each with a programmable cache policy:
  - Each region can be independently designated as noncache, write-through cache, and write-back cache.
  - User-defined boundaries are supported between the three regions with 1 KB granularity.

43.3 Functional description

CACHE64_POLSEL permits dividing the memory space that the cache controller accesses into three contiguous regions and applies one of these three cache policies to each region:

- First region (region 0) starts at the bottom (lowest address) in the target address space and goes up to the top of the 1 KB boundary designated in REG0_TOP.
- Second region (region 1) starts at the next address and extends to the top of the boundary designated in REG1_TOP.
- Third region starts at the address after that and continues to the top of the memory space accessed via the cache controller.

43.3.1 Region mapping

CACHE64_POLSEL controls the cache policy for memory between the base address and end address. Within the CACHE64_POLSEL memory area, you can define up to three contiguous memory regions, with each region having independently programmable cache policy settings.

As shown in the following figure, REG0_TOP and REG1_TOP control the boundaries between the regions. These registers provide bits 26-10 of the address and the lower 10 bits (9-0) are always 1s. For example, if REG0_TOP is 0001_0000h, then region 0 ends at BASE_ADDRESS + 0001_03FFh. Region 1 starts at 0001_0400h.

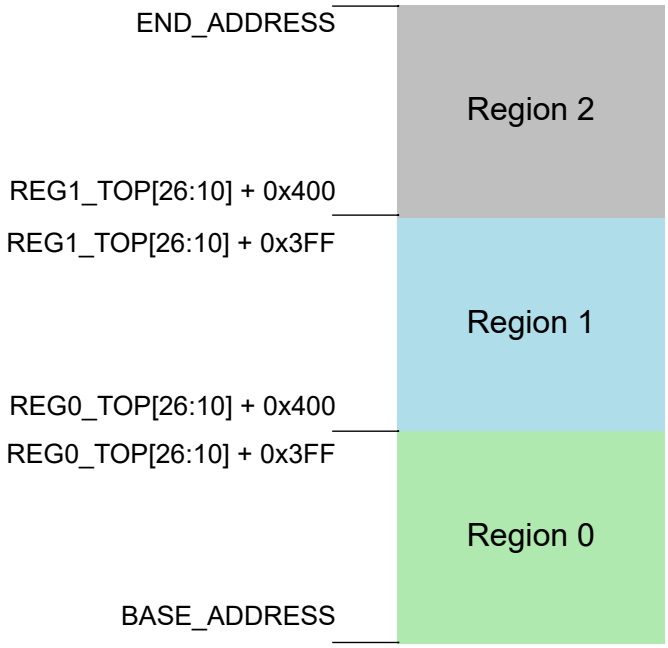


Figure 163. Cache memory regions

For CACHE64_POLSEL implementations that have multiple memory regions (more than one base address and end address), the regions are aliased across all of the memory regions. Therefore, the configured cache policies are also aliased across all of the memory regions.

43.3.2 Region priority

When you access an address, CACHE64_POLSEL attempts to match it to one of the regions in a sequential order that starts with region 0. In pseudocode, the region hit logic does this:

```
if (address < REG0_TOP + 0x3FF)
  address hits in Region 0;
else
  if (address < REG1_TOP + 0x3FF)
    address hits in Region 1;
  else
    address hits in Region 2;
```

NOTE

If you configure REG0_TOP and REG1_TOP to create overlapping regions, then the lowest number region in which the address hits, is used.

43.4 Application information

This section provides an example CACHE64_POLSEL configuration. The example assumes:

- The MCU has two memory regions mapped to CACHE64_POLSEL:

— Nonsecure region: BASE_ADDRESS = 0800_0000h, END_ADDRESS = 0FFF_FFFFh

— Secure region: BASE_ADDRESS = 1800_0000h, END_ADDRESS = 1FFF_FFFFh

- Region 0: 56 MB is configured as write-through.
- Region 1: 8 MB is configured as noncacheable.
- Region 2: The remaining memory region is unused and configured as invalid.

The following register settings are used to achieve this configuration:

- REG0_TOP = 037F_FC00h
- REG1_TOP = 03FF_FC00h
- POLSEL = 0000_0031h

The following figure shows the resulting memory configuration.

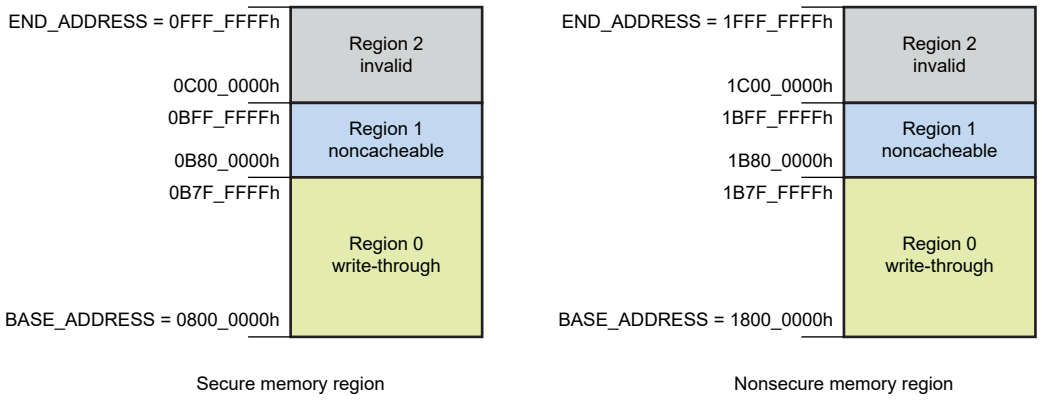


Figure 164. CACHE64_POLSEL configuration example

43.5 Memory map and register definition

The CACHE64_POLSEL registers are included in the register space of the CACHE64 module, utilizing addresses that CACHE64 does not use internally.

43.5.1 CACHE64_POLSEL register descriptions

43.5.1.1 CACHE64_POLSEL memory map

CACHE64_POLSEL0 base address: 4003_5000h

CACHE64_POLSEL1 base address: 4003_6000h

Offset	Register	Width (In bits)	Access	Reset value
14h	Region 0 Top Boundary (REG0_TOP)	32	RW	02AA_A800h
18h	Region 1 Top Boundary (REG1_TOP)	32	RW	0555_5400h
1Ch	Policy Select (POLSEL)	32	RW	0000_0000h

43.5.1.2 Region 0 Top Boundary (REG0_TOP)

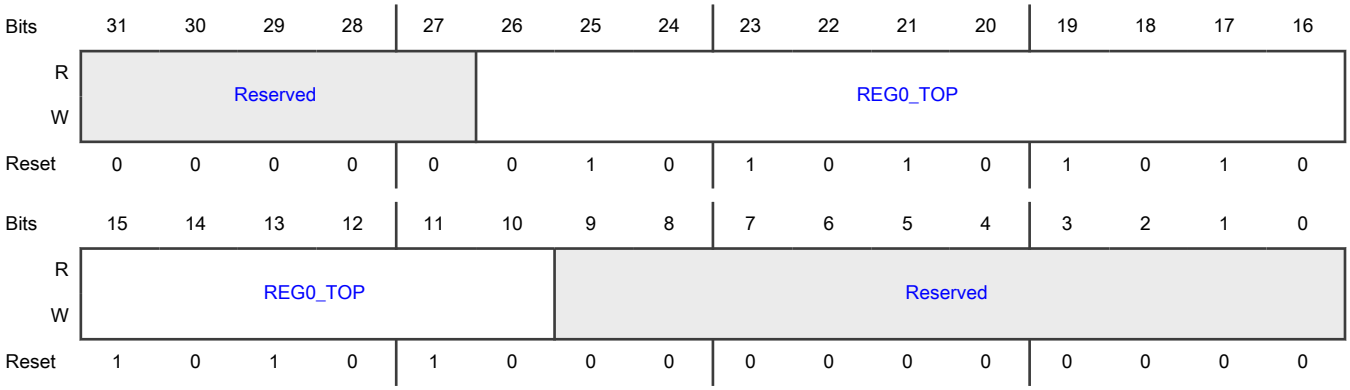
Offset

Register	Offset
REG0_TOP	14h

Function

Provides bits 26-10 of the region 0 compare address. The actual granularity implemented is 1 KB.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-10 REG0_TOP	Upper Limit Of Region 0 Provides bits 26-10 of the region 0 compare address. The value provided here is concatenated with 3FFh to create the full compare address. For Example, if REG0_TOP[26:10] = 0b00000000000_111111, the region 0 end address is BASE_ADDRESS + 0000_FFFFh (64 KB).
9-0 —	Reserved

43.5.1.3 Region 1 Top Boundary (REG1_TOP)

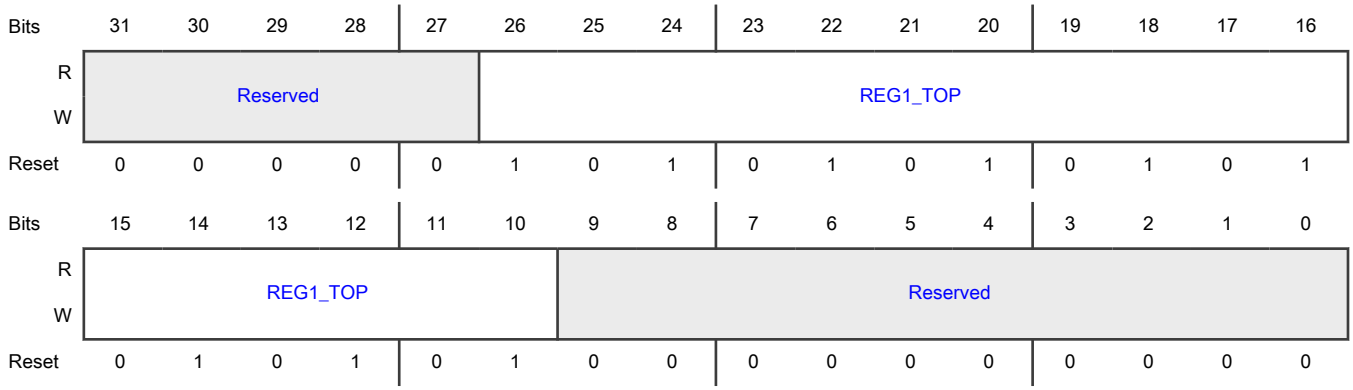
Offset

Register	Offset
REG1_TOP	18h

Function

Provides bits 26-10 of the region 1 compare address. The actual granularity implemented is 1 KB.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-10 REG1_TOP	Upper Limit Of Region 1 Provides bits 26-10 of the region 1 compare address. The value provided here is concatenated with 3FFh to create the full compare address. For example, if REG1_TOP[26:10] = 0b00000000000_111111, the region 0 end address is BASE_ADDRESS + 0000_FFFFh (64 KB).
9-0 —	Reserved

43.5.1.4 Policy Select (POLSEL)

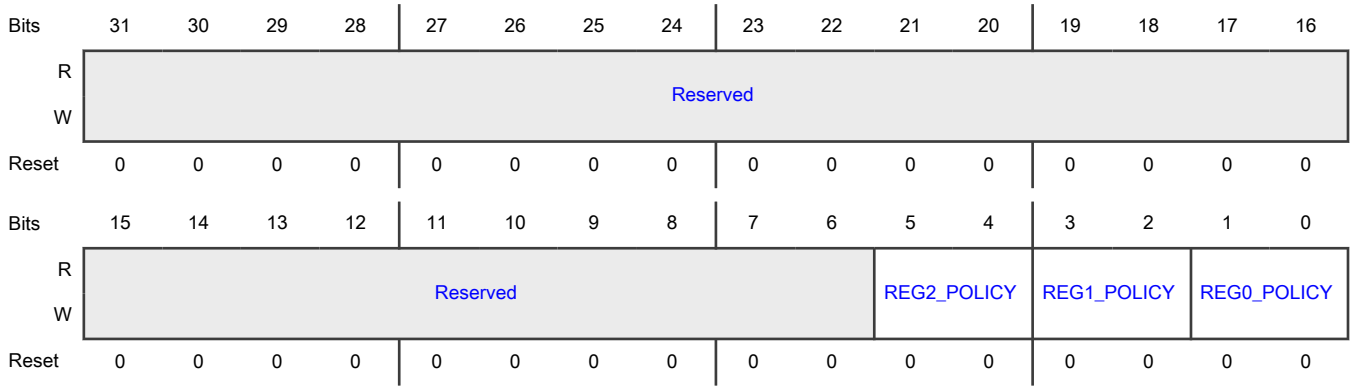
Offset

Register	Offset
POLSEL	1Ch

Function

Specifies the policy to be applied to all addresses falling within the designated space.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-4 REG2_POLICY	Policy Select For Region 2 Specifies the policy to be applied to all addresses falling within the space designated as region 2. 00b - Noncacheable 01b - Write-through 10b - Write-back 11b - Invalid
3-2 REG1_POLICY	Policy Select For Region 1 Specifies the policy to be applied to all addresses falling within the space designated as region 1. 00b - Noncacheable 01b - Write-through 10b - Write-back 11b - Invalid
1-0 REG0_POLICY	Policy Select For Region 0 Specifies the policy to be applied to all addresses falling within the space designated as region 0. 00b - Noncacheable 01b - Write-through 10b - Write-back 11b - Invalid

Chapter 44
Memory Map Unit (MMU)

44.1 Chip-specific MMU information

Table 418. Reference links to related information

Topic	Related module	Reference
Full description	MMU	MMU
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

44.1.1 Module instances

This device contains three instances of the MMU module, MMU0, MMU1 and MMU2.

Table 419. MMU instances

	MMU0	MMU1	MMU2
Located in	VDD2_COMP	VDD2_COMP	VDD2_MEDIA
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
Related XSPI	XSPI0	XSPI1	XSPI2
4 KB block-size unit support	No	No	Support only on memory region of 6000_0000h-67FF_FFFFh

44.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

44.1.3 Registers protection

The CTRL, ADDR_RANGE, LUT_WDATA registers are protected by GLIKEY modules.

Table 420. Registers protected by GLIKEY module

Registers in MMU instance	Protected by GLIKEY instance	WRITE_INDEX
MMU0	GLIKEY0	15
MMU1	GLIKEY0	14
MMU2	GLIKEY2	15

44.2 Overview

MMU remaps the advanced high-performance bus (AHB) access address to another address that the MMU lookup table (LUT) SRAM indicates. You can configure the remapped address in LUT SRAM by using the internal register bus (IPS).

MMU allows software a way to use large memory devices without having prior knowledge of the location of an application at compile time, thus spares the need of complex linker scripts. It can be used for example, to support partial retention of external pseudo-static RAM (PSRAM) by remapping external memory space into a single region for retention purposes. It can also support flash swap or provide support for multiple software images in external flash.

44.2.1 Block diagram

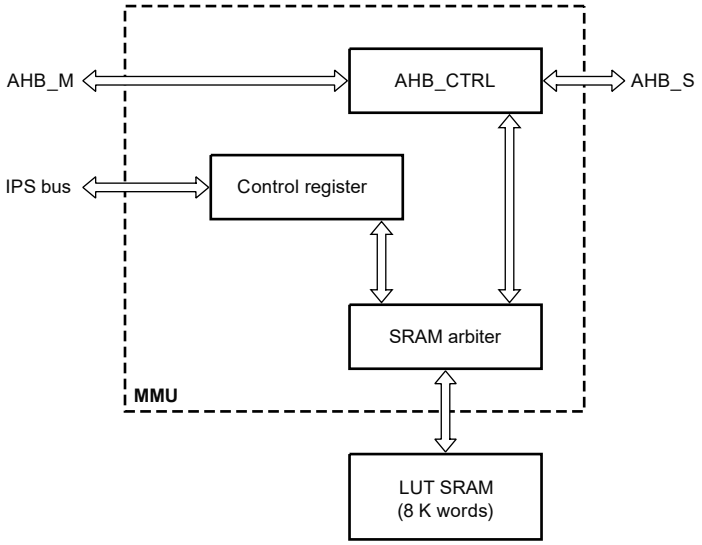


Figure 165. Block diagram

44.2.2 Features

- Automatically initializes LUT SRAM after reset
- Supports single- and burst-write modes for LUT SRAM settings
- Supports block sizes of 4 KB, 8 KB, 16 KB, 32 KB, and 64 KB
- Supports parity check of the address in LUT SRAM

44.3 Functional description

44.3.1 Operations

- LUT SRAM write operation, which includes these methods to initialize LUT SRAM: Single-write operation and Burst-write operation
- LUT SRAM read operation
- LUT SRAM read-verify operation
- LUT SRAM address-map operation

44.3.1.1 Single-write operation

Perform the following procedure for a single-write operation:

1. Configure Write Pointer (WRITE_PTR) with the LUT SRAM address.

2. Write data to LUT Write Data (LUT_WDATA). Ensure LUT_WDATA[COUNT] = 0 to trigger a write operation to LUT SRAM.

   Every write to LUT Write Data (LUT_WDATA) increases the value of Write Pointer (WRITE_PTR) by 1. Therefore, you can write to LUT Write Data (LUT_WDATA) multiple times without configuring Write Pointer (WRITE_PTR) again for subsequent LUT SRAM addresses.

   For example, to initialize LUT SRAM addresses 2 and 3 with values 1234h and 5678h, respectively, perform the following procedure:

   a. Write 2h to Write Pointer (WRITE_PTR).

   b. Write 1234h to LUT Write Data (LUT_WDATA).

   c. Write 5678h to LUT Write Data (LUT_WDATA).

44.3.1.2 Burst-write operation

1. Configure Write Pointer (WRITE_PTR) with the LUT SRAM address.
2. Write data to LUT_WDATA[VALID] and LUT_WDATA[ADDR], with LUT_WDATA[COUNT] > 0. This triggers COUNT + 1 write operations to LUT SRAM, incrementing the write address and data written each time.

   For example, to initialize LUT SRAM addresses 2 and 3 with values 1234h and 1235h, respectively, perform the following procedure:

   a. Write 2h to Write Pointer (WRITE_PTR).

   b. Write 00011234h to LUT Write Data (LUT_WDATA).

3. Wait for the burst-write operation to complete by polling STS[BUSY] before the next IPS action.

44.3.1.3 Read operation

1. Configure Read Pointer (READ_PTR) with the LUT SRAM address.
2. Read data from LUT Read Data (LUT_RDATA). This triggers a read operation from LUT SRAM.

   Every read from LUT Read Data (LUT_RDATA) increases the value of Read Pointer (READ_PTR) by 1. Therefore, you can read from LUT Read Data (LUT_RDATA) multiple times without configuring Read Pointer (READ_PTR) again for subsequent LUT SRAM addresses.

   For example, to read LUT SRAM addresses 2 and 3, perform the following procedure:

   a. Write 2h to Read Pointer (READ_PTR).

   b. Read from LUT Read Data (LUT_RDATA); the register returns data from LUT SRAM address 2.

   c. Read from LUT Read Data (LUT_RDATA); the register returns data from LUT SRAM address 3.

44.3.1.4 Read-verify operation

This operation performs a burst read, and compares the values read from the LUT against expected data.

1. Configure Read Pointer (READ_PTR) with the LUT SRAM address.
2. Write data to Read Verify (READ_VFY). This triggers COUNT + 1 read operations from LUT SRAM, incrementing the read address and expected data[1] each time.

   For example, the expected values for LUT SRAM addresses 2 and 3 are 1234h and 1235h, respectively. Perform the following procedure to verify:

   a. Write 2h to Read Pointer (READ_PTR).

   b. Write 00011234h to Read Verify (READ_VFY).

[1] The expected data is used for checks in read verify operations. MMU compares the read data against the expected data.

c. MMU reads LUT SRAM address 2 and compares read data with 1234h. STS[IPS_READ_ERROR] is set if the compare fails.

d. MMU reads LUT SRAM address 3 and compares read data with 1235h. STS[IPS_READ_ERROR] is set if the compare fails.

3. Wait for the read-verify operation to complete by polling STS[BUSY] before the next IPS action.

44.3.1.5 Address-map operation

MMU has 8 KB words of LUT SRAM. It can be separated in 4 pages as configured by CTRL[PAGE_MODE].

Example 1:

Perform address-map operation where LUT SRAM works as 1 page (CTRL[PAGE_MODE] = 0).

The following table shows the address-map bit settings.

Table 421. MMU address bits map 1

Unit size	Initiator address	Address for LUT SRAM	Data from LUT SRAM (map address)	Target address
4 KB	`mhaddr[31:0]`[1]	`mhaddr[24:12]`	`mapaddr[14:0]`[2]	`{mhaddr[31:28], 1'b0, mapaddr[14:0], mhaddr[11:0]}`
8 KB	`mhaddr[31:0]`	`mhaddr[25:13]`	`mapaddr[14:0]`	`{mhaddr[31:28], mapaddr[14:0], mhaddr[12:0]}`
16 KB	`mhaddr[31:0]`	`mhaddr[26:14]`	`mapaddr[14:0]`	`{mhaddr[31:28], mapaddr[13:0], mhaddr[13:0]}`
32 KB	`mhaddr[31:0]`	`mhaddr[27:15]`	`mapaddr[14:0]`	`{mhaddr[31:28], mapaddr[12:0], mhaddr[14:0]}`
64 KB	`mhaddr[31:0]`	`mhaddr[28:16]`	`mapaddr[14:0]`	`{mhaddr[31:28], mapaddr[11:0], mhaddr[15:0]}`

1. `mhaddr` = initiator AHB address
2. `mapaddr` = map address from LUT SRAM

Example 2:

Perform address-map operation where LUT SRAM works as 4 pages (CTRL[PAGE_MODE] = 1).

The following table shows the address-map bit settings.

Table 422. MMU address bits map 2

Unit size	Initiator address	Address for LUT SRAM	Data from LUT SRAM (map address)	Target address
4 KB	`mhaddr[31:0]`[1]	`{page_sel, mhaddr[22:12]}`[2]	`mapaddr[14:0]`[3]	`{mhaddr[31:28], 1'b0, mapaddr[14:0], mhaddr[11:0]}`

Table continues on the next page...

Table 422. MMU address bits map 2 (continued)

Unit size	Initiator address	Address for LUT SRAM	Data from LUT SRAM (map address)	Target address
8 KB	mhaddr[31:0]	{page_sel, mhaddr[23:13]}	mapaddr[14:0]	{mhaddr[31:28], mapaddr[14:0], mhaddr[12:0]}
16 KB	mhaddr[31:0]	{page_sel, mhaddr[24:14]}	mapaddr[14:0]	{mhaddr[31:28], mapaddr[13:0], mhaddr[13:0]}
32 KB	mhaddr[31:0]	{page_sel, mhaddr[25:15]}	mapaddr[14:0]	{mhaddr[31:28], mapaddr[12:0], mhaddr[14:0]}
64 KB	mhaddr[31:0]	{page_sel, mhaddr[26:16]}	mapaddr[14:0]	{mhaddr[31:28], mapaddr[11:0], mhaddr[15:0]}

1. mhaddr = initiator AHB address
2. page_sel = page selection value that ranges from 0 to 3
3. mapaddr = map address from LUT SRAM

Example 3:

For AHB secure access using address-map page 3, CTRL[UNIT_SIZE] = 64 KB:

- Assume that the AHB initiator address is A456CDEFh; therefore, mhaddr is A456CDEFh.
- The page_sel element is 3h and mhaddr[26:16] is 456h; therefore, read address for LUT SRAM must be 1C56h.
- Assume that the read data from LUT SRAM is 8AFAh. Bit 15 indicates that the read data is valid, and the lower 15 bits indicate the map address (mapaddr).
- The AHB target address must be AAFACDEFh.

44.3.2 Modes of operation

Table 423. Modes of operation

Mode	Description
Initialization	MMU enters Initialization mode when hardware reset is deasserted. Contents of the LUT SRAM become 0 in this mode. STS[BUSY] remains 1 until the initialization sequence completes.
Bypass	MMU enters Bypass mode when CTRL[MMU_EN] is 0. In this mode, the AHB transaction passes through MMU without address remapping.
Normal	MMU enters Normal mode when CTRL[MMU_EN] is 1. In this mode, the addresses of the AHB transaction are remapped by looking up LUT SRAM.

44.3.3 Clocking

CTRL[CLK_DIS] controls clock gating of the AHB clock. Write 1 to CTRL[CLK_DIS] to disable MMU's AHB clock to save power when MMU is not utilized.

44.3.4 Reset

Write 1 to CTRL[SWRST] to reset the MMU logic.

44.3.5 Interrupts

MMU includes an interrupt output signal. It is generated by interrupts in MMU Status (STS) when the related interrupt enable fields are set.

Interrupt sources are included when the following conditions are met:

- The LUT SRAM value is invalid.
- The AHB access address is not in the valid address range.
- Parity of the LUT SRAM value is not correct.
- You write to LUT Write Data (LUT_WDATA) when the LUT SRAM is locked.
- A read error occurs when reading from LUT Read Data (LUT_RDATA).

44.4 External signals

This module has no external signals.

44.5 Initialization

Step	Purpose	Programming
1	Enable the MMU clock.	Write 0 to CTRL[CLK_DIS].
2	Define the basic unit used in the memory map.	Configure CTRL[UNIT_SIZE].
3	Define the valid address range.	Configure Address Range (ADDR_RANGE).
4	Configure the LUT SRAM.	Write appropriate values to Write Pointer (WRITE_PTR) and LUT Write Data (LUT_WDATA).
5	Enable the MMU remap function.	Write 1 to CTRL[MMU_EN].

44.6 Memory map and register definition

44.6.1 MMU register descriptions

44.6.1.1 MMU memory map

MMU0 base address: 4003_0000h

MMU1 base address: 4003_1000h

MMU2 base address: 400A_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	MMU Control (CTRL)	32	RW	0000_0000h
4h	MMU Status (STS)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
8h	Interrupt Enable (INTEN)	32	RW	0000_0000h
Ch	Address Range (ADDR_RANGE)	32	RW	0000_0000h
10h	Fail Information (FAIL_INFO)	32	R	0000_0000h
14h	Fail Address (FAIL_ADDR)	32	R	0000_0000h
18h	LUT Write Data (LUT_WDATA)	32	W	0000_0000h
1Ch	LUT Read Data (LUT_RDATA)	32	R	0000_0000h
20h	Write Pointer (WRITE_PTR)	32	RW	0000_0000h
24h	Read Pointer (READ_PTR)	32	RW	0000_0000h
28h	Read Verify (READ_VFY)	32	RW	0000_0000h

44.6.1.2 MMU Control (CTRL)

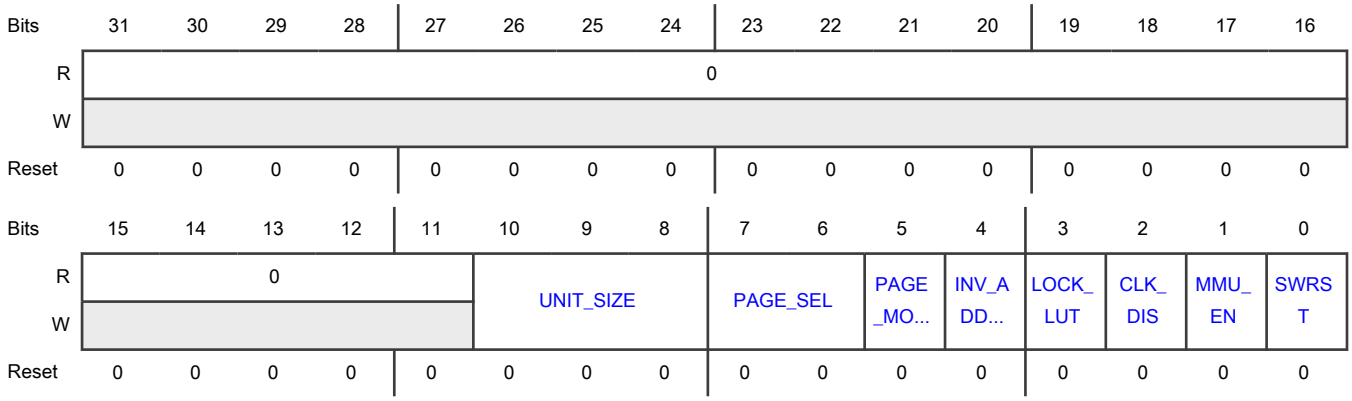
Offset

Register	Offset
CTRL	0h

Function

Provides the global settings of the module. This register is write-protected by the security component. See chip-specific information for details.

Diagram



Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-11
—</td><td>Reserved</td></tr>
<tr><td>10-8
UNIT_SIZE</td><td>Unit Size
Specifies the basic size of the unit used in the memory map.
NOTE
The descriptions of the field settings vary by module instance.
<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>MMU0
MMU1</td><td>000b - Not supported, must configure to other value
001b - 8 KB
010b - 16 KB
011b - 32 KB
100b-111b - 64 KB</td></tr><tr><td>MMU2</td><td>000b - 4 KB
001b - 8 KB
010b - 16 KB
011b - 32 KB
100b-111b - 64 KB</td></tr></table></td></tr>
<tr><td>7-6
PAGE_SEL</td><td>Page Select
Specifies the page select that is used in the memory map.
00b - Uses LUT SRAM low 2 KB words for address remapping
01b - Uses LUT SRAM mid-low 2 KB words for address remapping
10b - Uses LUT SRAM mid-high 2 KB words for address remapping
11b - Uses LUT SRAM high 2 KB words for address remapping</td></tr>
<tr><td>5
PAGE_MODE</td><td>Page Mode
Specifies the behavior of page mode.
0b - MMU takes LUT SRAM as 1 page for address remapping
1b - MMU takes LUT SRAM as 4 pages for address remapping</td></tr>
<tr><td>4
INV_ADDR_MODE</td><td>Invalid Address Mode
Specifies MMU behavior when an invalid address is detected. If this field is 0, MMU responds with a transfer error when an invalid address is detected. The transfer address is recorded in Fail Address (FAIL_ADDR), and STS[INVALID_ADDR] is 1. If this field is 1, MMU allows transfer when</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
	an invalid address is detected. The transfer address is recorded in Fail Address (FAIL_ADDR), and STS[INVALID_ADDR] is 1. 0b - Respond with a transfer error 1b - Allow transfer
3 LOCK_LUT	Lock LUT Locks LUT SRAM to avoid an accidental update. 0b - Unlock 1b - Lock
2 CLK_DIS	Clock Gate Disables the MMU clock for internal logic. 0b - Enable 1b - Disable
1 MMU_EN	MMU Enable Enables MMU memory map. 0b - Disable 1b - Enable
0 SWRST	Software Reset Specifies whether the MMU logic is reset. Writing 1 to this field resets the MMU logic. This field does not clear itself; therefore, when writing 1, you must allow a sufficient delay (4–6 NOP cycles) for reset propagation before writing 0 to disable the software reset. Asserting the software reset does not clear the registers. 0b - No reset 1b - Reset MMU logic

44.6.1.3 MMU Status (STS)

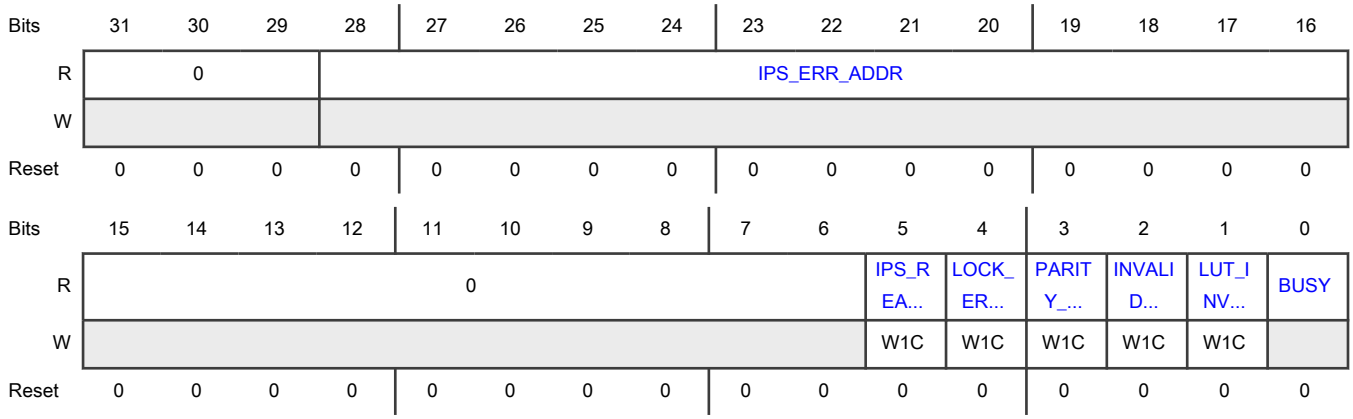
Offset

Register	Offset
STS	4h

Function

Provides interrupt fields that indicate the module status.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 IPS_ERR_ADDR	IPS Error Address Indicates the LUT SRAM address when an error is detected for IPS read operation.
15-6 —	Reserved
5 IPS_READ_ERROR	IPS Read Error Indicates whether an IPS read error is detected when writing to LUT Read Data (LUT_RDATA). This includes IPS read parity error and read-verify mismatch error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
4 LOCK_ERROR	Lock Error Indicates whether a lock error is detected when writing to LUT Write Data (LUT_WDATA). **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
3 PARITY_ERROR	Parity Error Indicates whether a parity error is detected in the access address. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
2 INVALID_ADDR	Invalid Address Indicates whether the AHB access address is out of the valid address range. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Within range 1b - Out of range When writing 0b - No effect 1b - Clear the flag
1 LUT_INVALID	LUT Invalid Indicates whether the valid bit in access address is 0, which means the access operation is invalid in LUT. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Valid 1b - Invalid When writing 0b - No effect 1b - Clear the flag
0 BUSY	Busy Indicates whether MMU is in a busy state. Wait for it to clear before taking action. Following are the possible reasons for this field to become 1: • Initialization sequence • Burst-write operation • Read-verify operation 0b - Not busy 1b - In initialization sequence, burst-write operation, or read-verify operation

44.6.1.4 Interrupt Enable (INTEN)

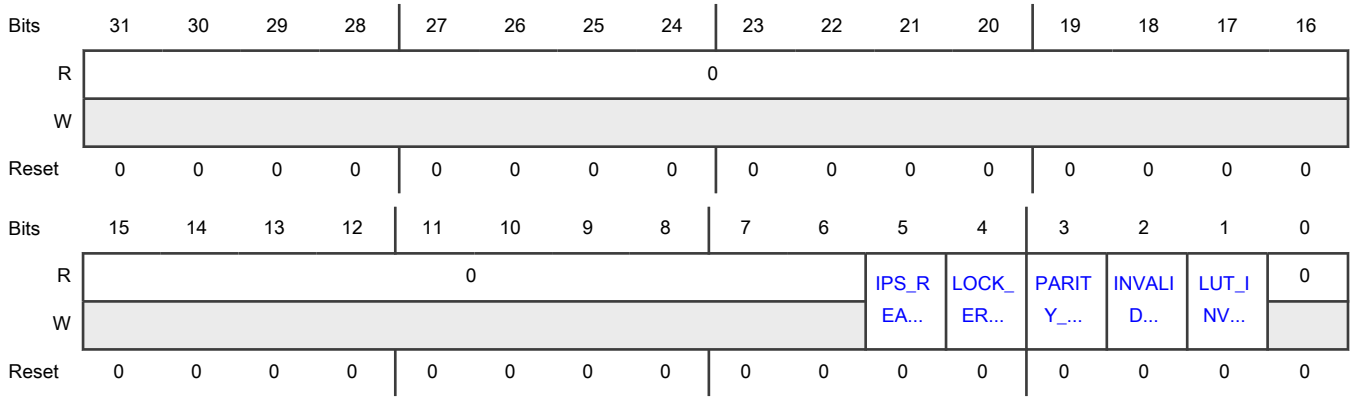
Offset

Register	Offset
INTEN	8h

Function

Enables the MMU interrupt fields.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 IPS_READ_ERROR_EN	IPS Read Error Enable Enables the interrupt for STS[IPS_READ_ERROR]. 0b - Disable 1b - Enable
4 LOCK_ERROR_EN	Lock Error Enable Enables the interrupt for STS[LOCK_ERROR]. 0b - Disable 1b - Enable
3 PARITY_ERROR_EN	Parity Error Enable Enables the interrupt for STS[PARITY_ERROR]. 0b - Disable 1b - Enable
2 INVALID_ADDR_EN	Invalid Address Enable Enables the interrupt for STS[INVALID_ADDR]. 0b - Disable 1b - Enable
1 LUT_INVALID_EN	LUT Invalid Enable Enables the interrupt for STS[LUT_INVALID]. 0b - Disable 1b - Enable
0 —	Reserved

44.6.1.5 Address Range (ADDR_RANGE)

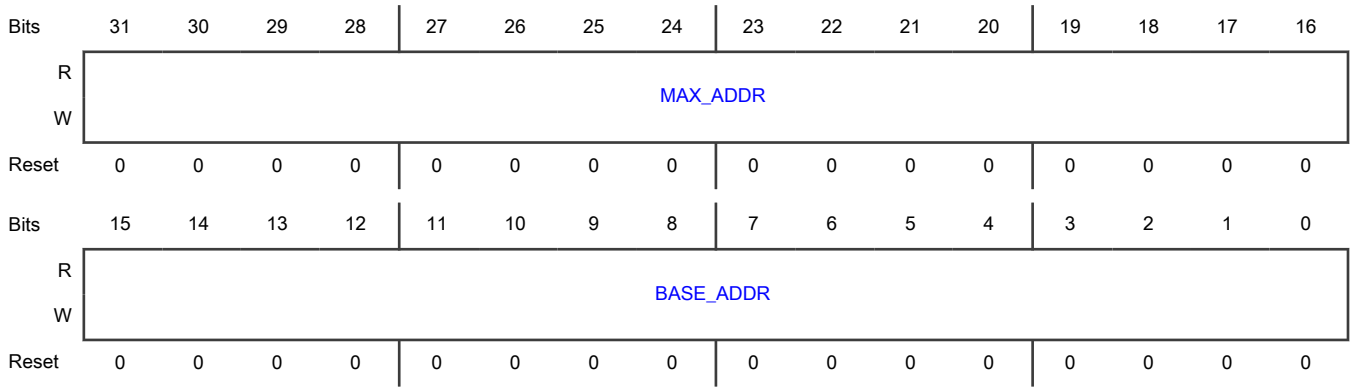
Offset

Register	Offset
ADDR_RANGE	Ch

Function

Specifies the upper 16 bits of valid address range definitions. Only the addresses between ADDR_RANGE[BASE_ADDR] and ADDR_RANGE[MAX_ADDR] are valid. For example, if ADDR_RANGE is 511F_5100h, only the AHB addresses from 5100_0000h to 511F_FFFFh are valid. This register is write-protected by a security component.

Diagram



Fields

Field	Function
31-16 MAX_ADDR	Maximum Address Specifies the upper 16 bits of the maximum address.
15-0 BASE_ADDR	Base Address Specifies the upper 16 bits of the base address.

44.6.1.6 Fail Information (FAIL_INFO)

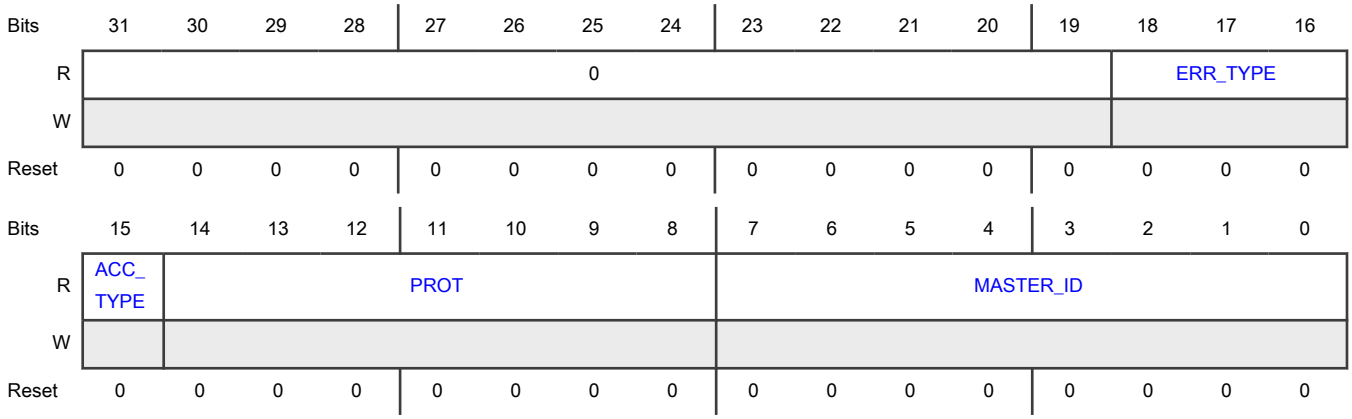
Offset

Register	Offset
FAIL_INFO	10h

Function

Indicates the transaction information when an error occurs. This register records the transfer information from the initiator side.

Diagram



Fields

Field	Function
31-19 —	Reserved
18-16 ERR_TYPE	Error Type Indicates the error type information of the transfer. 1xxb - Parity error x1xb - Invalid address error xx1b - LUT invalid error
15 ACC_TYPE	Access Type Indicates the access type of the transfer. 0b - Read 1b - Write
14-8 PROT	Protection Indicates the AHB HPROT state when an error occurs.
7-0 MASTER_ID	Initiator ID Indicates the initiator ID of the transfer.

44.6.1.7 Fail Address (FAIL_ADDR)

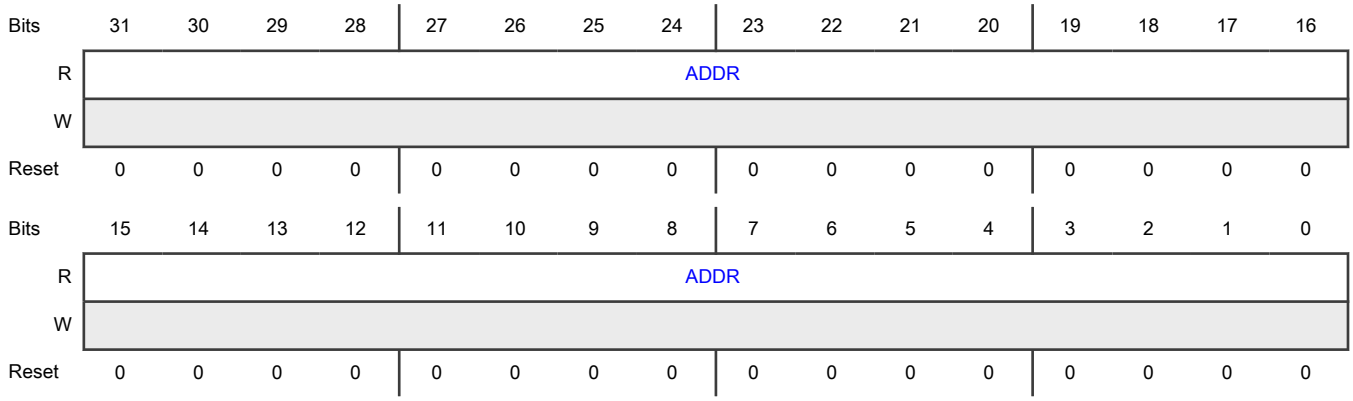
Offset

Register	Offset
FAIL_ADDR	14h

Function

Indicates the LUT address where an error occurs. This register records the AHB address from the initiator side.

Diagram



Fields

Field	Function
31-0 ADDR	Address Indicates the fail address.

44.6.1.8 LUT Write Data (LUT_WDATA)

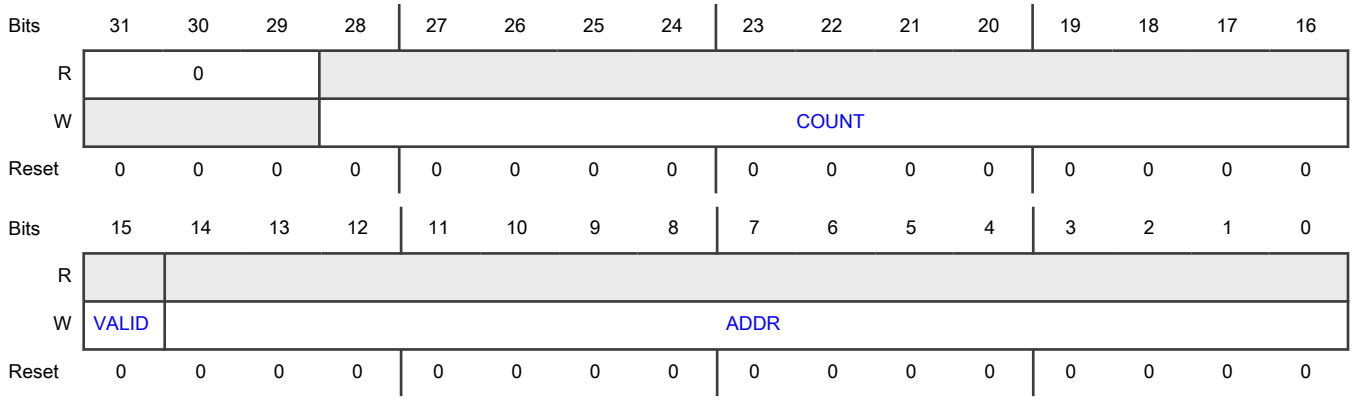
Offset

Register	Offset
LUT_WDATA	18h

Function

Specifies the write data to LUT SRAM that is addressed by Write Pointer (WRITE_PTR). This register is write-protected by a security component.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 COUNT	Write Count Specifies the write count to LUT SRAM. Writing to LUT Write Data (LUT_WDATA) triggers COUNT + 1 write operations to LUT SRAM, incrementing Write Pointer (WRITE_PTR) and the data written each time. When COUNT is 0, it triggers a write to LUT SRAM.
15 VALID	LUT Address Valid Specifies whether the address in the LUT is valid. 0b - Not valid 1b - Valid
14-0 ADDR	LUT Address Map Bits Specifies the address mapped to LUT SRAM.

44.6.1.9 LUT Read Data (LUT_RDATA)

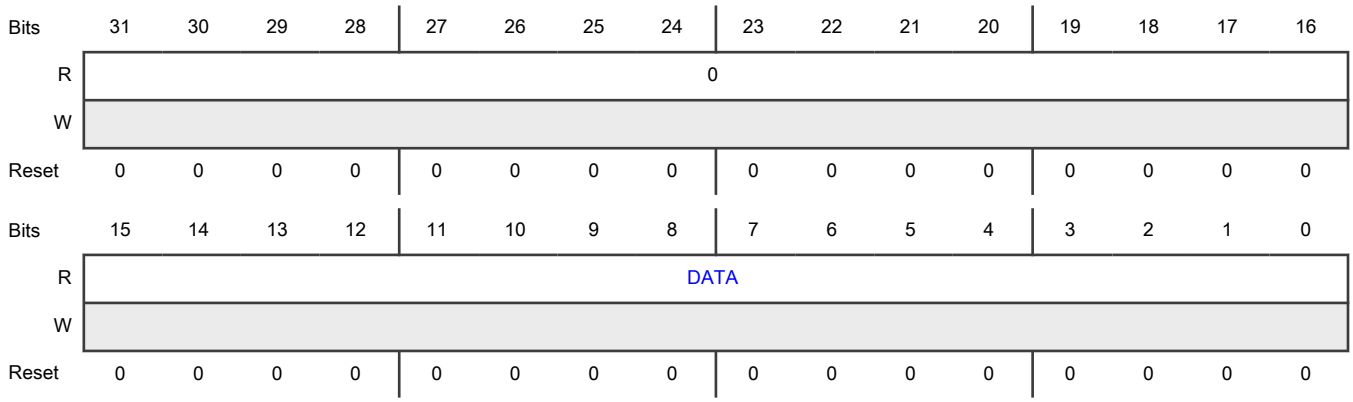
Offset

Register	Offset
LUT_RDATA	1Ch

Function

Indicates the read data from LUT SRAM addressed by Read Pointer (READ_PTR).

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Read Data Indicates the read data from LUT SRAM.

44.6.1.10 Write Pointer (WRITE_PTR)

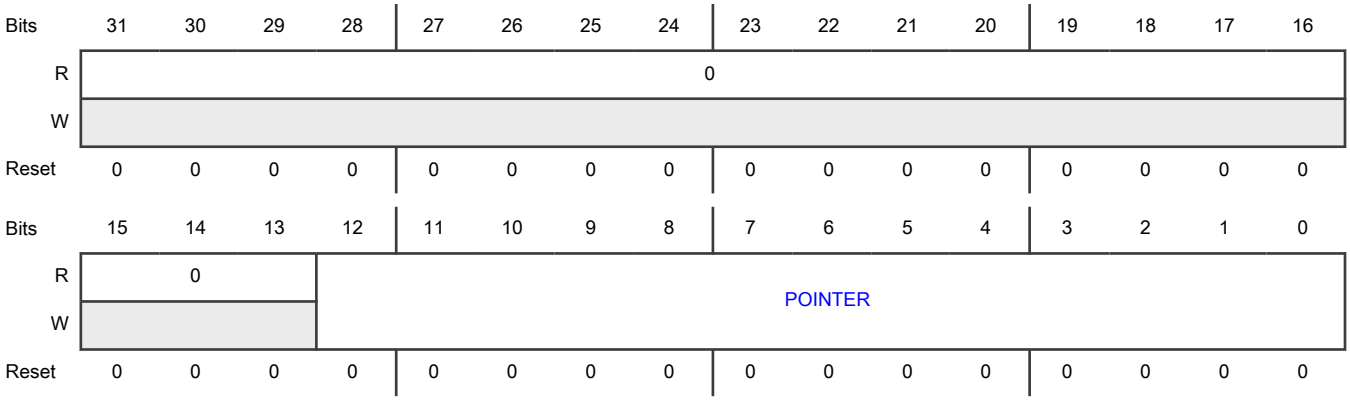
Offset

Register	Offset
WRITE_PTR	20h

Function

Specifies the write pointer for a write operation initiated by writing to LUT Write Data (LUT_WDATA).

Diagram



Fields

Field	Function
31-13 —	Reserved
12-0 POINTER	Write Pointer Specifies the write pointer of LUT SRAM. It is automatically increased by 1 when writing to LUT Write Data (LUT_WDATA).

44.6.1.11 Read Pointer (READ_PTR)

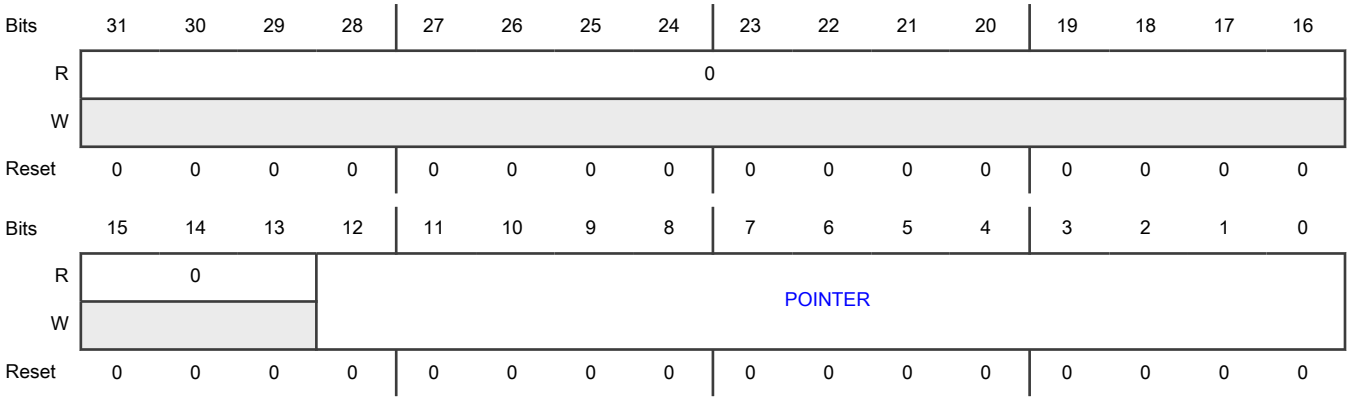
Offset

Register	Offset
READ_PTR	24h

Function

Specifies the read pointer for a read operation initiated by reading from LUT Read Data (LUT_RDATA).

Diagram



Fields

Field	Function
31-13 —	Reserved
12-0 POINTER	Read Pointer Specifies the read pointer of LUT SRAM. It is automatically increased by 1 when reading from LUT Read Data (LUT_RDATA).

44.6.1.12 Read Verify (READ_VFY)

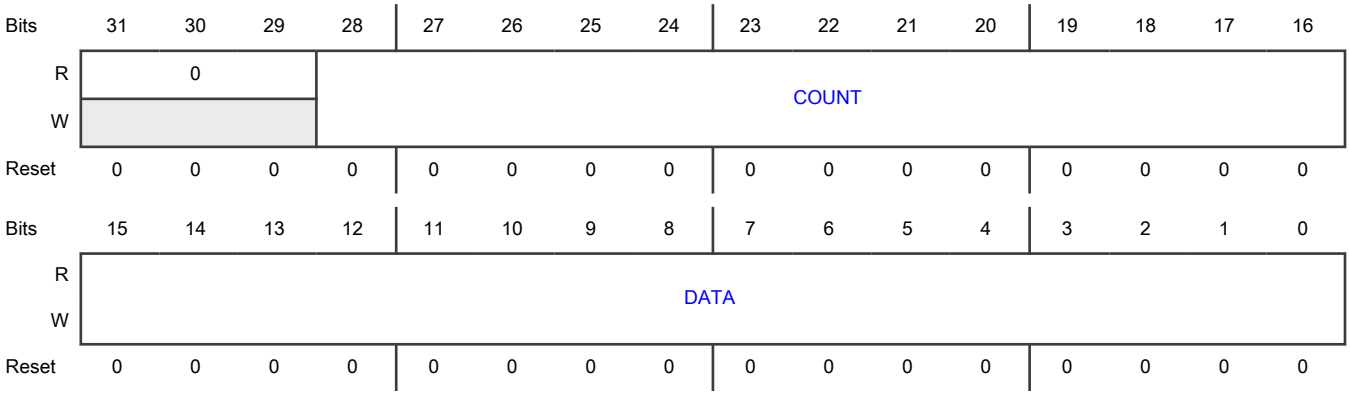
Offset

Register	Offset
READ_VFY	28h

Function

Specifies the golden data and count for read-verify check from LUT SRAM that is addressed by Read Pointer (READ_PTR).

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 COUNT	Read Count Specifies the read count to LUT SRAM. Writing to Read Verify (READ_VFY) triggers COUNT + 1 read-verify operations to LUT SRAM. This operation reads LUT SRAM and compares it against DATA. This increments Read Pointer (READ_PTR) and DATA each time. When COUNT is 0, it triggers one read-verify operation.
15-0 DATA	Read Data Specifies the golden data for the read-verify operation.

Chapter 45
Neural Processing Unit (NPU)

45.1 Chip-specific NPU information

Table 424. Reference links to related information

Topic	Related module	Reference
Full description	NPU	NPU
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

45.1.1 Module instances

This device contains one instance of the NPU module.

Table 425. NPU instance

	NPU
Located in	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

45.2 Overview

This chapter provides information on the features supported by the eIQ® Neutron N3-64 NPU. For enablement details and constraints, see the documentation that is part of eIQ Toolkit at http://nxp.com/eiq.

The NPU is an on-chip hardware accelerator that provides efficient processing of machine-learning workloads. It works in conjunction with the chip's CPU to deliver higher performance and lower power consumption for neural network (NN) inference.

In addition to managing Multiply-Accumulate (MAC) compute blocks throughput, NPU also:

- Optimizes arrangement of weights
- Manages memory accesses
- Provides rescaling, activation, and pooling functions

Enablement support for the eIQ Neutron N3-64 NPU is provided with the eIQ Toolkit and eIQ Machine Learning (ML) Software Development Environment. Some of the features and operators listed in this chapter may not be enabled in eIQ software. See References for links to eIQ documentation that lists the features and operators enabled in the latest eIQ software.

You can import your trained and quantized NN into eIQ Toolkit. The eIQ tooling generates an NPU-optimized model that can be integrated into an ML application running on the target device.

45.2.1 Block diagram

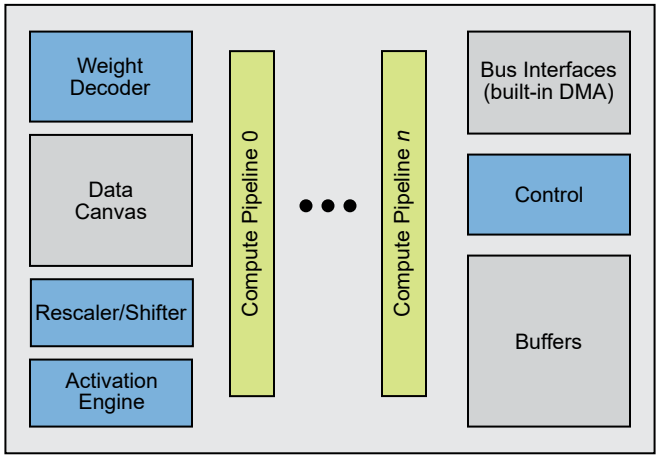


Figure 166. eIQ Neutron NPU architecture

Table 426.

Component	Description
Activation Engine	Provides activation functions. See the list of supported activation functions.
Buffers	Data buffers for pipeline input data and MAC results.
Bus Interfaces	Provide Interconnect fabric designed for efficient data transfer between memories and compute blocks.
Control	Provides control and status functions.
Data Canvas	Provides 2D circular data buffer that enables data reuse among compute pipelines.
Rescaler/Shifter	Provides post-scaling of MAC compute block results. See Rescale support.
Weight Decoder	Manages decompression, if necessary, for weight fetches. See Weight decode support.

45.2.2 Features

- Performance:
  - Up to 64 MACs/cycle for 8-bit × 8-bit operations
  - Up to 32 MACs/cycle for 8-bit × 16-bit operations
  - Up to 4 MACs/cycle for 16-bit × 16-bit operations
- Supported data types:
  - Input data:
    - 8-bit unsigned asymmetric
    - 8-bit signed asymmetric
    - 8-bit signed symmetric
    - 16-bit signed symmetric
    - 16-bit unsigned symmetric
  - Weights:

    - 8-bit signed symmetric
    - 8-bit unsigned symmetric
    - 16-bit signed/unsigned (only for fully connected operator element-wise multiplication)
  - Arithmetic operations:
    - 8-bit data × 8-bit weight
    - 16-bit data × 8-bit weight (at 1/2 peak performance of pure 8-bit)
    - 16-bit data × 16-bit weight (at 1/16 peak performance of pure 16-bit)
  - Activation × Activation (transformers, for example):
    - 8-bit symmetric × 8-bit symmetric (signed or unsigned)
    - symmetric or asymmetric weights and data for 8bit dynamic tensors
  - Bias:
    - Custom FP16 format (1-bit sign, 5-bit up-shift, 10 bits significant value)
  - Results:
    - 32-bit symmetric signed
    - 16-bit symmetric signed/unsigned
    - 8-bit symmetric/asymmetric signed/unsigned
- Supported operators: see Operator summary.
- Custom activation function f(x). Implement through Activate Function Interpolator (piece-wise polynomial interpolator):
  - Implement TFLite operators: RSQRT, EXP (for x>0), SWISH, ABS, and so on.
  - Cannot be fused with convolutions, except for fully connected and 16 × 16
  - Trade-off in accuracy versus speed and memory size (more information below)
- Constraints for Spatial Compute (Convolution, for example):
  - Padding. Individually selectable for Top/Bottom/Left/Right; maximum padding is 31 pixels.
  - Stride. Up to 31 pixels (horizontal and vertical).
  - Dilation. Up to 31 pixels (horizontal and vertical).
  - Kernel size. Maximum weight matrix row length = 64 k.
  - Kernel height. Maximum = 64 pixels (Includes pixels skipped due to dilation).
  - Kernel width. Maximum = 63 pixels (Includes pixels skipped due to dilation).

NOTE

Larger kernel-shapes, strides, and so on, can be achieved through cascading multiple layers.

- Accumulation support:

  eIQ Neutron N3-64 NPU uses a 32-bit accumulation scheme, which prevents saturation—even in large neural networks—for 8-bit × 8-bit computations.

  For 8-bit × 16-bit computations, eIQ Neutron NPU offers an accumulator downshift, configurable to 0, 4, 8, or 12 bits. This enables the data to keep sufficient precision with the tradeoff of losing some of the lower bits, to avoid saturation of the accumulator.
- Weight decode support:
  - 8-bit weights direct (no decode)
  - 16-bit weights direct (no decode)

- Mini-weights (<8 bits) through sign extension
- Mini-weights (≤6 bits). Weights are translated using a codebook. You can use weight clustering to have fewer used 8-bit values and then encode the chosen values by using an index table. NPU uses the weights as indices to select appropriate 8-bit values as the weights are loaded into compute.

- Rescale support:

  Rescales 32-bit accumulator results.

  - In a typical case in which the quantizer can rely on the integer precision dynamic range to perform the ReLU clipping and max value clipping, this function scales values to the desired 8-bit or 16-bit output directly.
  - Rescaling is also used to scale the accumulator result so it can be presented as 16-bit data to NPU's Activation Engine.

  The general form of the rescaling stage is:

```
Output = Round((accumulator × scaleFactor)>>downshiftFactor)
```

  eIQ Neutron N3-64 NPU supports the following rescaling schemes:

  - 32-bit rescale factor: 16-bit scaler, 6-bit downshift factor
  - 16-bit rescale factor: 10-bit scaler, 6-bit downshift factor
  - FP32 rescale factors

- Activation Engine:

  eIQ Neutron N3-64 NPU provides built-in activation functions and a hardware engine to perform activation. The Activation Engine works on 16-bit signed data produced by rescaling.

  The activation functions are executed “in-pipeline” together with the computation preceding it. No extra cycles are consumed and there is no need for spilling higher precision data into memory than the activated data size.

  Activation engine accuracy is 8-bit precision, except for ReLU, which essentially has no modification to values, and exp(x), which is output as 16-bit data.

  The supported built-in activation functions are:

  - 8-bit:
    - ReLU
    - ReLUx
    - Sigmoid
    - Tanh
    - LeakyReLU (limited to Alpha = 0.25, 0.125, and 0.0625)
  - 16-bit:
    - ReLU
    - ReLUx
    - LeakyReLU (8-bit precision for scaling LeakyReLU data)
    - exp(x) (For negative input values—this function is only valid for Softmax calculations)

  The second activation method is the use of the Activation Function Interpolator, which enables you to program your own custom activation functions.

  eIQ Neutron NPU supports first, second, third, and fourth order polynomial-based interpolation, enabling you to configure your own custom activation function, or to create an application-specific version of any activation function.

  This feature is controlled by a table that contains the desired number of segments (minimum is 16, maximum is 32k). For each segment, you assign 16-bit signed coefficients for computing the interpolated value. The data input to the interpolator

is 16 bits. The segment coefficients determine how many of the most significant bits of the data are used to decode the polynomial segment.

The interpolator enables you to trade-off accuracy for speed and memory footprint. It can theoretically represent any function at any accuracy (up to full/bit-exact 16 bit), by increasing the number of table entries and/or the interpolation order.

Note that an interpolator-based implementation of a function has a maximum throughput of at most 1 result every *n+1* cycles, where *n* is the maximum order of the used polynomial fit. In contrast, the built-in activations functions provide up to 8 results per cycle and are therefore much faster.

45.3 Functional description

45.3.1 Operator summary

Table 427. Operator summary

Operator[1]	TFLite operator	Data size (bits)		Notes
		Input[2]	Output[3]	
Absolute	ABS	8, 16	8, 16	
Add Tensor-Tensor	ADD	8, 16	8, 16	Includes scaling to match quantization scales; both inputs must be of equal bit width
Add Tensor + Constant	ADD	8, 16	8, 16	
Arg-Max	ARG_MAX	8,16	8,16	
Arg-Min	—	8,16	8,16	
Average Pooling	AVERAGE_POOL_2D	8, 16	8, 16	
Concatenate Tensors	CONCATENATE	8, 16	8, 16	
Convolution 2D/1D	CONV_2D	8, 16	8, 16	
Depthwise Convolution 2D/1D	DEPTHWISE_CONV_2D	8, 16	8, 16	
ELU	ELU	8, 16	8	As fused activation
Exponential function e^x	EXP	8, 16	8, 16	• Constraint: for numerical stability only on x≤0 (for positive x, use interpolator) • Any other a^x can be implemented via $e^{log(a)*x}$ using the rescale hardware
Expand Dimensions	EXPAND_DIMS	N/A	N/A	Supported by Activate Function Interpolator
Fully Connected / Dense / Linear	FULLY_CONNECTED	8, 16	8, 16	Support for 8- or 16-bit weights and data of 8- or 16-bit data
Hard-Swish	HARD_SWISH	8, 16	8	As fused activation function or TFLite operator HARD_SWISH

Table continues on the next page...

Table 427. Operator summary (continued)

Operator[1]	TFLite operator	Data size (bits)		Notes
		Input[2]	Output[3]	
LeakyReLU	LEAKY_RELU	8, 16	8, 16	• As fused activation function or TFLite operator LEAKY_RELU • 8-bit precision for scaling LeakyReLU data
Logistic / Sigmoid	LOGISTIC	8, 16	8	As fused activation or TFLite operator LOGISTIC
Maximum	MAXIMUM	8, 16	8, 16	
Max-Pooling	MAX_POOL_2D	8, 16	8, 16	
Mean / Average	MEAN	8, 16	8, 16	
Minimum	MINIMUM	8, 16	8, 16	
Multiplication	MUL, BatchMatMul	8, 16	8, 16	• Support for all types of multiplications—batched, non-batched, dot-product matrix-matrix, matrix-scalar, transpose matrix-matrix, Hadamard • 8-bit data × 8-bit weight • 16-bit data × 8-bit weight
Packing	PACK	8, 16	8, 16	
Padding	PAD	8, 16	8, 16	Implicit with compute
PRELU	PRELU	8, 16	8, 16	As fused activation or TFLite operator PRELU
Re-Quantize	QUANTIZE	8, 16	8, 16	• To change INT quantization scale and zero point • Supported by Activate Function Interpolator • Integer input, integer output
RELU	RELU	16	8, 16	As fused activation or TFLite operator RELU
RELU6	RELU6	8, 16	8, 16	• As fused activation or TFLite operator RELU6 • Supports min-max clipping factors other than (0, 6)
RELU	RELU_N1_TO_1	8, 16	8, 16	As fused activation or TFLite operator RELU_N1_TO_1
Reshape Tensor	RESHAPE	8, 16	8, 16	

Table continues on the next page...

Table 427. Operator summary (continued)

Operator[1]	TFLite operator	Data size (bits)		Notes
		Input[2]	Output[3]	
Resize Bilinear	RESIZE_BILINEAR	8, 16	8, 16	Constraint: Only 2×2. Larger resize factors can be achieved by cascading multiple 2×2 resize layers.
Resize Nearest Neighbor	RESIZE_NEAREST_NEIGHBOR	8, 16	8, 16	Constraint: Only 2×2. Larger resize factors can be achieved by cascading multiple 2×2 resize layers.
SELU	SELU	8, 16	8	
Shape	SHAPE	N/A	N/A	No operation is performed by NPU hardware
Slice Tensor	SLICE, STRIDED_SLICE	8, 16	8, 16	
SoftMax	SOFTMAX	8	8	• Constraint: Only on 8-bit accuracy • Reciprocal of sum of e^x offloaded to CPU
Split Tensor	SPLIT, SPLIT_V	8, 16	8, 16	
Squeeze Tensor	SQUEEZE	N/A	N/A	Operation is not performed by NPU hardware
Strided Slice	STRIDED_SLICE	8, 16	8, 16	
Subtract Tensor-Tensor	SUB	8, 16	8, 16	• Includes scaling to match quantization scales • 8-bit data × 8-bit weight • 16-bit data × 8-bit weight
Subtract Tensor-Constant	SUB	8, 16	8, 16	• Includes scaling to match quantization scales • 8-bit data × 8-bit weight • 16-bit data × 8-bit weight
Tanh / hyperbolic tangent function	TANH	8, 16	8	As fused activation or TFLite operator TANH
Transpose Convolution 2D	TRANSPOSE_CONV	8, 16	8, 16	Includes possibility of upsampling input (zero-filled) before operation
Unidirectional LSTM	UNIDIRECTIONAL_SEQUENCE_LSTM	8, 16	8, 16	
Unpack Tensor	UNPACK	N/A	N/A	No operation is performed by NPU hardware

1. Some of the operators in listed in this table may not be enabled in eIQ software. See References for links to eIQ documentation that lists the operators enabled in the latest eIQ software.
2. Inputs are always 8- or 16-bit.
3. Output can be 8-,16-, or 32-bit. Note that this is a format—it does not imply 32-bit precision, but output can be generated in int32 format with whatever data is generated. This can be useful if you want to run, for example, a convolution with the intent of running activation outside of Neutron. It is also sometimes useful to export results as 16-bit data to have symmetric data instead of 8-bit asymmetric data.

45.3.2 Clocking

This module has no clocking considerations.

45.3.3 Interrupts

This module has no interrupts.

45.3.4 References

Topic	Link
eIQ ML Software Development Environment	https://www.nxp.com/eiq
eIQ Toolkit	https://www.nxp.com/design/software/eiq-ml-development-environment/eiq-toolkit-for-end-to-end-model-development-and-deployment:EIQ-TOOLKIT

45.4 External signals

This module has no external signals.

45.5 Initialization

This module does not require initialization.

45.6 Memory map and register definitions

eIQ Neutron NPU does not have any user-configurable registers. All NPU configuration and control functions must be performed using NXP's eIQ Toolkit software. See References for links.

Chapter 46
EZ Handler RISC-V (EZH-V)

46.1 Chip-specific EZH-V information

Table 428. Reference links to related information

Topic	Related module	Reference
Full description	EZH-V	EZH-V
System memory map		Memory map overview
System Controller	SYSCON	SYSCON
Clock controller	CLKCTL	Clock controller
Sleep Controller	SLEEPCON	SLEEPCON

46.1.1 Module instances

This device has one instance of the EZH-V module.

Table 429. EZH-V instance

	EZH-V
Located in	VDD2_MEDIA

46.1.2 EZH-V related registers in SYSCON

You can use SYSCON registers to configure the following EZH-V features.

Table 430. EZH-V related registers in SYSCON

Registers	Descriptions
SYSCON4.EZHV_MEM_CTRL	Configures input and power gating for EZH-V memory.
SYSCON4.EZHV2ARM_INT_EN	Controls EZH-V to Arm interrupts.
SYSCON4.EZHV2ARM_INT_CHAN	indicates EZH-V interrupt channel.
SYSCON4.ARM2EZHV_INT_CTRL	Controls Arm To EZH-V interrupts.
SYSCON4.EZHV_RSTBASE	Allows system integrators to be able to specify and relocate the base address of the reset exception handler. These inputs are used to form the upper 30 bits of the instruction access following negation of reset which is used to fetch the initial instruction of the reset exception handler.
SYSCON4.EZHV_RSTMTVEC	Allows system integrators to be able to specify and relocate the base address of the exception vector table at reset. These inputs are used to initialize MTVEC on reset to a value appropriate for the system in order to support certain exceptions which could occur prior to software being able to configure the MTVEC register.
SYSCON4.EZHV_RSTSTVEC	Allows system integrators to be able to specify and relocate the base address of the exception vector table at reset. These inputs are used to initialize STVEC on reset to a value appropriate for the system in order to support certain exceptions which could occur prior to software being able to configure the STVEC register.

Table continues on the next page...

Table 430. EZH-V related registers in SYSCON (continued)

Registers	Descriptions
SYSCON4.EZHV_VOFFSET_T	Provides a vector offset to be used when exception processing begins for an incoming interrupt request.

46.2 Introduction

The EZH-V is primarily implemented to provide a “smart DMA” engine for post-processing graphics data assembled by either or both the GPU and CPU, and then passing it to the FLEXIO or MIPI DSI for output. Processing operations may include data packing and endianness adjustment. The EZH-V is available for any other general-purpose handling that may be required and supports an extensive selection of trigger inputs from GPIOs and most on-chip peripherals.

The EZH-V is a 4-stage little-endian single-issue 32-bit RISC-V Zen-V core. The ISA of EZH-V was extended to support multiple additional instructions that provide hardware multiply and divide capability, improved code density, a separate address space of Control and Status registers, and additional instructions for floating point embedded numerics and bit manipulation that are not part of the standard RISC-V ISA.

46.3 Features

The EZH-V supports the following instructions:

- Full support for RV32I Base Integer Instruction Set
- Full support for RV32C Standard Extension for Compressed Instructions
- Full support for RV32M Integer Multiplication and Division Instruction Set Extension
- Specific extensions
  - Zicsr, which supports 4096 control and status registers (CSR), as well as instructions to atomically read-modify-write a single CSR
  - Zfinx, which provides for support real-time single-precision floating-point embedded numerics operations using the general-purpose registers.
  - Zbb, which enables bit manipulation to enable code size reduction, performance improvement, and energy reduction.

With the assistance of the EZH-V, the Cortex-M33 cores can be freed to perform other tasks. While the EZH-V is executing the assigned task, the CM33 cores can execute other tasks in parallel.

46.4 Block diagram

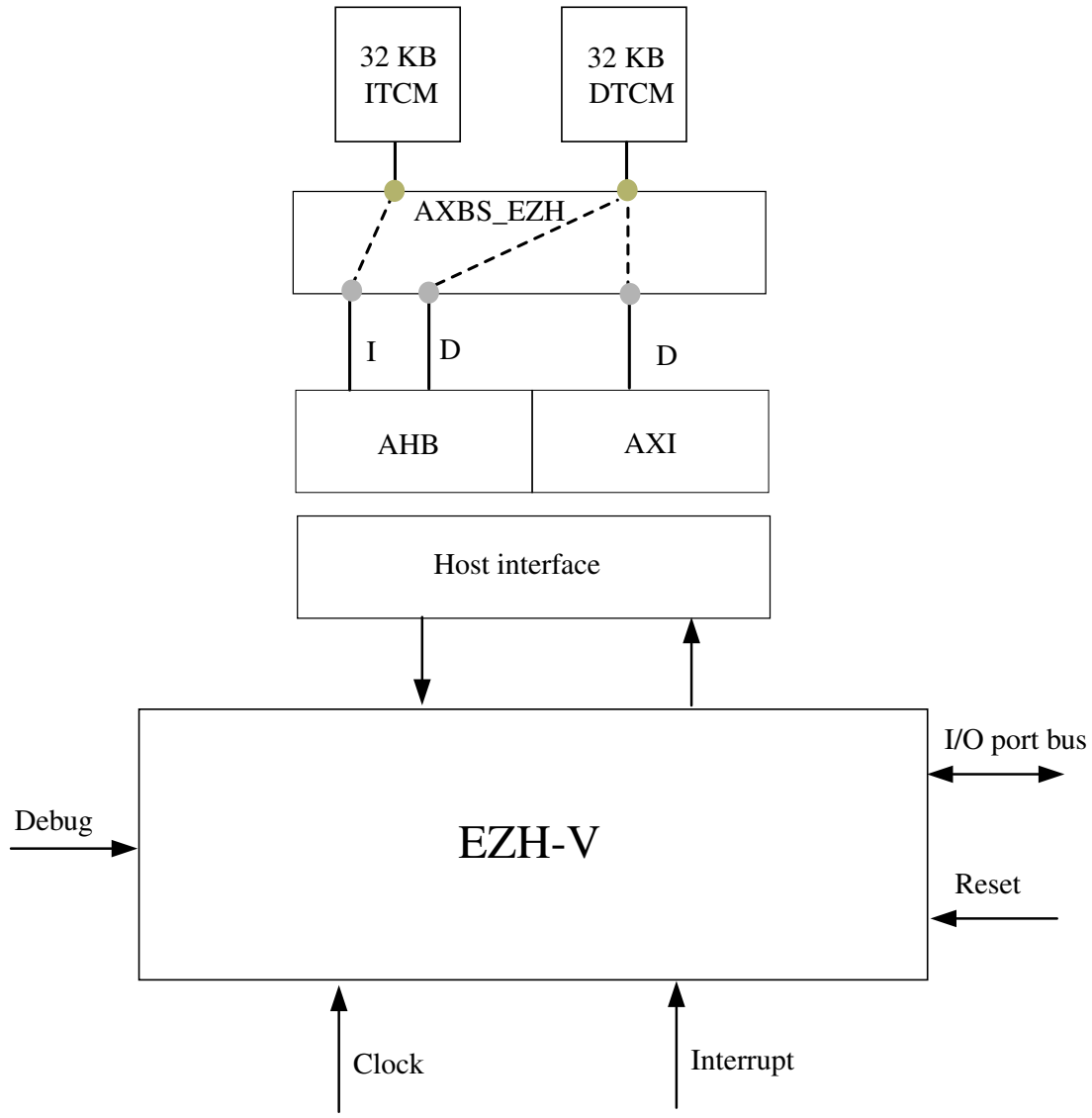


Figure 167. EZH-V block diagram

46.5 Functional description

46.5.1 Host and memory interface

The EZH-V has two dedicated Tightly Coupled Memories (TCM):

- 32 KB Instruction TCM
- 32 KB Data TCM

It also has three bus interfaces:

- Single 32-bit AHB system interface that provides connections to peripherals, DTCM, ITCM, system memory
- Single 64-bit AXI system interface that provides connections to peripherals, DTCM, and system memory
- I/O port bus interfacing to the GPIO

46.5.2 Clocking

The EZH-V employs a simple clocking scheme. It is derived from the system bus clock, media_main_clk.

46.5.3 Interrupts

EZH-V interrupts are available for CPU0 and CPU1 along with the HiFi4 DSP via the INPUTMUX. Additionally, this IP can also wake CPU0 and CPU1 from Deep Sleep mode.

46.5.4 Reset

This module can be reset via a domain reset (warm/cold) or software reset.

46.5.5 External signals

The GPIO port bus consists of up to 28 external signals associated with the EZH-V: 28 input/output signals, EZH_PIOn.

46.5.6 Debug

This EZH-V supports standard SWD/JTAG debug port.

Chapter 47
JPEG Decoder (JPEGDEC)

47.1 Chip-specific JPEGDEC information

Table 431. Reference links to related information

Topic	Related module	Reference
Full description	JPEGDEC	JPEGDEC
JPEG decoder wrapper	JPGDECWRP	JPGDECWRP
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

47.1.1 Module instances

This device contains one instance of the JPEGDEC module.

Table 432. JPEGDEC instance

Features	JPEGDEC
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

47.2 Overview

JPEGDEC is one of the submodule in JPGDECWRAP. In addition to decoding standard-compliant baseline and extended JPEG streams, JPEGDEC can decompress the video payload of many standard motion JPEG container formats.

NOTE

This Chapter is only feature specification. See Chapter JPGDECWRAP for more usage reference.

47.2.1 Block diagram

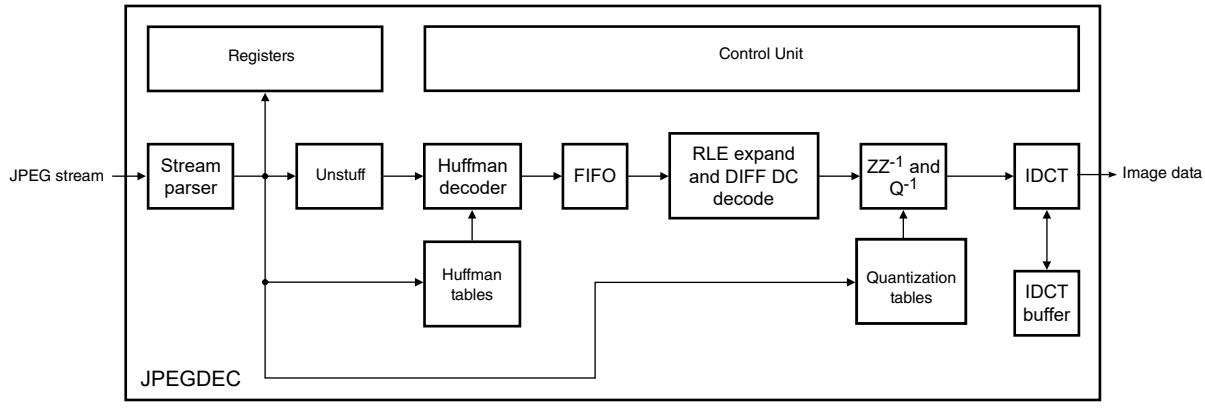


Figure 168. Block diagram

47.3 Functional description

47.3.1 Architectural overview

JPEGDEC is fully self-configured by parsing the marker segments present in the input JPEG stream.

JPEGDEC receives the JPEG stream via its JPEG_In interface. It checks input JPEG marker segments against a subset of possible errors and signals when an error is detected. You can configure the decoded image parameters to control other system peripherals, such as a block-to-raster scan converter.

JPEGDEC decodes one or more entropy-coded data segments after parsing the marker segments. Then, it outputs the decoded image samples in their native Minimum coded unit (MCU) block scan order via its Pixel-Out interface.

A single control register can control JPEGDEC, which you can access via the control interface.

47.3.2 Clocking

All JPEGDEC input and output ports operate synchronously to the input clock, except for the RST input port, which is an asynchronous reset line.

The RST asynchronous reset line drives all registers using the clock. All inputs are sampled only on the rising edge of the clock. All outputs are updated only on the rising edge of the clock.

47.3.3 Reset

JPEGDEC uses a common RST, which can be asserted at any time, but must be deasserted in synchronization with the clock.

47.3.4 Interrupts

This module has no interrupts.

47.4 External signals

This module has no external signals.

47.5 Initialization

CLK, CLR, RST, and ENABLE signals will be initialized by the JPEG decoder wrapper module.

47.6 Memory map and register definition

For JPEGDEC register information, see "JPEGDEC register descriptions" in JPEG Decoder Wrapper (JPGDECWRP) chapter.

Chapter 48
JPEG Decoder Wrapper (JPGDECWRP)

48.1 Chip-specific JPGDECWRP information

Table 433. Reference links to related information

Topic	Related module	Reference
Full description	JPGDECWRP	JPGDECWRP
JPEG decoder	JPEGDEC	JPEGDEC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

48.1.1 Module instances

This device contains one instance of the JPGDECWRP module.

Table 434. JPGDECWRP instance

Features	JPGDECWRP
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

48.2 Overview

JPGDECWRP is a high-performance JPEG decoder for still image and video decompression applications. It supports 230 MSamples/S for decode with 230 MHz clock. JPGDECWRP has DMA channels for bit stream fetching and decoded pixel data storing, as well as a register interface and chained descriptors for frame or bit stream buffer switching.

48.2.1 Block diagram

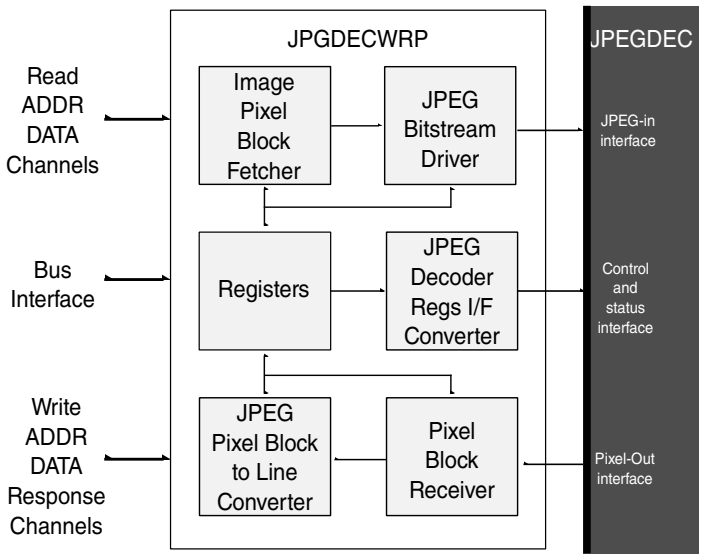


Figure 169. Block diagram

48.2.2 Features

- Baseline and extended ISO/IEC 10918-1 JPEG compliance[2]:
  - Four Huffman tables (2 DC and 2 AC coefficients tables at maximum)
  - Four 8-bit or 16-bit quantization tables (at maximum)
  - Single scan configuration and all possible JPEG formats for input data
  - Restart markers[3]
- DMA engines for fetching the JPEG bit stream from system memory and feeding it to the decoder
- DMA engines for storing the decoded pixel data into the system memory
- Conversion of the pixel MBT format to linear format
- Multi-bit-stream context switch based on frame
- Chained DMA or bit stream descriptors for frame or bit stream buffer updates with minimum software interference
- Bit stream buffer, half or full, and return features for bit stream buffer management
- YUV444, YUV420, YUV422, RGB, ARGB, and Gray formats are supported at the maximum resolution of 8k * 8k. As for minimum resolution,
  - YUV422: larger or equal to 64 * 32 (width is a multiple of 16, height is a multiple of 8).
  - YUV420: larger or equal to 64 * 64 (both width and height are a multiple of 16).
  - YUV444, RGB, ARGB, and GRAY: larger or equal to 32 * 32 (both width and height are a multiple of 8).
- 8-bit and 12-bit pixels precision

[2] See Limitations related to the ISO/IEC 10918-1 JPEG standard for limitations associated with the ISO/IEC 10918-1 JPEG standard.

[3] JPEGDEC supports decoding of noncorrupted JPEG bit streams with or without restart segments and does not support decoding of corrupted bit streams with or without restart segments.

48.3 Functional description

48.3.1 Modes and operations

Table 435 lists modes of operation.

Table 435. Modes of operation

Mode	Description	Section
Single Bit Stream and Single Frame	Decoding stops at each frame.	Single Bit Stream and Single Frame mode
Single Bit Stream Repeat	Decoding gets a new frame buffer address from the descriptor and continues.	Single Bit Stream Repeat mode
Context Switch	Switches to new bit stream or frame buffer after the current frame or bit stream decoding is completed on a frame basis. The descriptor defines a new bit stream or frame and supports up to four bit streams.	Context Switch mode

For each mode, perform the following procedures at first:

1. write 0 to CTRL[LP] and CTRL[SWR], and write 1 to CTRL[GO].

Then, follow the operation procedures of each mode.

48.3.1.1 Single Bit Stream and Single Frame mode

If SLOT*n*_NXT_DESCPT_PTR[NXT_DESCPT_EN] is 0, the wrapper is in Single Bit Stream and Single Frame mode. To start decoding, write 1 to GLB_CTRL[DEC_GO]. After the decoding of each frame is completed, the decoding pauses until you write 1 to GLB_CTRL[DEC_GO].

48.3.1.2 Single Bit Stream Repeat mode

If SLOT*n*_NXT_DESCPT_PTR[NXT_DESCPT_EN] is 1, and if GLB_CTRL[SLOT_EN] enables only one slot, JPGDECWRP works in Single Bit Stream Repeat mode. It completes the current frame decoding, then fetches the next frame buffer address from the location that the descriptor pointer defines. In this mode, the bit stream remains the same.

48.3.1.3 Context Switch mode

If SLOT*n*_NXT_DESCPT_PTR[NXT_DESCPT_EN] is 1, and if GLB_CTRL[SLOT_EN] enables more than one slot, JPGDECWRP is in Content Switch mode. It completes the current frame decoding, then fetches the next frame buffer address from the location that the descriptor pointer defines, as well as the bit stream buffer for the next bit stream. When switching, the current bit stream buffer pointer is saved and restored when the current bit stream is switched back.

In this mode, each bit stream is assigned with one bit stream ID and up to four bit streams are supported. Each bit stream context has its own interrupt and status register, and occupies one 64 KB slot.

48.3.2 Limitations related to the ISO/IEC 10918-1 JPEG standard

- Up to four image components are supported: component identifiers 0 to 3 or 1 to 4.
- Sampling factors three or above four are not supported.
- Up to four (two DC and two AC) Huffman tables are supported.
- RST markers decoding support is available:

— No corrupted, valid streams without restart markers: Yes

— No corrupted, valid streams with restart markers: Yes

— Corrupted, invalid streams without restart markers: No

— Corrupted, invalid streams with restart markers: No

48.3.3 Clocking

JPGDECWRP uses two clocks: main (global) clock and APB bus clock.

Table 436. Clock signal

Clock signal	Description
Global clock (clk)	It aligns with the "jpeg_clk" of System clocks in this chip.
APB bus clock (pclk)	Controls the access to the registers of the JPGDECWRP module.

48.3.4 Interrupts

Each slot has its interrupt status or enable registers. See JPGDECWRP register descriptions for a detailed description of these registers.

48.3.5 Reset

JPGDECWRP uses a common reset, which can be asserted at any time, but must be deasserted in synchronization with the clock. It also has software resets. You can do software reset by setting GLB_CTRL[SFTRST] and CTRL[SWR] independently.

48.4 External signals

JPGDECWRP does not have any chip external interfaces. All its interface are chip internal AXI bus, APB interface, and interrupts.

48.5 Initialization

To initialize JPGDECWRP:

1. Resets JPGDECWRP.
2. Enables clock.
3. Configures registers (see Mode and operation for more details).
4. Writes 1 to GLB_CTRL[JPG_DEC_EN].

48.6 Memory map and register descriptions

48.6.1 JPGDECWRP register descriptions

Decoding descriptors define the bit stream buffer location, output frame buffer locations, image format, and so on. JPGDECWRP supports context switching on a frame basis, which means you can load those registers from the memory that the Next Decoding Descriptor Pointer (Next Descriptors (SLOT0_NXT_DESCPT_PTR - SLOT3_NXT_DESCPT_PTR)) points to. NXT_DEC_DESCPT can support up to 4 bitstream context switching or single-bit-stream decoding with different output buffer switching on a frame-based interval.

48.6.1.1 JPGDECWRP memory map

JPGDECWRP base address: 400A_6000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Global Control (GLB_CTRL)	32	RW	0000_0000h
4h	Common Status (COM_STATUS)	32	R	0000_0000h
14h	Output Image Frame Buffer0 Base Address (OUT_BUF_BASE0)	32	RW	0000_0000h
18h	Output Image Frame Buffer1 Base Address (OUT_BUF_BASE1)	32	RW	0000_0000h
1Ch	Image Output Buffer Pitch (OUT_PITCH)	32	RW	0000_0000h
20h	Input JPEG Stream Buffer Base Address (STM_BUFBASE)	32	RW	0000_0000h
24h	Input JPEG Stream Buffer Size (STM_BUFSIZE)	32	RW	0000_2000h
28h	Image Resolution (IMGSIZE)	32	RW	0000_0000h
2Ch	Bit Stream and Switching Control (STM_CTRL)	32	RW	0000_0000h
200h	Bit Stream Status (SLOT0_STATUS)	32	RW	0000_0000h
204h	Bit Stream Interrupt Enable (SLOT0_IRQ_EN)	32	RW	0000_0000h
208h	Bit Stream Buffer Pointer (SLOT0_BUF_PTR)	32	R	0000_0000h
20Ch	Current Descriptors (SLOT0_CUR_DESCPT_PTR)	32	R	0000_0000h
210h	Next Descriptors (SLOT0_NXT_DESCPT_PTR)	32	RW	0000_0000h
400h	Bit Stream Status (SLOT1_STATUS)	32	RW	0000_0000h
404h	Bit Stream Interrupt Enable (SLOT1_IRQ_EN)	32	RW	0000_0000h
408h	Bit Stream Buffer Pointer (SLOT1_BUF_PTR)	32	R	0000_0000h
40Ch	Current Descriptors (SLOT1_CUR_DESCPT_PTR)	32	R	0000_0000h
410h	Next Descriptors (SLOT1_NXT_DESCPT_PTR)	32	RW	0000_0000h
600h	Bit Stream Status (SLOT2_STATUS)	32	RW	0000_0000h
604h	Bit Stream Interrupt Enable (SLOT2_IRQ_EN)	32	RW	0000_0000h
608h	Bit Stream Buffer Pointer (SLOT2_BUF_PTR)	32	R	0000_0000h
60Ch	Current Descriptors (SLOT2_CUR_DESCPT_PTR)	32	R	0000_0000h
610h	Next Descriptors (SLOT2_NXT_DESCPT_PTR)	32	RW	0000_0000h
800h	Bit Stream Status (SLOT3_STATUS)	32	RW	0000_0000h
804h	Bit Stream Interrupt Enable (SLOT3_IRQ_EN)	32	RW	0000_0000h
808h	Bit Stream Buffer Pointer (SLOT3_BUF_PTR)	32	R	0000_0000h
80Ch	Current Descriptors (SLOT3_CUR_DESCPT_PTR)	32	R	0000_0000h
810h	Next Descriptors (SLOT3_NXT_DESCPT_PTR)	32	RW	0000_0000h

48.6.1.2 Global Control (GLB_CTRL)

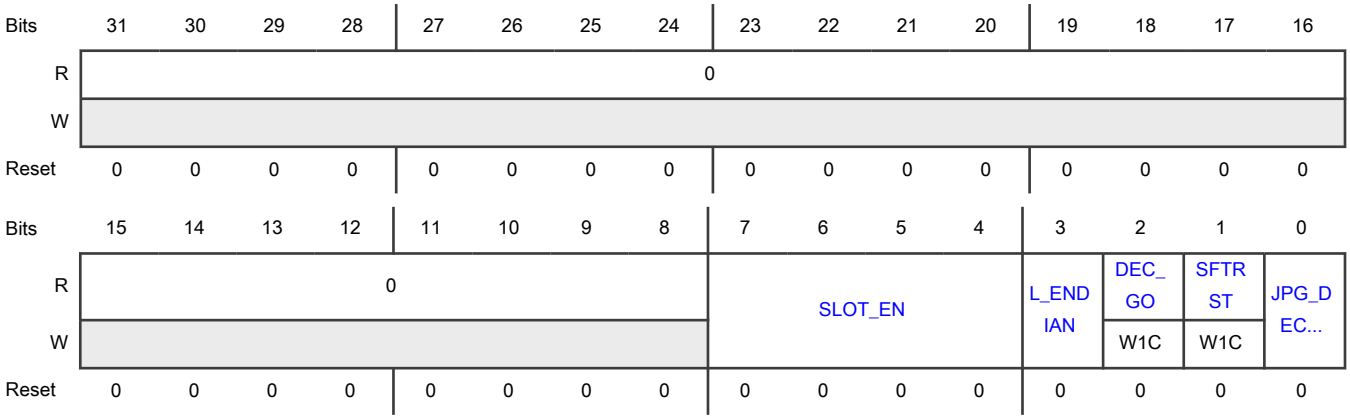
Offset

Register	Offset
GLB_CTRL	0h

Function

Provides control fields for JPGDECWRP.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-4 SLOT_EN	Slots Enable Enables specific bit stream slots for switching on a frame basis. Each bit of this field corresponds to one slot. When this slot is enabled, JPGDECWRP includes the descriptor of this slot for context switching rotation. You must first enable one slot and then update SLOT*n*_NXT_DESCPT_PTR[NXT_DESCPT_PTR] for this slot. If no valid NXT_DESCPT_PTR is available for an enabled slot, JPGDECWRP does not add this slot to the switching rotation. 0000b - Disables all slots. 0001b - Enables SLOT0. 0010b - Enables SLOT1. 0100b - Enables SLOT2. 1000b - Enables SLOT3. All other values are reserved.
3	Little-Endian Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
L_ENDIAN	Enables the little-endian bit stream format. The default bit stream format is big endian. 0b - Big-Endian 1b - Little-Endian
2 DEC_GO	Start Decoding Starts decoding on a frame basis. The decoding process stops after the frame decoding is complete, if there are no other valid descriptors available. This field becomes 0 automatically. There are two ways to start decoding: • Configure all the common registers and write 1 to this field to manually start the decoding. • Enable one slot and then write to Next Descriptors (SLOT0_NXT_DESCPT_PTR - SLOT3_NXT_DESCPT_PTR) with SLOT*n*_NXT_DESCPT_PTR[NXT_DESCPT_EN] = 1. You can use only manually-started decoding — and no other method — for debug purposes. 0b - Do not start decoding manually. 1b - Starts decoding manually.
1 SFTRST	Engine Soft Reset Resets internal engines and state machines in the submodule JPEG Decoder Wrapper. This soft reset does not reset register bits. This field becomes 0 automatically. 0b - No effect 1b - Performs a soft reset.
0 JPG_DEC_EN	JPEGDEC and JPGDECWRP Enable Enables JPEGDEC and JPGDECWRP simultaneously. 0b - Disable 1b - Enable

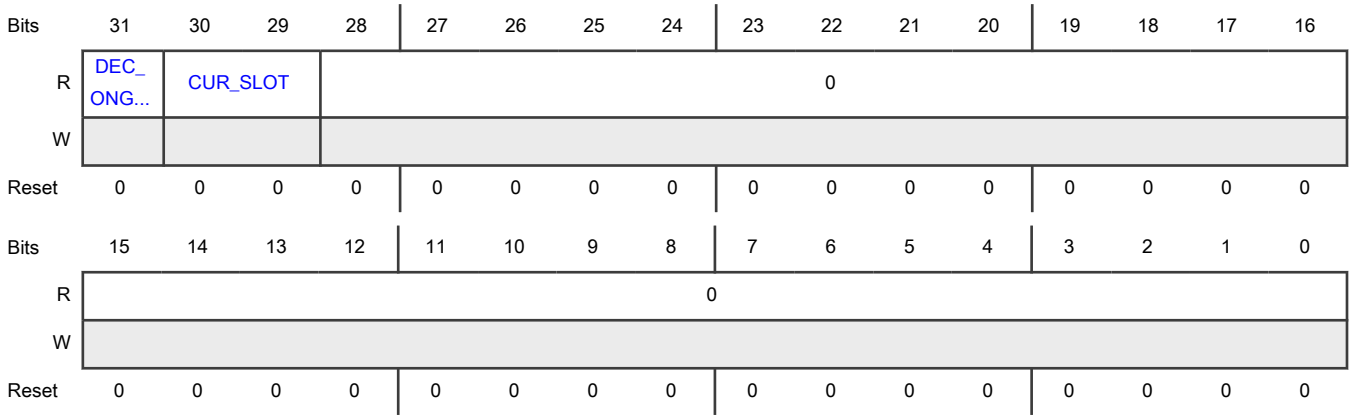
48.6.1.3 Common Status (COM_STATUS)

Offset

Register	Offset
COM_STATUS	4h

Function
Indicates the status of JPGDECWRP.

Diagram



Fields

Field	Function
31 DEC_ONGOING	Decoding Ongoing Indicates whether the decoding is ongoing. Decoding is paused or stopped usually after the decoding of one frame is completed. This field becomes 0 in the following cases: • During the slot switching or descriptor loading, when there is no bit stream for decoding • If there are no other valid descriptors on all enabled slots
30-29 CUR_SLOT	Current Slot Indicates the 2-bit ID of the current executing bit stream slot. This field becomes 0 if no slot is enabled.
28-0 —	Reserved

48.6.1.4 Output Image Frame Buffer0 Base Address (OUT_BUF_BASE0)

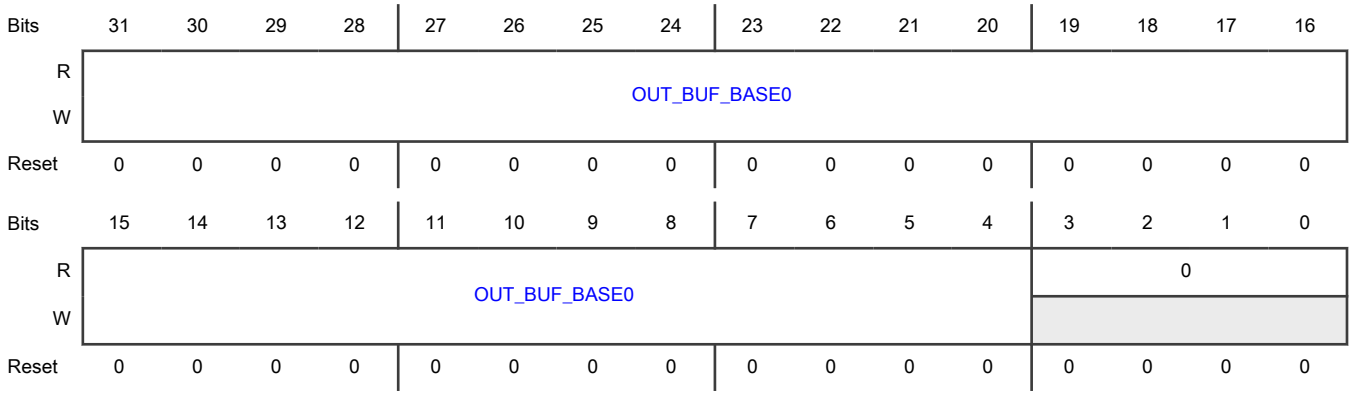
Offset

Register	Offset
OUT_BUF_BASE0	14h

Function

Specifies the base address for output frame buffer0. This register points to the beginning of the output image frame buffer. After the decoding of a frame is complete and the image is output to the memory, you can update this base address to switch to a new frame buffer, which needs to be 16-byte aligned so that the four LSBs are always zero. In context switching during frame interval, the descriptor (NXT_DESCPT_PTR[4h]) loads this register.

Diagram



Fields

Field	Function
31-4 OUT_BUF_BASE0	Pixel Frame Buffer0 Base Specifies the output buffer0 base address. For images in the YUV444, RGB/ARGB, YUV422, and Gray formats, the decoded image pixel is in Interleave mode and stored on the frame buffer starting from this field. For the images in the YUV420 format, the first planner is the Y component and is stored in the frame buffer that this field defines.
3-0 —	Reserved

48.6.1.5 Output Image Frame Buffer1 Base Address (OUT_BUF_BASE1)

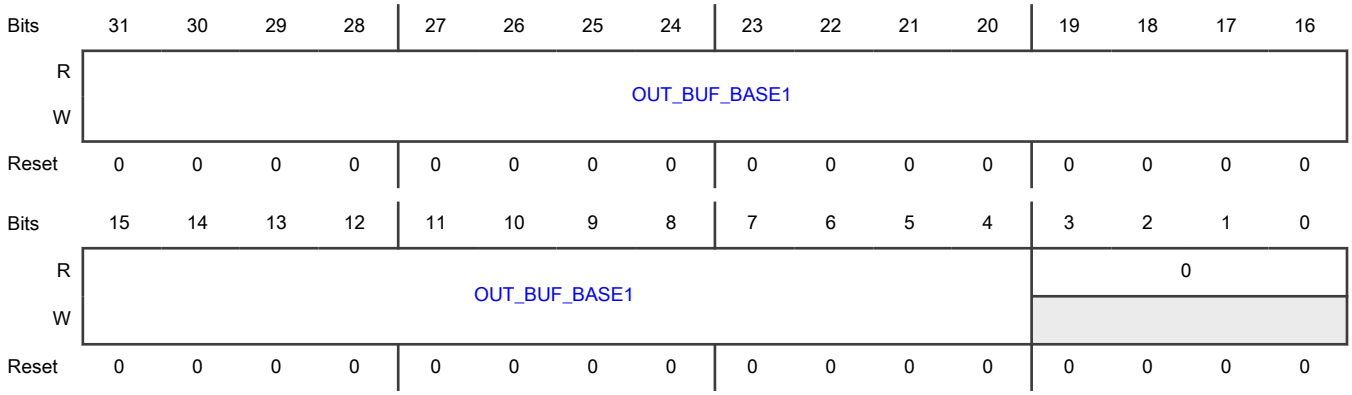
Offset

Register	Offset
OUT_BUF_BASE1	18h

Function

Specifies the base address for output frame buffer1. This register points to the beginning of the second planner output frame buffer. It is only enabled for images in the YUV420 format, and the second planner is UV interleaved. UV components are stored in the frame buffer that this register defines. The first planar for YUV420 is YY and is stored in frame buffer0. After one frame of image decoding is complete, and the image is output to the memory, you can update the base address when the wrapper is fetching the descriptor from memory. In context switching during frame interval, the descriptor (NXT_DESCPT_PTR[8h]) loads this register.

Diagram



Fields

Field	Function
31-4 OUT_BUF_BASE1	Pixel Frame Buffer1 Base Specifies the output buffer1 base address. It is only enabled for images in the YUV420 format, and is used to store the second planner (UV) component.
3-0 —	Reserved

48.6.1.6 Image Output Buffer Pitch (OUT_PITCH)

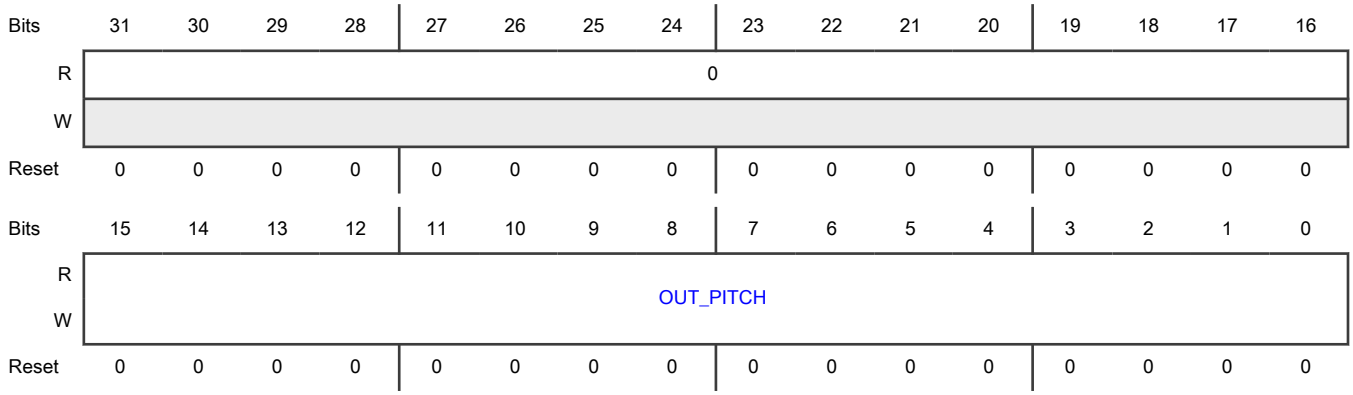
Offset

Register	Offset
OUT_PITCH	1Ch

Function

Specifies the vertical pixel distance, in bytes, of the image in memory (in the frame buffer). Any byte value indicates the vertical pitch, which is used for output pixel address calculation. For YUV420, the image is in a 2-planar format. The line pitch is applied to both planar. In context switching during the frame interval, the descriptor (NXT_DESCPT_PTR[Ch]) loads this register.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUT_PITCH	Output Image Pitch Specifies the vertical pixel distance, in bytes, of the image in memory.

48.6.1.7 Input JPEG Stream Buffer Base Address (STM_BUFBASE)

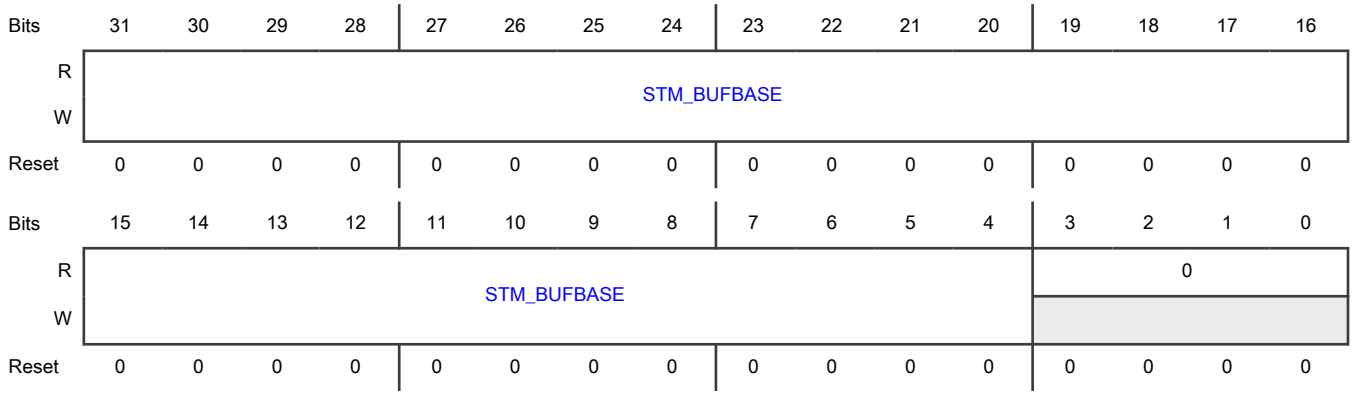
Offset

Register	Offset
STM_BUFBASE	20h

Function

Defines the base address of the stream buffer for decoding. The memory must have a 64K boundary so that the lower 4 bits of this register are always zero. Each slot has one bit stream buffer and one bit stream buffer pointer. The bit stream buffer pointer keeps tracking from where the slot is switched out and when this slot is switched in again. The decoding also continues from this bit stream buffer pointer for this slot. In context switching during frame interval, the register (NXT_DESCPT_PTR[10h]) loads this register.

Diagram



Fields

Field	Function
31-4 STM_BUFBASE	Bit Stream Buffer Base Specifies the base address of the stream buffer.
3-0 —	Reserved

48.6.1.8 Input JPEG Stream Buffer Size (STM_BUFSIZE)

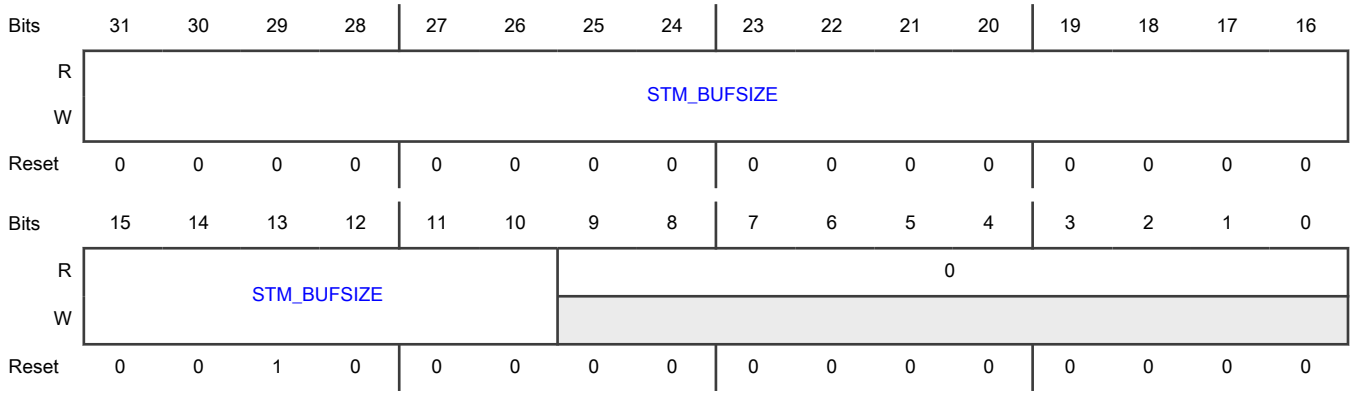
Offset

Register	Offset
STM_BUFSIZE	24h

Function

Defines the size of the stream buffer. JPGDECWRP can generate an interrupt when the stream buffer pointer reaches half the size of the buffer, or wraps around the buffer. So, the buffer works like double-bit-stream buffer mode. In context switching during frame interval, the descriptor (NXT_DESCPT_PTR[14h]) loads this register.

Diagram



Fields

Field	Function
31-10 STM_BUFSIZE	Bit Stream Buffer Size Specifies the stream buffer size. It has to be integer times of 1 KB.
9-0 —	Reserved

48.6.1.9 Image Resolution (IMGSIZE)

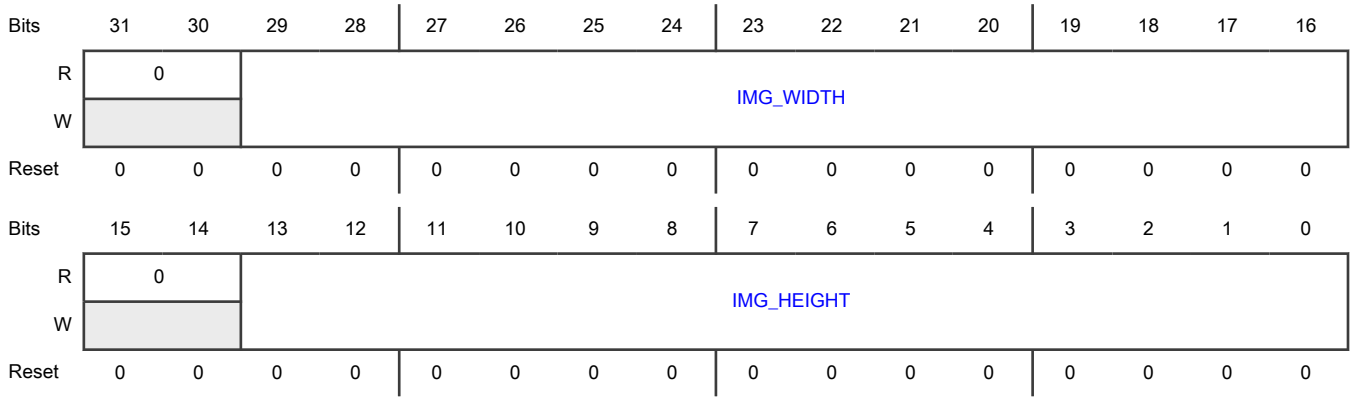
Offset

Register	Offset
IMGSIZE	28h

Function

Defines the resolution of the image to be decoded. In context switching during frame interval, the descriptor (NXT_DESCPT_PTR[18h]) loads this register.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-16 IMG_WIDTH	Image Width Specifies image width resolution in pixel unit. The maximum horizontal resolution supported is 8K (2000h). The horizontal resolution must be integer times of 8.
15-14 —	Reserved
13-0 IMG_HEIGHT	Image Height Specifies image height resolution in pixel unit. The maximum vertical resolution supported is 8K (2000h). The vertical resolution must be integer times of 8.

48.6.1.10 Bit Stream and Switching Control (STM_CTRL)

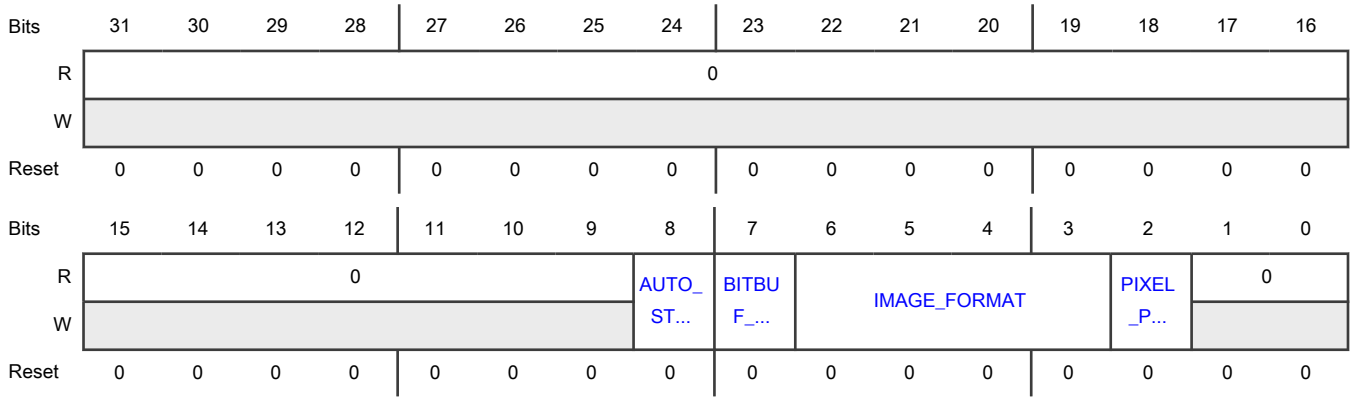
Offset

Register	Offset
STM_CTRL	2Ch

Function

Enables you to configure the current bit stream slot for decoding. In context switching during frame interval, the descriptor (NXT_DESCPT_PTR[1Ch]) loads this register.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 AUTO_START	Auto Start In some cases, JPEGDEC needs to write 1 to CTRL[GO] to start operation (configuration or decoding). Automatically writes 1 to CTRL[GO] in JPEGDEC when this field is 1 after the context is switched. 0b - Do not write 1 to CTRL[GO] in JPEGDEC. 1b - Writes 1 to CTRL[GO] in JPEGDEC.
7 BITBUF_PTR_CLR	Bit Buffer Pointer Clear Clears or restores the bit stream buffer pointer from the save pointer. If this field is 0, after context switching completion, the bit stream buffer pointer is restored to the last time when the bit stream was switched out. If this field is 1, the bit stream buffer pointer is cleared after context switching completion. 0b - Restores the bit stream buffer pointer from the save pointer. 1b - Clears the bit stream buffer pointer from the save pointer.
6-3 IMAGE_FORMAT	Image Format Specifies the color component number in the current bit stream. 0h–3h indicate the current decoding image format and its output format in memory. 0000b - YUV420 (2-planar, Y at the first planar, and UV at the second planar) 0001b - YUV422 (1-planar in the YUYV sequence) 0010b - RGB (BGRBGR packed format) 0011b - YUV444 (first planar in the YUVYUV sequence) 0100b - Gray (Y8 or Y12) or single component 0101b - Reserved 0110b - ARGB
2 PIXEL_PRECISION	Pixel Precision Specifies current decoding precision of the pixel to be decoded. 0b - 8-bit 1b - 12-bit
1-0 —	Reserved

48.6.1.11 Bit Stream Status (SLOT0_STATUS - SLOT3_STATUS)

Offset

Register	Offset
SLOT0_STATUS	200h
SLOT1_STATUS	400h
SLOT2_STATUS	600h
SLOT3_STATUS	800h

Function

Indicates bit stream slot status. Each bit stream ID has one set of status registers that only takes effect for its own bit stream or decoding operation. Each bit stream ID has its interrupt lines associated with the status register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEC_ONG...	CUR_SLOT			0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PIXEL_W...	BIT_RD_...	DES_RD_...	DECERR	0				FRMDONE	SWITCHE...	STMBUF_...	STMBUF_...
W					W1C	W1C	W1C	W1C					W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 DEC_ONGOING	Decoding Ongoing Indicates whether the decoding is ongoing. The field becomes 0 if there are no more descriptors to be fetched for each slot, or when the slot is switched in frame interval. 0b - Paused or stopped 1b - Ongoing
30-29 CUR_SLOT	Current Slot Indicates the current executing bit stream slot.
28-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 PIXEL_WT_ERR	Pixel Write Error Flag Indicates whether a pixel write error happened when writing the decoded image pixel to memory. You must clear any slot errors for JPGDECWRP to continue working. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No pixel write error occurred. 1b - A pixel write error occurred. When writing 0b - No effect 1b - Clears the flag.
10 BIT_RD_ERR	Bit Read Error Flag Indicates whether a bit read error happened during bit stream fetching. You must clear any slot errors for JPGDECWRP to continue working. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No bit read error occurred. 1b - A bit read error occurred. When writing 0b - No effect 1b - Clear the flag
9 DES_RD_ERR	Descriptor Read Error Flag Indicates whether a descriptor read error happened during bit stream fetching. You must clear any slot errors for the JPGDECWRP to continue working. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No descriptor read error occurred. 1b - A descriptor read error occurred. When writing 0b - No effect 1b - Clears the flag.

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 DECERR	Decoding Error Flag Indicates whether there is an error in the JPEG bit stream. You must clear any slot errors for JPGDECWRP to continue working. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No decoding error occurred. 1b - A decoding error occurred. When writing 0b - No effect 1b - Clears the flag.
7-4 —	Reserved
3 FRMDONE	Frame Done Flag Indicates whether one full frame of image is decoded and the image pixels are stored in the memory frame buffer. The wrapper switches to the next slot. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Decoding is not completed. 1b - Decoding is completed. When writing 0b - No effect 1b - Clears the flag.
2 SWITCHED_IN	Switched In Flag Indicates whether the current slot is switched in during context switching. JPGDECWRP completes fetching the descriptor and switches in to the current slot. New decoding operation immediately starts. **NOTE** This field behaves differently for register reads and writes. When reading 0b - The current slot is not switched in during context switching. 1b - The current slot is switched in during context switching. When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clears the flag.
1 STMBUF_RTND	Stream Buffer Returned Flag Indicates whether the bit stream buffer for the current slot has passed the top mark of the buffer and is wrapped around to the bottom of the buffer. You can now safely refresh the higher half of the bit stream buffer. **NOTE** This field behaves differently for register reads and writes. When reading 0b - The bit stream buffer pointer for the current slot has not passed the top mark of the buffer. 1b - The bit stream buffer pointer for the current slot passed the top mark of the buffer. When writing 0b - No effect 1b - Clears the flag.
0 STMBUF_HALF	Stream Buffer Half Flag Indicates whether the bit stream buffer pointer for the current slot passed half of the buffer size. You can safely refresh the lower half of the bit stream buffer after that for decoding. **NOTE** This field behaves differently for register reads and writes. When reading 0b - The bit stream buffer pointer for the current slot has not passed half of the buffer size. 1b - The bit stream buffer pointer for the current slot passed half of the buffer size. When writing 0b - No effect 1b - Clears the flag.

48.6.1.12 Bit Stream Interrupt Enable (SLOT0_IRQ_EN - SLOT3_IRQ_EN)

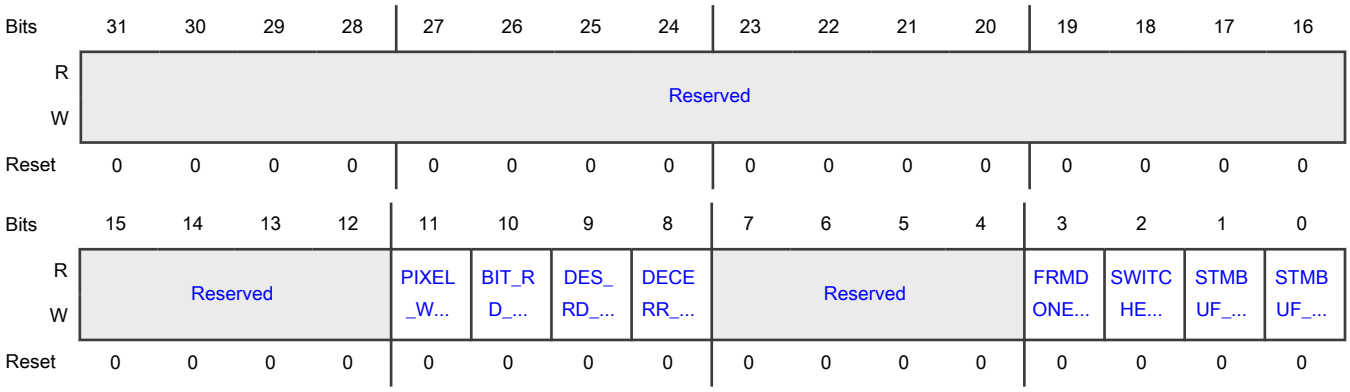
Offset

Register	Offset
SLOT0_IRQ_EN	204h
SLOT1_IRQ_EN	404h
SLOT2_IRQ_EN	604h
SLOT3_IRQ_EN	804h

Function

Enables IRQ for the bit stream context operation. Each bit stream ID has its own interrupt enable register and interrupt request.

Diagram



Fields

Field	Function
31-12 —	Reserved
11 PIXEL_WT_ERR_IRQ_EN	Pixel Write Error Interrupt Enable Enables an AXI write error interrupt for pixel storing of the current slot when SLOT0_STATUS[PIXEL_WT_ERR] = 1. 0b - Disable 1b - Enable
10 BIT_RD_ERR_IRQ_EN	Bit Read Error Interrupt Enable Enables an AXI read error interrupt for bit stream fetching of the current slot when SLOT0_STATUS[BIT_RD_ERR] = 1. 0b - Disable 1b - Enable
9 DES_RD_ERR_IRQ_EN	Descriptor Read Error Interrupt Enable Enables an AXI read error interrupt for descriptor fetching of the current slot when SLOT0_STATUS[DES_RD_ERR] = 1. 0b - Disable 1b - Enable
8 DECERR_IRQ_EN	Decoding Error Status Interrupt Enable Enables an interrupt for the current slot when a decoder error happens. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
7-4 —	Reserved
3 FRMDONE_IRQ_EN	Frame Done Interrupt Enable Enables an interrupt for the current bit stream after decoding is complete. 0b - Disable 1b - Enable
2 SWITCHED_IN_IRQ_EN	Switched In Interrupt Enable Enables an interrupt for the current bit stream when the current slot is switched into during context switching. 0b - Disable 1b - Enable
1 STMBUF_RTND_IRQ_EN	Stream Buffer Returned Interrupt Enable Enables an interrupt for the current slot when its bit stream buffer pointer has wrapped around from the buffer top to bottom. 0b - Disable 1b - Enable
0 STMBUF_HALF_IRQ_EN	Stream Buffer Half Interrupt Enable Enables an interrupt for the current slot when its bit stream buffer pointer has passed half the size of the buffer. 0b - Disable 1b - Enable

48.6.1.13 Bit Stream Buffer Pointer (SLOT0_BUF_PTR - SLOT3_BUF_PTR)

Offset

Register	Offset
SLOT0_BUF_PTR	208h
SLOT1_BUF_PTR	408h
SLOT2_BUF_PTR	608h
SLOT3_BUF_PTR	808h

Function

Points to the location of stream buffer for the current bit stream ID. The next frame decoding must begin with this bit stream ID. This register keeps the bit stream buffer pointer on track in context switching for each slot. When the context is switched back to one of the bit streams (identified by the bit stream ID), the decoding can continue to move forward from where it was switched. The pointer starts from the bit stream buffer base, and goes up to the upper buffer size limit, and then wraps back to the bit stream base again. You can configure JPGDECWRP to generate an interrupt when the pointer hits half the size of the buffer or top of the buffer, and wraps around to the base again. This can make the buffer work in a double buffer mode while each buffer is half of the buffer size limit. You can clear the bit stream buffer during context switching when switched to one slot, with STM_CTRL[BITBUF_PTR_CLR] in the descriptor.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STMBUF_PTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STMBUF_PTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 STMBUF_PTR	Stream Buffer Pointer Points to the location of the bit stream buffer that you can clear during context switching by writing 1 to STM_CTRL[BITBUF_PTR_CLR] in the descriptor. Each bit stream ID has one stream buffer pointer.

48.6.1.14 Current Descriptors (SLOT0_CUR_DESCPT_PTR - SLOT3_CUR_DESCPT_PTR)

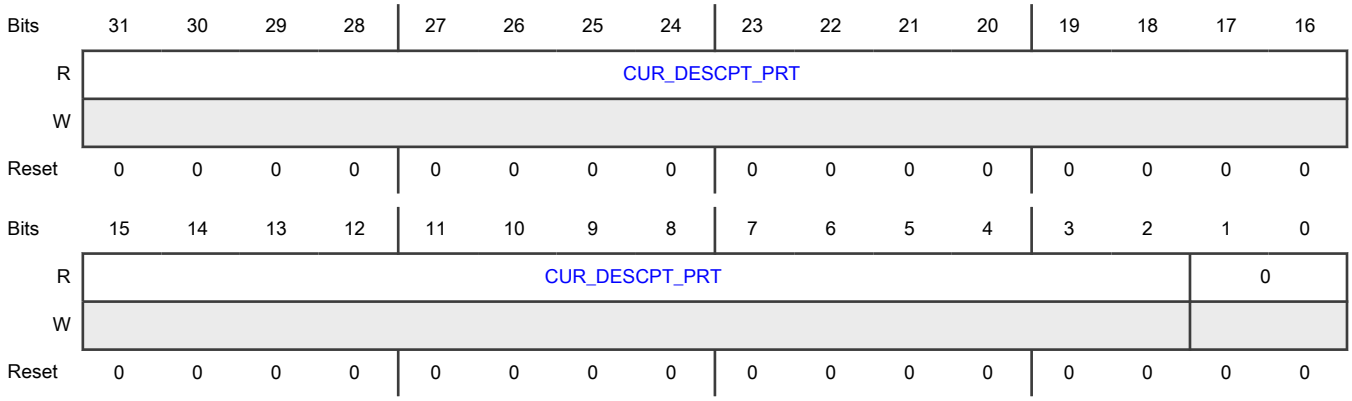
Offset

Register	Offset
SLOT0_CUR_DESCPT_PTR	20Ch
SLOT1_CUR_DESCPT_PTR	40Ch
SLOT2_CUR_DESCPT_PTR	60Ch
SLOT3_CUR_DESCPT_PTR	80Ch

Function

Indicates the address of the descriptor that has already been executed or is currently executing for the current slot.

Diagram



Fields

Field	Function
31-2 CUR_DESCPT_PRT	Current Decoding Descriptor Pointer Indicates a pointer to the current descriptor that is currently executing or has already been executed.
1-0 —	Reserved

48.6.1.15 Next Descriptors (SLOT0_NXT_DESCPT_PTR - SLOT3_NXT_DESCPT_PTR)

Offset

Register	Offset
SLOT0_NXT_DESCPT_PTR	210h
SLOT1_NXT_DESCPT_PTR	410h
SLOT2_NXT_DESCPT_PTR	610h
SLOT3_NXT_DESCPT_PTR	810h

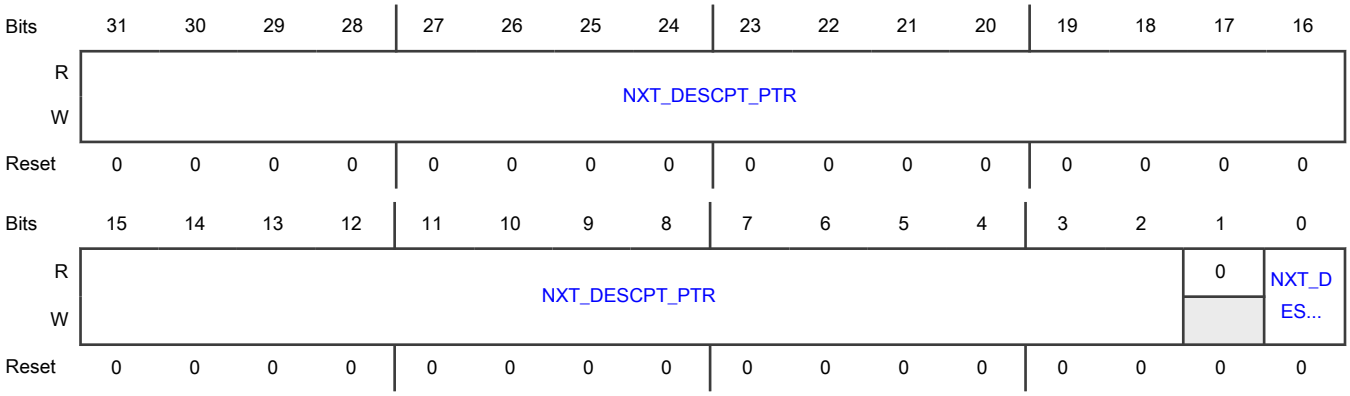
Function

Points to the base of the next decoding descriptors in the current slot. NXT_DESCPT_EN indicates whether the next descriptor is valid. If it is not, JPGDECWRP does not execute it. The current slot is excluded from rotation in decoding. When the controller of the slot gets the descriptor pointed by this register ready, it writes 1 to NXT_DESCPT_EN. If GLB_CTRL[SLOT_EN] = 1, this slot is scheduled into the execution rotation in a round-robin sequence. At the end of a frame decoding, the wrapper fetches the descriptor that this register points to. Also, the descriptor (NXT_DESCPT_PTR[0h]) loads this register. The descriptor contains eight 32-bit words:

- NXT_DESCPT_PTR[0h]: NXT_DESCPT_PTR

- NXT_DESCPT_PTR[4h]: OUT_BUF_BASE0
- NXT_DESCPT_PTR[8h]: OUT_BUF_BASE1
- NXT_DESCPT_PTR[Ch]: OUT_LINE_PITCH
- NXT_DESCPT_PTR[10h]: STM_BUFBASE
- NXT_DESCPT_PTR[14h]: STM_BUFSIZE
- NXT_DESCPT_PTR[18h]: IMGSIZE
- NXT_DESCPT_PTR[1Ch]: STM_CTRL

Diagram



Fields

Field	Function
31-2 NXT_DESCPT_PTR	Next Descriptor Pointer Contains a pointer to the next descriptor in memory. When the next descriptor is enabled for the current slot, this slot is included in context switching in a round-robin sequence. Also, when the current slot is executing, JPGDECWRP fetches the descriptor that this register points to.
1 —	Reserved
0 NXT_DESCPT_EN	Next Stream Descriptor Pointer Enable Enables the next stream descriptor pointer. If the next descriptor pointer is invalid, JPGDECWRP does not include the current slot in context switching rotation, even the current slot is set in GLB_CTRL[SLOT_EN]. After a slot if enabled in Global Control (GLB_CTRL), the slot owner can write to this register to enable the descriptor pointer, and thus the current slot is included in the context-switching rotation. If you want to pause the current slot and exclude it from the decoding rotation, then write 0 to this field. 0b - Disable 1b - Enable

48.6.2 JPEGDEC register descriptions

This section includes the JPEGDEC module memory map and detailed descriptions of all registers.

The submodule JPEG Decoder Wrapper has four slots. Only the active slot can access JPEGDEC's control register. This means, for example, you can access the jpeg_decode0 registers only when slot0 is active, and so on.

48.6.2.1 JPEGDEC memory map

JPEGDEC base address: 400A_6100h

Offset	Register	Width (In bits)	Access	Reset value
34h	Control (CTRL)	32	RW	0000_0001h

48.6.2.2 Control (CTRL)

Offset

Register	Offset
CTRL	34h

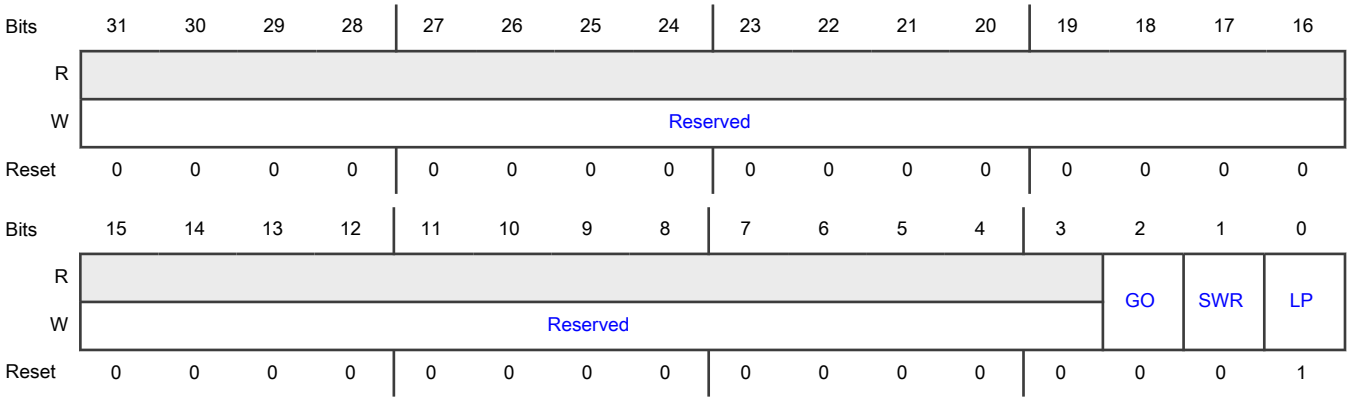
Function

Enables control for the core.

Although the Control register can be written at all times, it affects the core operation after the completion of the current decoding process. An exception to this rule is the SWR and LP bits, which affect the core operation four cycles after they have been altered. The Control Register is also mapped at address 0x0D of the status interface.

Bit 0 (LP) provides the means to force the core to enter into a low-power state, where all internal registers apart from those participating in the control register programming are frozen. Software resetting the core is feasible by setting bit 1 (SWR) to 1.

Diagram



Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
2 GO	Go Enables the core to exit the idle mode. 0b - No effect 1b - Enable
1 SWR	Soft Reset Enables soft reset for the core. 0b - No effect 1b - Enable
0 LP	Low Power Enables the core to enter the low-power mode. 0b - No effect 1b - Enable

Chapter 49
PNG Decoder (PNGDEC)

49.1 Chip-specific PNGDEC information

Table 437. Reference links to related information

Topic	Related module	Reference
Full description	PNGDEC	PNGDEC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

49.1.1 Module instances

This device contains one instance of the PNGDEC module.

Table 438. PNGDEC instance

Features	PNGDEC
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

49.2 Overview

PNGDEC decodes pictures in the PNG source format. PNGDEC includes a DMA engine.

49.2.1 Block diagram

Figure 170 shows the submodules of PNGDEC.

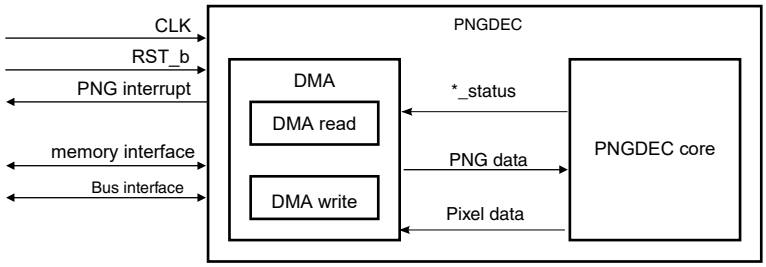


Figure 170. Block diagram

49.2.2 Features

PNGDEC has the following features:

- 64-bit AXI DMA engines used to fetch PNG bitstreams from the system memory and feed them into PNGDEC, and store decoded RGB-A raw pixel data and ancillary data

- If the image is grayscale, R, G and B will have the same value. If the image uses 16 bits per channel, R, G, and B are down converted to 8 bits by eliminating the 8 least significant bits. Finally, if the image has no transparency layer, then A is 255.
- The ancillary data consists of the PNG header followed by the ancillary chunks (except CRC) in the standard PNG format.
- Byte-order configuration
- Huffman encoded data (both static and dynamic)
- Gray scale, RGB (8- or 16-bit depth) and indexed color (8-bit depth) with and without alpha channel
- Full control through PNG header while decoding
- LZ77 sliding window with a maximum length of 32 KB
- Image with a maximum width of 1024 pixels
- Ancillary data drop

NOTE

PNGDEC does not support interlacing, uncompressed data chunks, or bit depths lower than 8 bits.

49.3 Functional description

The following sections describe the functionalities of PNGDEC.

49.3.1 Operating modes

PNGDEC operates in two modes: DMA and Drop.

Table 439. Operating modes

Mode	Description
DMA	Reads the PNG file, sends it to the decoder for decoding, and writes RGB-A raw pixel and ancillary data.
Drop	Enables ancillary data drop. DMA only writes RGB-A raw pixel data.

49.3.1.1 DMA mode

When the enable bit fields in the Global Control (GLB_CTRL) register are set to 1, DMA read copies the PNG file from the system memory and performs decoding. During decoding, DMA write starts copying the decoded RGB-A raw pixel and ancillary data to the system memory until the DMA interrupt showing a status of done occurs, and the PNG file decoding is completed.

49.3.1.2 Drop mode

In drop mode, PNGDEC operates in the similar way as DMA mode. However, you must write 1 to GLB_CTRL[ANC_DROP_EN] to enable the ancillary data drop function. You do not need to check DEC_INT_STS[DEC_ANC_DMA_DONE].

49.3.2 Operations

49.3.2.1 DMA mode operation

To operate in DMA mode, perform the following procedures:

1. Write 1 and then write 0 to CNT_CTRL_CLR to clear the dirty counter.
2. Write 1 to every bit of DEC_INT_STS to clear the dirty status.
3. Set ENC_DATA_DMA_SRC_LEN[SRC_LEN] to configure the PNG file DMA source length.
4. Set ENC_DATA_DMA_SRC_ADDR[SRC_ADDR] to configure the PNG file DMA source address.

5. Set DEC_PIXEL_DMA_DES_ADDR[PIXEL_DES_ADDR] to configure the PNG decode pixel destination address.
6. Set DEC_ANC_DMA_DES_ADDR[ANC_DES_ADDR] to configure the PNG decode ancillary destination address.
7. Write 1 to GLB_CTRL[DEC_EN] to enable the decode function.
8. Write to DEC_INT_STS_EN to enable the interrupt status.
9. Write 1 and then write 0 to DMA_TRIG[DMA_TRIG] to trigger the DMA copy engine. After some time, the interrupt appears.
10. Read DEC_INT_STS, making sure that the "*_done" status appears and "err_*" does not appear.
11. Read DEC_PIXEL_DMA_LEN and Decode Ancillary Destination Length (DEC_ANC_DMA_DES_LEN) to get the information of read total decoded pixel length and total decoded ancillary length, respectively. This data is your decoded result.

After one PNG decoding is done, wait for the next picture and repeat the aforementioned steps.

49.3.2.2 Drop mode operation

In drop mode, perform the following procedures (most of the steps are the same as in DMA mode operation):

1. Write 1 and then write 0 to CNT_CTRL_CLR to clear the dirty counter.
2. Write 1 to every bit of DEC_INT_STS to clear the dirty status.
3. Set ENC_DATA_DMA_SRC_LEN[SRC_LEN] to configure the PNG file DMA source length.
4. Set ENC_DATA_DMA_SRC_ADDR[SRC_ADDR] to configure the PNG file DMA source address.
5. Set DEC_PIXEL_DMA_DES_ADDR[PIXEL_DES_ADDR] to configure the PNG decode pixel destination address.
6. Write 1 to GLB_CTRL[DEC_EN] to enable the decode function.
7. Write 1 to GLB_CTRL[ANC_DROP_EN] to enable the ancillary drop function.
8. Write to DEC_INT_STS_EN to enable the interrupt status.
9. Write 1 and then write 0 to DMA_TRIG[DMA_TRIG] to trigger the DMA copy engine. After some time, the interrupt appears.
10. Read DEC_INT_STS, making sure that:
    - Both the "enc_data_dma_done" and "dec_pixel_dma_done" statuses appear.
    - The "err_*" status does not appear.
11. Read DEC_PIXEL_DMA_DES_LEN to get the information of read total decoded pixel length.

After one PNG decoding is done, wait for the next picture and repeat the aforementioned steps.

49.3.3 Clocking

PNGDEC uses the following clock.

Table 440. Clock

Clock	Description
CLK	All submodules of PNGDEC share the same clock.

49.3.4 Reset

PNGDEC has hardware and software resets. You can do software reset for PNGDEC by setting GLB_CTRL[DMA_SW_LOGIC_RST_EN] and GLB_CTRL[IPCORE_SW_LOGIC_RST_EN] independently, for example, write 1 and then write 0 to reset logic.

49.3.5 Interrupts

You can check the interrupt status of PNGDEC in the DEC_INT_STS register, and enable interrupts by setting bitfields in DEC_INT_STS_EN.

49.4 External signals

This module has no external signals.

49.5 Initialization

PNGDEC does not require initialization.

49.6 PNGDEC register descriptions

This section includes the PNGDEC module memory map and detailed descriptions of all registers for accessing DMA registers and PNG decode core status through PNGDEC's bus.

49.6.1 PNGDEC memory map

PNGDEC base address: 400A_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Global Control (GLB_CTRL)	32	RW	0000_0000h
4h	DMA Trigger (DMA_TRIG)	32	RW	0000_0000h
8h	Count Control Clear (CNT_CTRL_CLR)	32	RW	0000_0000h
20h	Encode Data Source Address (ENC_DATA_DMA_SRC_ADDR)	32	RW	0000_0000h
24h	Encode Data Source Length (ENC_DATA_DMA_SRC_LEN)	32	RW	0000_0000h
28h	Decode Pixel Destination Address (DEC_PIXEL_DMA_DES_ADDR)	32	RW	0000_0000h
2Ch	Decode Pixel Destination Length (DEC_PIXEL_DMA_DES_LEN)	32	R	0000_0000h
30h	Decode Ancillary Destination Address (DEC_ANC_DMA_DES_ADDR)	32	RW	0000_0000h
34h	Decode Ancillary Destination Length (DEC_ANC_DMA_DES_LEN)	32	R	0000_0000h
80h	Decode Interrupt Status (DEC_INT_STS)	32	RW	0000_0000h
84h	Interrupt Enable (DEC_INT_STS_EN)	32	RW	0000_0000h
A0h	AXI3 Encoded Data Read Bytes Counter (AXI3_ENC_DATA_RD_BYTE_CNT)	32	R	0000_0000h
A4h	AXI3 Decoded Pixel Write Bytes Counter (AXI3_DEC_PIXEL_WR_BYTE_CNT)	32	R	0000_0000h
A8h	AXI3 Decode Ancillary Write Bytes Counter (AXI3_DEC_ANC_WR_BYTE_CNT)	32	R	0000_0000h

49.6.2 Global Control (GLB_CTRL)

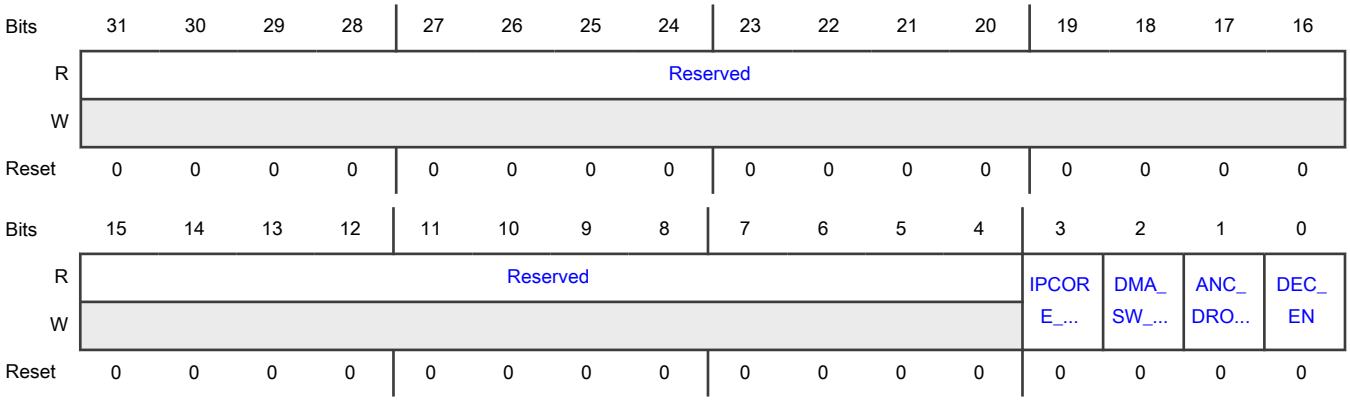
Offset

Register	Offset
GLB_CTRL	0h

Function

Controls PNGDEC operation.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 IPCORE_SW_LOGIC_RST_EN	Core Software Logic Reset Enable Enables software reset for all PNGDEC core logic except for registers. You can write 1 and then write 0 to reset logic. 0b - Disable 1b - Enable
2 DMA_SW_LOGIC_RST_EN	DMA Software Logic Reset Enable Enables software reset for all DMA logic except for registers. You can write 1 and then write 0 to reset logic. 0b - Disable 1b - Enable
1	Ancillary Data Drop Enable Enables the ancillary data drop function.

Table continues on the next page...

Table continued from the previous page...

Field	Function
ANC_DROP_EN	0b - Disable 1b - Enable
0 DEC_EN	Decode Enable Enables PNG decoding. 0b - Disable 1b - Enable

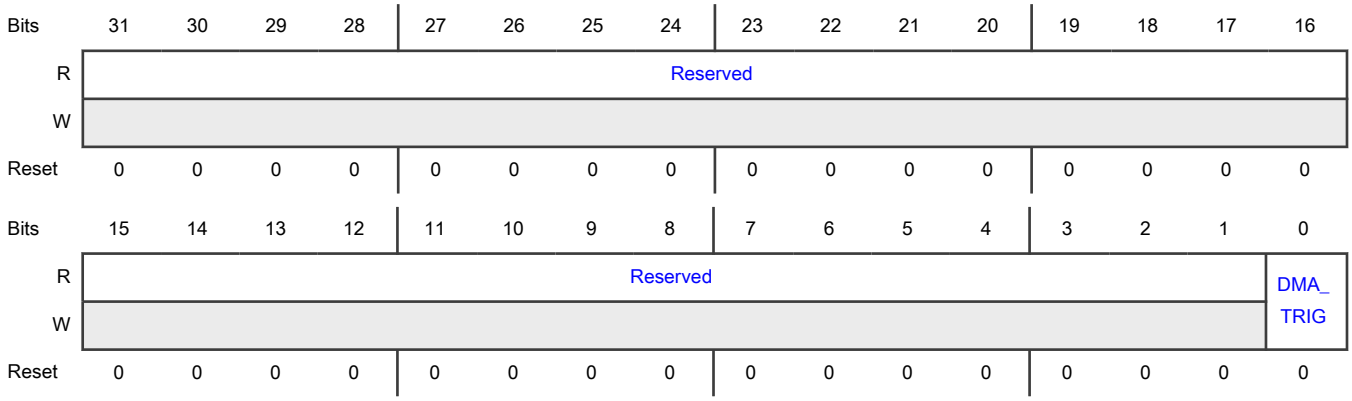
49.6.3 DMA Trigger (DMA_TRIG)

Offset

Register	Offset
DMA_TRIG	4h

Function
Triggers DMA.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 DMA_TRIG	DMA Trigger Triggers PNGDEC to start the DMA-copy and PNG-decoding operations.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Write 1 and then write 0 to trigger DMA. 0b - Disable 1b - Enable

49.6.4 Count Control Clear (CNT_CTRL_CLR)

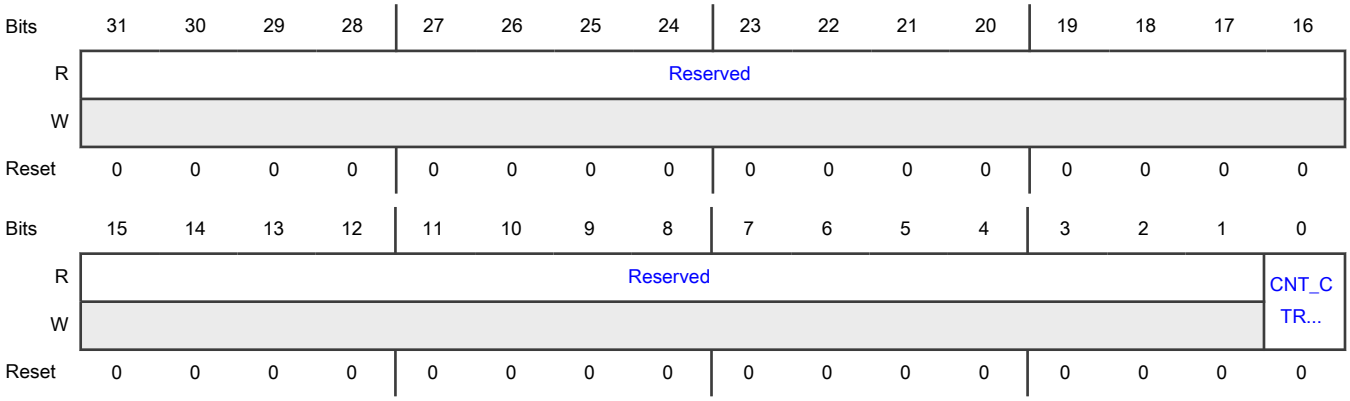
Offset

Register	Offset
CNT_CTRL_CLR	8h

Function

Clears AXI3 Encoded Data Read Bytes Counter, AXI3 Decoded Pixel Write Bytes Counter, and AXI3 Decode Ancillary Write Bytes Counter.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CNT_CTRL_CLR	Count Control Clear Clears AXI3 Encoded Data Read Bytes Counter, AXI3 Decoded Pixel Write Bytes Counter, and AXI3 Decode Ancillary Write Bytes Counter.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable

49.6.5 Encode Data Source Address (ENC_DATA_DMA_SRC_ADDR)

Offset

Register	Offset
ENC_DATA_DMA_SRC_ADDR	20h

Function

Specifies the PNG coded data source address.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SRC_ADDR	Source Address Specifies the PNG coded data source address, which must be 8-byte aligned.

49.6.6 Encode Data Source Length (ENC_DATA_DMA_SRC_LEN)

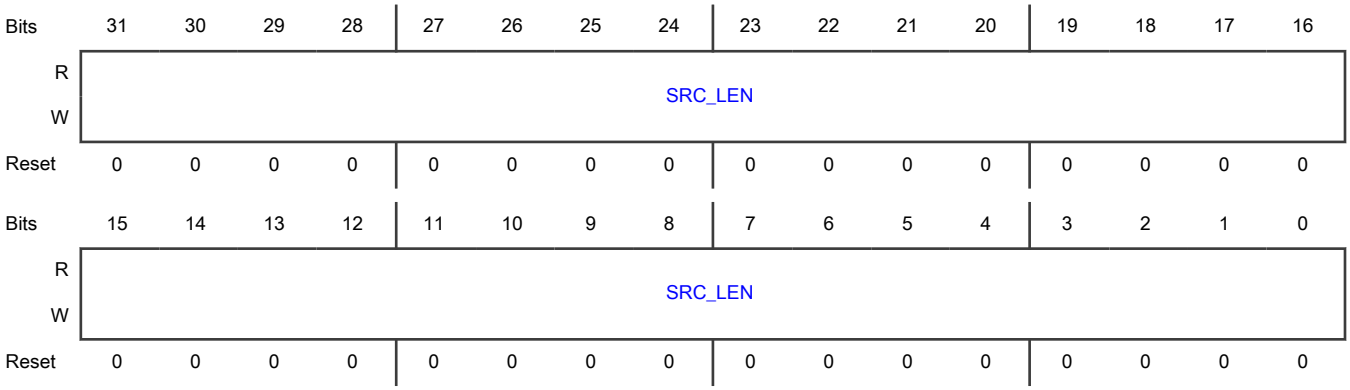
Offset

Register	Offset
ENC_DATA_DMA_SRC_LEN	24h

Function

Specifies the PNG coded data source length.

Diagram



Fields

Field	Function
31-0 SRC_LEN	Source Length Specifies the PNG coded data source length (in bytes).

49.6.7 Decode Pixel Destination Address (DEC_PIXEL_DMA_DES_ADDR)

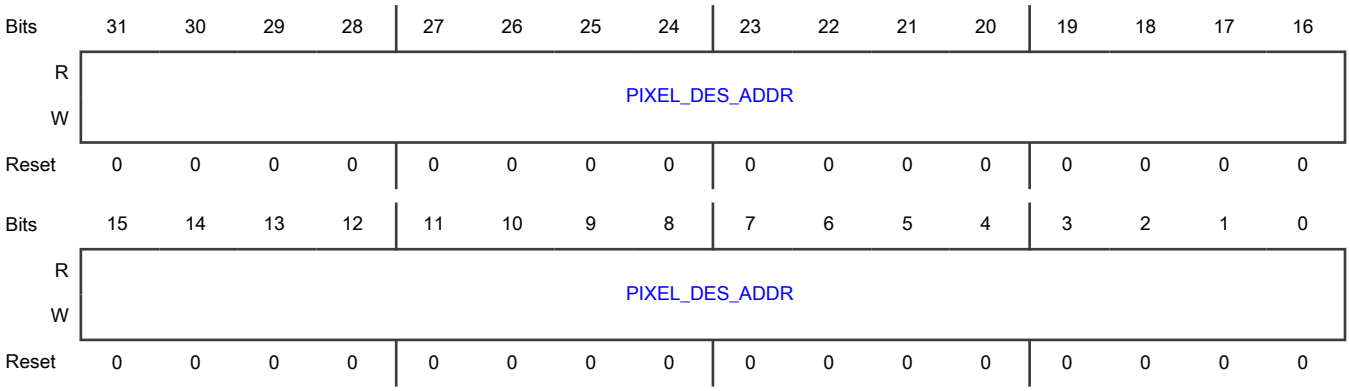
Offset

Register	Offset
DEC_PIXEL_DMA_DES_ADDR	28h

Function

Specifies the PNG decode pixel destination address.

Diagram



Fields

Field	Function
31-0 PIXEL_DES_ADDR	Pixel Destination Address Specifies the PNG decode pixel destination address, which must be 8-byte aligned.

49.6.8 Decode Pixel Destination Length (DEC_PIXEL_DMA_DES_LEN)

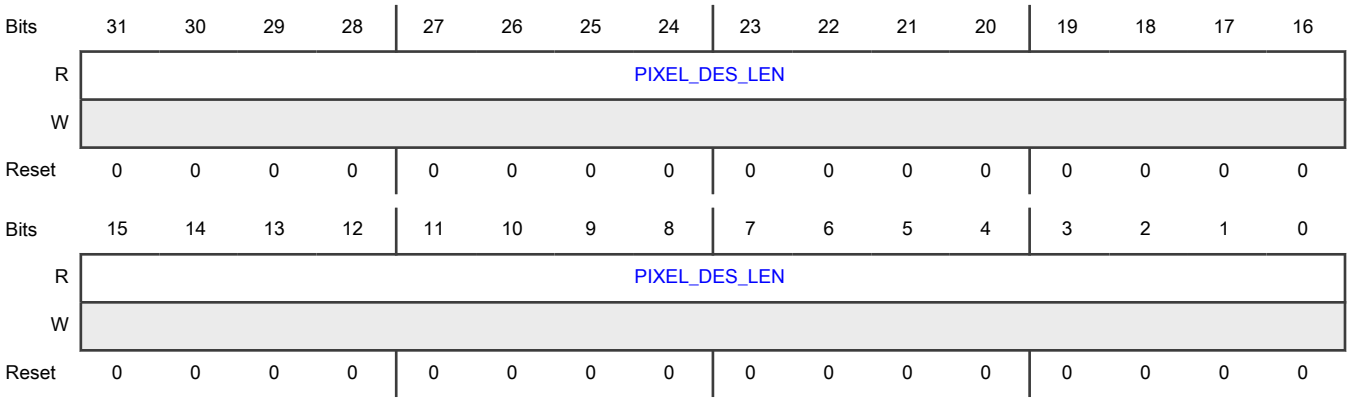
Offset

Register	Offset
DEC_PIXEL_DMA_DES_LEN	2Ch

Function

Specifies the PNG decode pixel destination length.

Diagram



Fields

Field	Function
31-0 PIXEL_DES_LEN	Pixel Destination Length Specifies the PNG decode pixel destination length (in bytes).

49.6.9 Decode Ancillary Destination Address (DEC_ANC_DMA_DES_ADDR)

Offset

Register	Offset
DEC_ANC_DMA_DES_ADDR	30h

Function

Specifies the PNG decode ancillary data destination address.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANC_DES_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANC_DES_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ANC_DES_ADDR	Ancillary Destination Address Specifies the PNG decode ancillary data destination address, which must be 8-byte aligned.

49.6.10 Decode Ancillary Destination Length (DEC_ANC_DMA_DES_LEN)

Offset

Register	Offset
DEC_ANC_DMA_DES_LEN	34h

Function

Specifies the PNG decode ancillary destination length.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ANC_DES_LEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ANC_DES_LEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ANC_DES_LEN	Ancillary Destination Length Specifies the PNG decode ancillary destination length (in bytes).

49.6.11 Decode Interrupt Status (DEC_INT_STS)

Offset

Register	Offset
DEC_INT_STS	80h

Function

Indicate the status of PNG decode interrupts.

NOTE

You can enable interrupts by setting bit fields in DEC_INT_STS_ENregister.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					DEC_ANC...	DEC_PIX...	ENC_DAT...	ERR_WID...	ERR_INT...	ERR_BIT...	ERR_ZLI...	ERR_BTY...	ERR_PNG...	ERR_CRC	ERR_ADL...
W						W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-11 —	Reserved
10 DEC_ANC_DMA_DONE	Ancillary Decode and DMA Copy Done Flag Indicates whether the ancillary-decode and DMA-copy-done interrupt occurred. If this field is 0, ancillary decoding and DMA copying are not done. If this field is 1, ancillary decoding and DMA copying are done. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
9 DEC_PIXEL_DMA_DONE	Decode Pixel and DMA Copy Done Indicates whether the decode-pixel and DMA-copy-done interrupt occurred. If this field is 0, pixel decoding and DMA copying are not done. If this field is 1, pixel decoding and DMA copying are done. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
8 ENC_DATA_DMA_DONE	Encoded Data and DMA Copy Done Indicates whether the encoded-data and DMA-copy-done interrupt occurred. If this field is 0, DMA is not initiated or DMA loading is in progress. If this field is 1, DMA completes loading encoded PNG data to the decoder module. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
7 ERR_WIDTH	Width Error Indicates whether an interrupt occurred due to a width error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
6 ERR_INTERLACE	Interlace Error Indicates whether an interrupt occurred due to an interlace error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
5 ERR_BITDEPTH	Bit Depth Error Indicates whether an interrupt occurred due to a bit-depth error. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
4 ERR_ZLIB_HEADER	Zlib Header Error Indicates whether an interrupt occurred due to a Zlib header error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
3 ERR_BTYPE	B-type Error (Block Encoding Type Error in IDAT Chunk) Indicates whether an interrupt occurred due to a B-type error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
2 ERR_PNG_HEADER	PNG Header Error Indicates whether an interrupt occurred due to a PNG header error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
1 ERR_CRC	CRC Error Indicates whether an interrupt occurred due to a CRC error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.
0 ERR_ADLER	ADLER-32 Checksum Error Indicates whether an interrupt occurred due to an ADLER-32 checksum error. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No interrupt occurred. 1b - Interrupt occurred. When writing 0b - No effect 1b - Clears status flag.

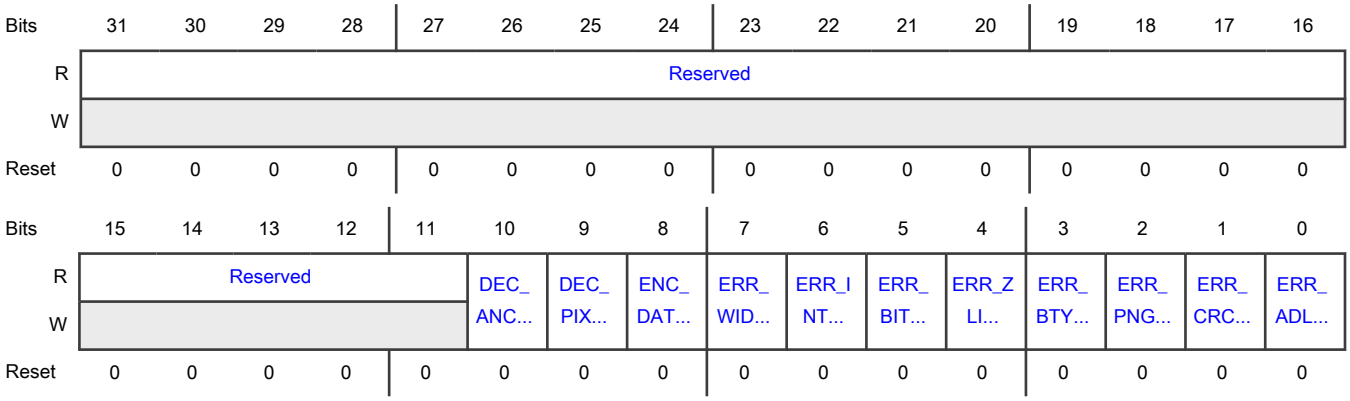
49.6.12 Interrupt Enable (DEC_INT_STS_EN)

Offset

Register	Offset
DEC_INT_STS_EN	84h

Function
Enables various PNG interrupts.

Diagram



Fields

Field	Function
31-11 —	Reserved
10 DEC_ANC_DMA_DONE_EN	Ancillary Decode and DMA Copy Done Interrupt Enable Enables the interrupt that occurs when an ancillary-decode operation and DMA copy finish. 0b - Disable 1b - Enable
9 DEC_PIXEL_DMA_DONE_EN	Decode Pixel and DMA Copy Done Interrupt Enable Enables the interrupt that occurs when a decode-pixel operation and DMA copy finish. 0b - Disable 1b - Enable
8 ENC_DATA_DMA_DONE_EN	Encoded Data and DMA Copy Done Interrupt Enable Enables the interrupt that occurs when DMA completes loading encoded PNG data to the decoder module. 0b - Disable 1b - Enable
7 ERR_WIDTH_EN	Width Error Interrupt Enable Enables the interrupt that occurs when a width error occurs. 0b - Disable 1b - Enable
6 ERR_INTERLACE_EN	Interlace Error Interrupt Enable Enables the interrupt that occurs when an interlace error occurs.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
5 ERR_BITDEPTH_EN	Bit Depth Error Interrupt Enable Enables the interrupt that occurs when a bit-depth error occurs. 0b - Disable 1b - Enable
4 ERR_ZLIB_HEADER_EN	Zlib Header Error Interrupt Enable Enables the interrupt that occurs when a Zlib header error occurs. 0b - Disable 1b - Enable
3 ERR_BTYPE_EN	B-Type Error Interrupt Enable Enables the interrupt that occurs when a B-type error occurs. 0b - Disable 1b - Enable
2 ERR_PNG_HEADER_EN	PNG Header Error Interrupt Enable Enables the interrupt that occurs when a PNG header error occurs. 0b - Disable 1b - Enable
1 ERR_CRC_EN	CRC Error Interrupt Enable Enables the interrupt that occurs when a CRC error occurs. 0b - Disable 1b - Enable
0 ERR_ADLER_EN	ADLER-32 Checksum Error Interrupt Enable Enables the interrupt that occurs when an ADLER-32 checksum error occurs. 0b - Disable 1b - Enable

49.6.13 AXI3 Encoded Data Read Bytes Counter (AXI3_ENC_DATA_RD_BYTE_CNT)

Offset

Register	Offset
AXI3_ENC_DATA_RD_BYTE_CNT	A0h

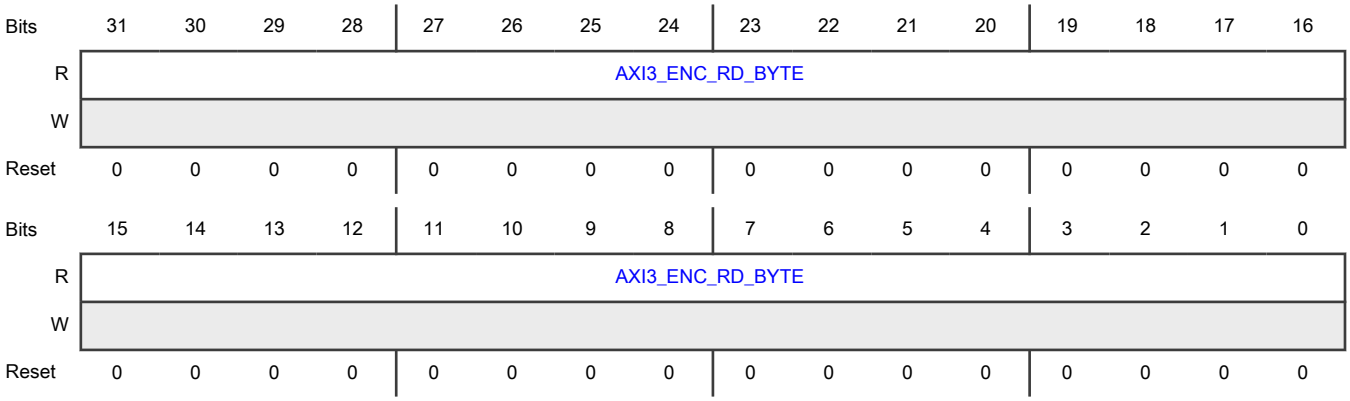
Function

Indicates the total read bytes of AXI3 encoded data.

NOTE

The data can be cleared by CNT_CTRL_CLR[CNT_CTRL_CLR].

Diagram



Fields

Field	Function
31-0 AXI3_ENC_RD_BYTE	AXI3 Encoded Data Read Bytes Indicates the read byte numbers of AXI3 encoded data.

49.6.14 AXI3 Decoded Pixel Write Bytes Counter (AXI3_DEC_PIXEL_WR_BYTE_CNT)

Offset

Register	Offset
AXI3_DEC_PIXEL_WR_BYTE_CNT	A4h

Function

Indicates the total write bytes of AXI3 decoded pixel.

NOTE

The data can be cleared by CNT_CTRL_CLR[CNT_CTRL_CLR].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AXI3_PIXEL_WR_BYTE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AXI3_PIXEL_WR_BYTE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 AXI3_PIXEL_WR_BYTE	AXI3 Pixel Write Bytes Indicates the write byte numbers of AXI3 decoded pixel.

49.6.15 AXI3 Decode Ancillary Write Bytes Counter (AXI3_DEC_ANC_WR_BYTE_CNT)

Offset

Register	Offset
AXI3_DEC_ANC_WR_BYTE_CNT	A8h

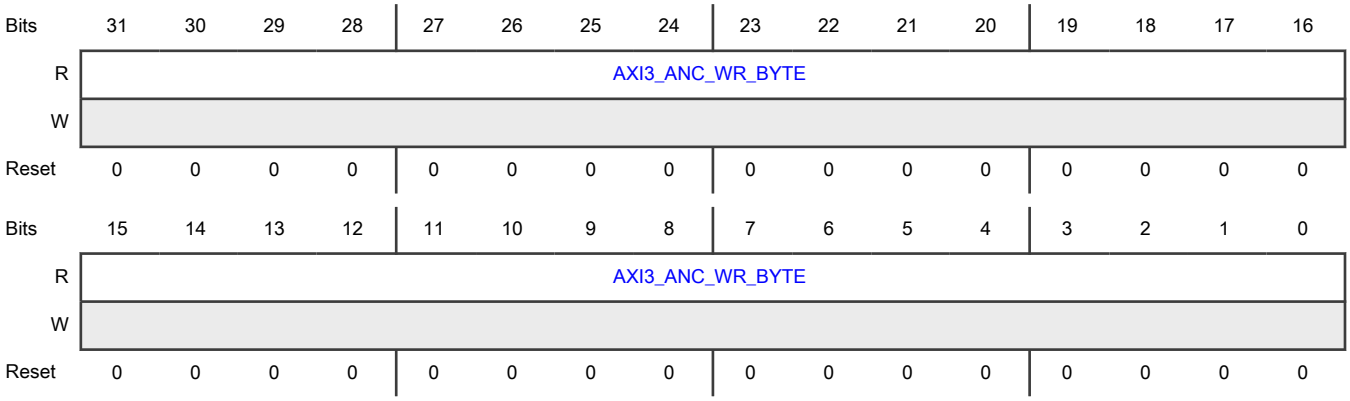
Function

Indicates the total write bytes of AXI3 decoded ancillary data.

NOTE

The data can be cleared by CNT_CTRL_CLR[CNT_CTRL_CLR].

Diagram



Fields

Field	Function
31-0 AXI3_ANC_WR_BYTE	AXI3 Ancillary Write Bytes Indicates the write type numbers of AXI3 decoded ancillary data.

Chapter 50
Vector Graphics Processing Unit (VGPU)

50.1 Chip-specific VGPU information

Table 441. Reference links to related information

Topic	Related module	Reference
Full description	VGPU	VGPU
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

50.1.1 Module instances

This device contains one instance of the VGPU module.

Table 442. VGPU instance

	VGPU
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

50.2 Overview

NOTE

This chapter describes the Vivante® IP GC555 GPU and contains copyright material disclosed with permission of VeriSilicon.

VGPU provides a high-performance UI-based graphic core to enable hardware acceleration of the OpenVG API and display vector graphics. The DECNano (Decoder/Encoder Core) series pixel compression and decompression fabric provides a unified lossy pixel compression and decompression infrastructure and support for pixel processing pipelines.

In addition to support for the OpenVG API, VGPU also supports the Vivante® VGLite API to enable the GPU acceleration through customized applications. The Vivante® VGLite API also provides an ultralow memory footprint.

50.2.1 Block diagram

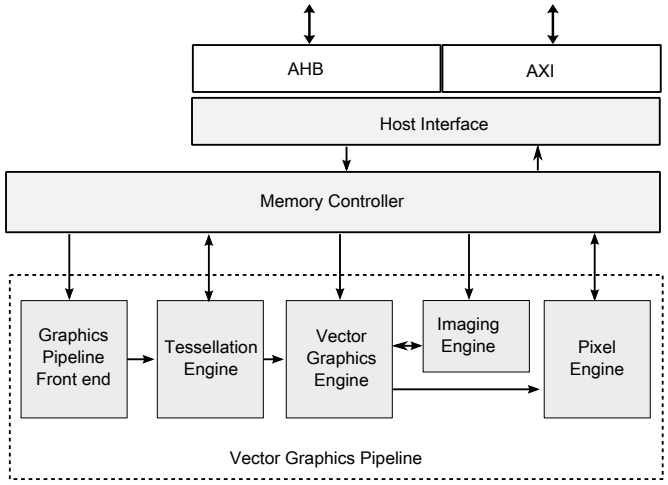


Figure 171. Block diagram with vector graphics pipeline

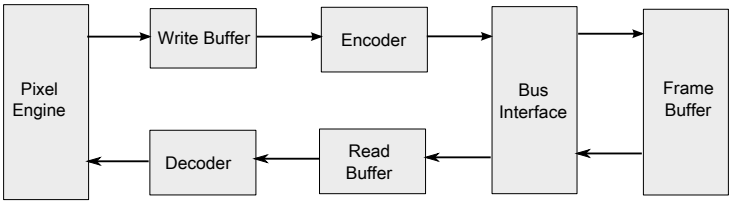


Figure 172. DECNano series architecture

The block diagram outlines the functional blocks of VGPU with a vector graphic pipeline as below:

Table 443. VGPU functional blocks

Module name	Description
Host interface	Allows the VGPU to communicate with external memory and the CPU through the bus in this block data crosses clock domain boundaries.
Memory controller	Internal memory management unit that controls the block-to-host memory request interface.
Graphics pipeline front end	Inserts high level primitives and commands into the graphics pipeline.
Tessellation engine	Transforms vertices and control points, and tessellates lines, quadratic and cubic Bezier curves.
Vector graphics engine	Rasterizer that converts primitives to pixels.
Imaging engine	Paint and image generator that colors each pixel.
Pixel engine	Renderer that combines different sources into the final pixel value. It provides pixel manipulation and filtering on rendered images.

Table continues on the next page...

Table 443. VGPU functional blocks (continued)

Module name	Description
DEC encoder	Encoder unit processes uncompressed RGB data (scanline or tiled data)
DEC decoder	Decoder unit uncompressed RGB data (scanline or tiled data)
Write buffers	Buffer for data being input from a pixel engine to the DEC encoder
Read buffers	Buffer for output data being sent from the DEC decoder to a pixel engine
Frame buffer	Output result buffer
Tessellation buffer	Tessellation result buffer

50.2.2 Features

VGPU has the following features:

- External data access: 1 x 64-bit AXI
- Configuration interface: 1 x3 2-bit AHB
- AXI version compatibility: 4 (partial)
- Clock source: clk1x
- Image source read
- Open VG 1.1
- Vivante VGLite API support
- Linear paint
- Pixel input format support
- YUY2 input support
- INDEX 1/2/4/8 support
- DECNano compression/decompression
- ETC2 decompression
- Transformation support: 3 x 3 perspective and affine

50.3 Functional description

The following sections describe VGPU functionalities.

50.3.1 Host and memory interface features

Table 444 shows the host and memory interface features of VGPU.

Table 444. Host and memory interface features

Feature	VGPU support
AHB interface	32-bit (alternately APB) for configuration
AXI interface	1 64-bit AXI interface for external memory access

Table continues on the next page...

Table 444. Host and memory interface features (continued)

Feature	VGPU support
DEC interface	Encoder and decoder have the same signals.
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
Physical address	32 bits
Read write request size support	Memory controller supports 8, 16, 32, 48, 64 byte read and write requests
Resource locks with CPU	Semaphore lock
Memory optimization	• fixed (reduced size) path and stroke caches • fixed (reduced size) tessellation window and buffer size • RGBA color transformations

50.3.2 Hardware features

Table 445 describes vector graphics hardware features.

Table 445. Graphics hardware features

Feature	Description
Coordinate systems and transformations	Image drawing uses a 3 x 3 perspective transformation matrix
Scissoring and viewport clipping	Yes
Porter-Duff blending	Yes
Blending	Porter Duff blending plus additional blend modes
Rotation	To any angle
Fill rules	Odd/even and non-zero
Paint paths	Linear and radial gradient paint paths
Paints	Solid, linear, and radial gradient
Compression/Decompression	DECNano/ETC2 (only supports decompression)
Coordinate systems and transformations	Tessellation uses 3 x 3 affine transformation
YUY2 and YUV420 video image conversion	YUY2 and YUV420 video image conversion
Anti-aliasing	4x and 16x Coverage Sample Anti-aliasing(CSAA)
Image filters and interpolations	Point, linear, and bi-linear blends
Pixel formats for image interpolation	Pixel formats for image interpolation
Transformations	sRGBA color transformation

50.3.3 Format support

Table 446. Graphics formats

Feature	VGPU Support			
Image formats	The graphics engine supports the following formats for source image and destination render targets.			
	Bit depth	Format	Supported for source image	Supported for destination
	1	INDEX1	Yes	-
	2	INDEX2	Yes	-
	4	INDEX4	Yes	-
	4	A4	Yes	-
	8	A8	Yes	Yes
	8	INDEX8	Yes	-
	8	L8	Yes	Yes
	8	A2R2G2B2	Yes	Yes
	12	NV12	Yes	-
	12	YV12	Yes	-
	16	R5G6B5	Yes	Yes
	16	A1R5G5B5	Yes	Yes
	16	YUY2	Yes	-
	16	NV16	Yes	-
	16	YV16	Yes	-
	16	A4R4G4B4	Yes	Yes
	20	ANV12	No	-
	24	A8R5G6B5	Yes	Yes
	24	R8G8B8	Yes	Yes
	24	AYUY2	No	-
	24	YV24	Yes	-
	32	A8R8G8B8	Yes	Yes
	32	X8R8G8B8	Yes	Yes
DECNano format compression support	**Bit Depth**	**Format**	**Supported for Source**	**Supported for Destination**
	32	A8R8G8B8	No	Yes
	32	X8R8G8B8	No	Yes

Table continues on the next page...

Table 446. Graphics formats (continued)

	NOTE DECNano can be bypassed by software configuration.
ETC2 format decompression support	**Bit Depth** / **Format** / **Supported for Source** / **Supported for Destination** 32 / A8R8G8B8 / Yes / -
Color component swizzle support	SWIZZLE_ABGR, SWIZZLE_ARGB, SWIZZLE_BGRA and SWIZZLE_RGBA

50.3.4 Encode and decode hardware features

Below lists the key features of the DEC compression unit.

Table 447. Encode and decode hardware features

Feature	DEC Support
Input and output pixel formats	See Data formats.
Compression scheme	Lossy compression and decompression
Compression modes	Three modes: • No sample • Horizontal only • Horizontal and vertical
Compression de-compression ratio	1.8 to 4.0 dependent on color format and compression mode
Pixel data organization support (in pixels)	Tile (4 x 4) and raster scan (16 x 1)
User controls*	GPU controls compression.
Compression units	64 bytes
Burst size	4 bytes
By-pass scheme	Uncompressed data travels through the GPU pipeline directly bypassing the DEC compression or decompression logic, and then travels across the AXI interface through standard memory-to-the memory display path. You can configure DECNano bypass via software.
Codec pixel data rate	1 pixel per cycle encode/decode for 32 bpp
Bandwidth reduction	2-3X with lossy compression

50.3.4.1 DECNano lossy data compression overview

Supported formats, tile modes and compression modes

The DECNano series supports the following formats, tile modes and compression modes.

- Formats are 32 bit-per-pixel ARGB8 and XRGB8, specified in signal format. A or X is in the upper bits and B is in the lower bits.

- Tile mode is either linear 16 x 1 pixels or tiled 4 x 4 pixels. Each tile occupies 64 bytes.
- Three compression modes are supported:
  - None
  - Horizontal sample
  - Horizontal and vertical sample. Note that this mode is not supported for linear scanlines.

One of three compression modes is specified in the signal comp_mode, where 0 = no sample, 1 = Horizontal sample of 4:1, or 2 = Horizontal and vertical sample of 1. A Nano_core_error_o=1 signal will be generated if a configuration attempts to configure a tile with a tilemode=0: 16x1 and comp_mode =2: Horizontal and Vertical Sampling.

Table 448. Compression ratios per compression mode

Compression mode	Color Subsample	
	ARGB	XRGB
No sample	1.8:1	2.4:1
Horizontal only (raster or tile)	2.4:1	3.0:1
Horizontal and vertical (tile only)	3.0:1	4.0:1

Table 449. Format, tile mode and compression mode summary for DECNano

Format	Tile Mode tilemode 0 = 16 x 1; 1 = 4 x 4	Compression Mode comp_mode 0= ; 1=no sample; 2=Horizontal 4:1; 3=Horizontal&Vertical 1
ARGB8 (format=0) or XRGB8 (format=1)	16 x 1	No sample
		H
	4 x 4	No sample
		H
		HV

Considerations

The DECNano series provide basic lossy compression capabilities. As such, it provides a limited subset of formats, tile modes, compression modes and advanced features. Note that:

- Compression and decompression are supported.
- Lossless compression is not supported. Only lossy compression/decompression is available.
- The tile to be compressed must be 64 bytes. If not, an error will result.
- The decompressed tile must be 64 bytes. If not, an error will result.
- No YUV format support is provided. Only A/XRGB8 formats are supported.
- A2RGB10 is not supported.
- Bypass is supported.
- Burst size is 4 bytes.
- Throughput is 1 RGBA or RGB pixel per cycle.

50.3.4.2 Codec data throughput and formats

50.3.4.2.1 Pixel throughput

- Compression throughput maximum with 32-bit data
  - 1 pixel/cycle for 32 bpp formats (A8R8G8B8, X8R8G8B8)
- Decompression throughput maximum with 32-bit data
  - 1 pixel/cycle for 32 bpp formats (A8R8G8B8, X8R8G8B8)

50.3.4.2.2 Alignment

The companion IP controls the alignment:

- Compression unit is 64 bytes.
- Burst size is 4 bytes.

50.3.4.2.3 Formats for encode and decode with DECNano

Supported color formats and tile mode combinations are summarized in the table below.

Table 450. Color format bit depths supported for encode and decode

Color formats	Format	Bits per component	DECNano series Support
RGB formats	A/X4R4G4B4	4 (16 bpp)	NA
	R5G6B5	16 bpp	NA
	A/X8R8G8B8	8 (32 bpp)	Yes
	A/XR10G10B10	32 bpp	NA
YUV formats	NA	NA	NA

50.3.5 Clocking

VGPU employs a simple clocking scheme. There are four independent clock domains:

- Core clock domain, which is derived from the clk1x pin,
- AHB interface clock domain for register access, which is derived from the HCLK pin, and
- AHB interface clock domain for memory access, which is derived from the ACLK pin.
- AXI interface clock domain, which is derived from the clk pin.

HCLK is the clock for register access. clk1x is the functional clock for the core logic. ACLK is the interface clock domain for memory access.

Communication between the different domains in the GPU occurs mostly by way of asynchronous FIFOs. All the clocks are asynchronous to each other. There is no communication between clock domains.

Table 451. Clock definitions

GPU clock pins	Description
HCLK	AHB interface clock for register access. It is provided external to the IP.

Table continues on the next page...

Table 451. Clock definitions (continued)

GPU clock pins	Description
ACLK	AXI or AHB interface clock for memory access. It is provided external to the IP.
clk1x	clk1x is the main core clock. It is provided external to the IP.
clk	clk is the DEC core clock. It is provided internal to the IP.

50.3.6 Interrupts

This module has no interrupts.

50.4 Software API

The following hardware platform and operating system independent APIs are available. These APIs provide user controls for optimizing the acceleration capabilities available with VGPU.

50.4.1 OpenVG 1.1 API standard for vector graphics acceleration

OpenVG is a royalty-free, cross-platform API managed by the member-funded consortium known as Khronos Group. It provides a low-level hardware acceleration interface for vector graphics libraries such as Flash and SVG. OpenVG is used for acceleration of high-quality vector graphics for user interfaces and text on small screen devices.

50.4.1.1 Core API features of OpenVG

VGPU fully supports the following key features of the Khronos API:

- Coordinate systems and transformations (image drawing uses a 3x3 perspective transformation matrix)
- Viewport clipping, scissoring and alpha masking
- Paths
- Images
- Image filters
- Paint (gradient and pattern)
- Blending
- Dithering

50.4.1.2 OpenVG rendering pipeline

The OpenVG rendering pipeline is that:

- Stage 1: Path, transformation, stroke, and paint
- Stage 2: Stroked path generation
- Stage 3: Transformation
- Stage 4: Rasterization
- Stage 5: Clipping and masking
- Stage 6: Paint generation
- Stage 7: Image interpolation
- Stage 8: Blending and anti-aliasing

50.4.2 VGLite graphics API

VGLite Graphics API is an ultralow memory footprint API, provides an easier mechanism to use the VGPU. It provides mechanisms for hardware accelerated two-dimensional vector and raster graphics, thereby allowing to develop high quality graphics application optimized for a overall resource requirements. The VGLite Graphics API allows for fine granularity in the features supported by the GPU and their memory usage.

50.4.2.1 Organization of VGLite graphics API

The VGLite API is partitioned to provide controls for three main areas of functionality:

- Initialization - for hardware and software initialization
- Blit - for raster rendering
- Draw - for draw operations

Hardware supported blit and/or draw capabilities available for software control through the API include:

Table 452. Blit and draw controls in VGLite API

Capability	GC555 GPU with VGLite API support
Fill	Non-zero or even-odd
Path coordinate formats	S8, S16, S32, and FP32
HW anti-alias levels	Low, medium, and high
Blending	SrcOver, DstOver, SrcIn, DstIn, Screen, Multiply, Additive, and Subtractive
Data formats	INDEX1/2/4/8, ARGB2/4/8, XRGB8, RGB8, A8RGB565, A4/8, L8, YUY2, and BRGA/X variants.
Path definitions	Bounding box, path quality, coordinate format, buffer pointer, path length, and path data

Chapter 51
Universal Serial Bus Controller (USB)

51.1 Chip-specific USB information

Table 453. Reference links to related information

Topic	Related module	Reference
Full description	USB	USB
USBPHY	USBPHY	USBPHY
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

51.1.1 Module instances

This device has two instances of the USB module, USB0 and USB1.

Table 454. USB instances feature support

Feature	USB0	USB1
Located in	VDD2_MEDIA	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
Connected to which PHY	USBPHY	eUSBPHY
Over-current detect support	Yes	No
Low power clock for Suspend (L2) mode	• Select 32 kHz clock by CLKCTL4_USBFCLKSEL[SEL] • Enable the clock by writing 1 to CLKCTL4_USBFCLKSEL[SEL_EN]	• Select 32 kHz clock by CLKCTL4_eUSBFCLKSEL[SEL] • Enable the clock by writing 1 to CLKCTL4_eUSBFCLKSEL[SEL_EN]
Low power clock for Sleep (L1) mode	• Select 1 MHz LPOSC or 24 MHz OSC_CLK by CLKCTL4_USBFCLKSEL[SEL] • Enable the clock by writing 1 to CLKCTL4_USBFCLKSEL[SEL_EN]	• Select 1 MHz LPOSC or 24 MHz OSC_CLK by CLKCTL4_eUSBFCLKSEL[SEL] • Enable the clock by writing 1 to CLKCTL4_eUSBFCLKSEL[SEL_EN]

Table continues on the next page...

Table 454. USB instances feature support (continued)

Feature	USB0	USB1
External signals	OC from GPIO, PWR to GPIO	No external signals
Host mode supports wake on over-current	Yes	No

51.2 Overview

USB provides a high-performance functionality that conforms to *Universal Serial Bus Specification Revision 2.0*.

USB supports *USB 2.0 Link Power Management (LPM) Addendum* and *Errata for LPM*.

USB includes a port that you can use either as a downstream (host) or an upstream (device) port.

USB supports a PHY that conforms to *Embedded USB 2.0 Physical Layer (eUSBPHY) supplement to the USB revision 2.0 specification (revision 1.1)*.

51.2.1 Block diagram

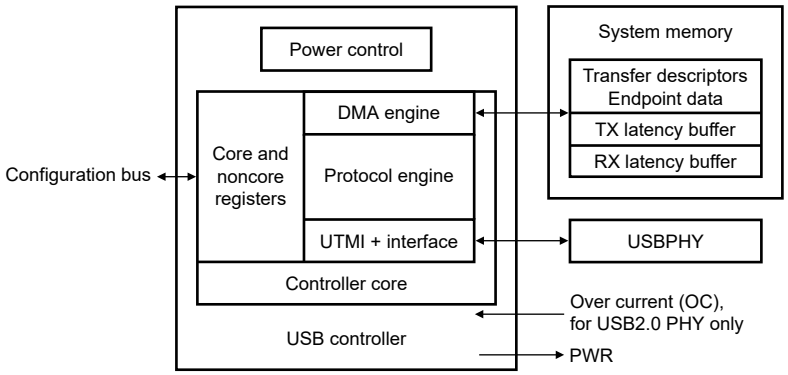


Figure 173. Block diagram

51.2.2 Features

- Compatibility with the enhanced host controller interface (EHCI) specification in Host mode
- High-speed (HS), full-speed (FS), and low-speed (LS) operations in Host mode
- HS and FS operations in Device mode
- Eight bidirectional endpoints in Device mode
- Low-Power Suspend mode (L2) with local and remote wake-up capability
- Low-Power Sleep mode (L1) with local and remote wake-up capability
- Embedded DMA controller
- UTMI+ interface to PHY

51.3 Functional description

51.3.1 Register interface

USB configuration, control, and status registers are divided into the categories shown in the following table.

EHCI registers are listed along with device registers to show the complementary nature of host and device control.

Table 455. Core register sets

Offset	Register sets	Functional description
000h–07Ch	Identification registers	• Used to declare the subordinate interface presence • Include the complete set of hardware configuration parameters
100h–124h	Static, read-only capability registers	• Define the software limits, restrictions, and capabilities of host and device controller implementation • Used as parameters to the host and device controller driver
080h–0FCh 140h–1FCh	Operational registers	• Dynamic control or status registers that can have read-only, read/write, or read/write-to-clear accesses • Used to control and monitor the operational state of the host and device controller

51.3.1.1 Configuration, control, and status register sets

Table 456. Device and host capability registers

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
000	4	Identification (ID)	O[1]	O
004	4	Hardware General (HWGENERAL)	O	O
008	4	Host Hardware Parameters (HWHOST)	X[2]	O
00C	4	Device Hardware Parameters (HWDEVICE)	O	X
010	4	TX Buffer Hardware Parameters (HWTXBUF)	O	O
014	4	RX Buffer Hardware Parameters (HWRXBUF)	O	O
018–07F	—	Reserved	—	—
080	4	General Purpose Timer 0 Load (GPTIMER0LD)	O	O
084	4	General Purpose Timer 0 Controller (GPTIMER0CTRL)	O	O
088	4	General Purpose Timer 1 Load (GPTIMER1LD)	O	O
08C	4	General Purpose Timer 1 Controller (GPTIMER1CTRL)	O	O
090	4	System Bus Configuration (SBUSCFG)	O	O
094–09F	—	Reserved	—	—
100	1	Capability Registers Length (CAPLENGTH)	O	O
101	—	Reserved	—	—

Table continues on the next page...

Table 456. Device and host capability registers (continued)

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
102	2	Host Controller Interface Version (HCIVERSION)	X	O
104	4	Host Controller Structural Parameters (HCSPARAMS)	X	O
108	4	Host Controller Capability Parameters (HCCPARAMS)	X	O
10C–11F	—	Reserved	—	—
120	2	Device Controller Interface Version (DCIVERSION)	O	X
122	2	Reserved	—	—
124	4	Device Controller Capability Parameters (DCCPARAMS)	O	X
128–13F	—	Reserved	—	—
140	4	USB Command (USBCMD)	O	O
144	4	USB Status (USBSTS)	O	O
148	4	Interrupt Enable (USBINTR)	O	O
14C	4	USB Frame Index (FRINDEX)	O	O
150	4	Reserved	—	—
154	4	Frame List Base Address (PERIODICLISTBASE)	X	O
	—	Device Address (DEVICEADDR)	O	X
158	4	Next Asynchronous Address (ASYNCLISTADDR)	X	O
	4	Endpoint List Address (ENDPTLISTADDR)	O	X
15C	4	Reserved	—	—
160	4	Programmable Burst Size (BURSTSIZE)	O	O
164	4	TX FIFO Fill Tuning (TXFILLTUNING)	X	O
168	4	Reserved	—	—
170	4	Reserved	—	—
178	4	Endpoint NAK (ENDPTNAK)	O	X
17C	4	Endpoint NAK Enable (ENDPTNAKEN)	O	X
184	4	Port Status and Control (PORTSC1)	O	O

Table continues on the next page...

Table 456. Device and host capability registers (continued)

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
188–1A3	—	Reserved	—	—
1A8	4	USB Device Mode (USBMODE)	O	O
1AC	4	Endpoint Setup Status (ENDPTSETUPSTAT)	O	X
1B0	4	Endpoint Prime (ENDPTPRIME)	O	X
1B4	4	Endpoint Flush (ENDPTFLUSH)	O	X
1B8	4	Endpoint Status (ENDPTSTAT)	O	X
1BC	4	Endpoint Complete (ENDPTCOMPLETE)	O	X
1C0 1C4 1C8 1CC 1D0 1D4 1D8 1DC	64	Endpoint Control 0 (ENDPTCTRL0) Endpoint Control 1 (ENDPTCTRL1) Endpoint Control 2 (ENDPTCTRL2) Endpoint Control 3 (ENDPTCTRL3) Endpoint Control 4 (ENDPTCTRL4) Endpoint Control 5 (ENDPTCTRL5) Endpoint Control 6 (ENDPTCTRL6) Endpoint Control 7 (ENDPTCTRL7)	O	X

1. O means that the register is available in Host and Device operation modes.
2. X means that the register is reserved in Host and Device operation modes.

51.3.1.2 Identification registers

USB uses identification registers to declare the subordinate interface presence. They include hardware configuration parameters.

51.3.2 Host data structures

For detailed information, see the "Data Structures" chapter in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

51.3.3 Host operational model

For detailed information, see the "Operational Model" chapter in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

51.3.4 EHCI deviation

For the purposes of a dual-role host or device controller with support for On-The-Go (OTG) applications, you must deviate from EHCI specification. Device and OTG operations are not specified in EHCI; therefore, the implementation supported in this core is proprietary.

The Host mode operation of the core is EHCI compatible, with a few minor differences documented in this section. The following is a summary of these deviations.

Table 457. EHCI deviation

Deviations	Description
Embedded transaction translator	Allows direct attachment of the FS and LS devices in Host mode without the need for a companion controller (see Embedded transaction translator function for more information).
Device operation	Generally disabled; therefore, Device mode is mostly transparent when in Host mode (exceptions are provided in the following sections). See Device operation for more information.
Embedded design interface	Does not have a PCI; therefore, the PCI configuration registers described in the EHCI specification are not applicable (see Embedded design interface for more information).
OTG operation	Includes an OTG for port #1.

51.3.4.1 Embedded transaction translator function

USB supports directly connected FS and LS devices, without requiring a companion controller, by including the capabilities of a USB 2.0 HS hub transaction translator.

Although there is no separate transaction translator block in the system, the transaction translator function that is normally associated with an HS hub is implemented within the DMA and protocol engine blocks. The embedded transaction translator function is an extension of the EHCI interface, but makes use of the standard data structures and operational models that already exist in the EHCI specification to support FS and LS devices.

51.3.4.1.1 Discovery-EHCI deviation

In a standard EHCI controller design, the EHCI host controller driver detects an FS or LS device by analyzing whether PORTSC1[PE] becomes 1 after the port reset operation.

PORTSC1[PE] becomes 1 in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate an HS connection (that is, after chirp completes successfully). Because this controller has an embedded transaction translator, PORTSC1[PE] always becomes 1 after the port reset operation, regardless of the host device chirp result. PORTSC1[PSPD] indicates the resulting port speed. Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected FS and LS devices or hubs.

Table 458. Summary of EHCI

Standard EHCI	EHCI with embedded transaction translator
After PORTSC1[PE] becomes 1 following a connection and reset sequence, the device or hub is assumed to be in HS.	After PORTSC1[PE] becomes 1 following a connection and reset sequence, PORTSC*n* indicates the device or hub speed.
FS and LS devices are assumed to be downstream from an HS hub; therefore, all port-level controls are performed using the hub class through the nearest hub.	FS and LS devices can either be downstream from an HS hub or directly attached. When an FS or LS device is downstream from an HS hub, the port-level control is performed using the hub class through the nearest hub. When an FS or LS device is directly attached, the port-level control is accomplished using PORTSC*n*.
FS and LS devices are assumed to be downstream from an HS hub with HubAddr = X, where HubAddr > 0 and HubAddr is the address of the hub where the bus transitions from HS to either FS or LS (that is, a split target hub).	FS and LS devices can be either downstream from an HS hub with HubAddr = X [HubAddr > 0] or directly attached, where HubAddr = 0 and HubAddr is the address of the root hub where the bus transitions from HS to either FS or LS (that is, a split target hub is the root hub).

51.3.4.1.2 Operational model

The operational models support the behavior of the transaction translator (see the USB 2.0 specification) and support the EHCI controller moving packets between the system memory and a USB-HS hub. Because the embedded transaction translator exists within the host controller, there is no physical bus between the EHCI host controller driver and the USB FS and LS bus.

The following sections briefly discuss the operational model for how the EHCI and transaction translator operational models are combined without a physical bus existing between them. These sections assume that you are familiar with both the EHCI and USB 2.0 transaction translator operational models.

51.3.4.1.2.1 Microframe pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the host (H) and the bus (B). The embedded transaction translator must use the same pipeline algorithms specified in the USB 2.0 specification for a hub-based transaction translator.

All periodic transfers begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the microframe pipeline implemented in the embedded transaction translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 are ready to execute on the bus in B-frame 0.

When programming the S-mask and C-mask fields in the EHCI data structures to schedule periodic transfers for the embedded transaction translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming those fields for downstream hub-based transaction translators.

After periodic transfers are exhausted, any stored asynchronous transfer is moved. Asynchronous transfers are opportunistic; they must execute whenever possible, and their operation is not tied to H-frame and B-frame boundaries. However, an asynchronous transfer cannot babble through the start of B-frame 0 (SOF).

51.3.4.1.2.2 Split-state machines

The start- and complete-split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded transaction translator.

A start- or complete-split operation occurs by requesting the split to the HS hub. The start-complete split operation is an internal operation to the embedded transaction translator. The following table summarizes the conditions where handshakes are emulated from an internal state instead of actual handshakes to the HS split bus traffic.

Table 459. Summary of handshake conditions

Condition	Emulate TT response
Start-split: All asynchronous buffers full	NAK
Start-split: All periodic buffers full	ERR
Start-split: Success for start of an asynchronous transaction	ACK
Start-split: Start periodic transaction	No handshake (OK)
Complete-split: Failed to find transaction in queue	Bus time out
Complete-split: Transaction in queue is busy	NYET
Complete-split: Transaction in queue is complete	Actual handshake from an LS or FS device

51.3.4.1.2.3 Asynchronous transaction scheduling and buffer management

Table 460. Asynchronous transaction scheduling and buffer management

USB 2.0 specification items	Implementation in the embedded transaction translator
11.17.3	Sequencing is provided and a packet length estimator ensures that no FS or LS packet babbles exist in the SOF time.
11.17.4	Transaction tracking happens for two data pipes.

51.3.4.1.2.4 Periodic transaction scheduling and buffer management

Table 461. Periodic transaction scheduling and buffer management

USB 2.0 specification items	Implementation in the embedded transaction translator
11.18.6.[1-2]	• Abort of pending start-split conditions: — End of frame (EOF), and not started in six microframes — Idle for more than four microframes • Abort of pending complete-split conditions: — EOF — Idle for more than four microframes
11.18.[7-8]	• Transaction tracking for up to 16 data pipes • Complete-split transaction searching **NOTE** There is no data schedule mechanism for these transactions other than the microframe pipeline. The embedded TT assumes that the number of packets scheduled in a frame does not exceed the frame duration (1 ms); otherwise, it results in an undefined behavior.

51.3.4.1.2.5 Multiple transaction translators

HCSPARAMS[N_TT] indicates the maximum number of embedded transaction translators that USB currently supports.

51.3.4.2 Device operation

The coexistence of a device operational controller within the host controller has little effect on the EHCI compatibility for host operation, except as discussed in the following sections.

51.3.4.2.1 USBMODE

You cannot initialize the dual-role controller in Host or Device modes. Considering this, you must program USB Device Mode (USBMODE) for a host operation before the EHCI host controller driver can begin EHCI host operations.

51.3.4.2.2 Fields with nonzero values in USB core registers

Some of the reserved fields and reserved addresses in the capability and operational registers are used in Device mode. For Device mode, you must adhere to the following requirements:

- Always write 0 to all EHCI reserved fields (some of which are device fields) in the operational registers. This is an EHCI requirement of the device controller driver.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusively for the device are undefined in Host mode.

51.3.4.2.3 SOF interrupt

The start of frame (SOF) interrupt for Device mode is shared as a free-running 125 µs interrupt for Host mode.

EHCI does not specify this interrupt, but it has been added for convenience and as a potential software time base. See USB Status (USBSTS) and Interrupt Enable (USBINTR) for more information.

51.3.4.3 Embedded design interface

This is an embedded USB host controller, defined by the EHCI specification, and thus does not implement the PCI configuration registers.

51.3.4.3.1 Frame adjust register

Considering that the optional PCI configuration registers are not included in this implementation, there is no corresponding bit-level timing adjustment. The start of microframes is timed precisely to 125 µs using the transceiver clock as a reference clock. That is, a 30 MHz transceiver clock for 16-bit physical interfaces.

51.3.4.4 Miscellaneous variations from EHCI

51.3.4.4.1 Programmable physical interface behavior

This design supports multiple physical interfaces that can operate in different modes if you program the core with software-programmable physical interface modes. Software programmability allows you to select the physical interface part during the board design phase, instead of selection during the chip design phase. Port Status and Control (PORTSC1) provides the control fields for selecting the physical interface operating mode, a capability that EHCI does not define.

51.3.4.4.2 Device discovery

51.3.4.4.2.1 Port reset

The port connect methods that EHCI specifies require you to set PORTSC1[PR] for a duration of 10 ms. Because of the complexity involved in supporting the attachment of devices that are not high speed, there are counters already present in the design that can count the 10 ms reset pulse to alleviate the software requirement to measure this duration. Perform the following procedure to establish the basic connection.

Table 462. Port reset

Step	Purpose	Programming	Notes
1	Change the port.	Wait for the port change interrupt.	A port connect change occurs to notify the host controller driver that a device is attached.
2	Reset the device.	Write 1 to PORTSC1[PR].	—
3	Reset the device after 10 ms.	Write 0 to PORTSC1[PR].	This step, which is necessary in a standard EHCI design, can be omitted with this implementation. If the EHCI host controller driver attempts to write 0 to PORTSC1[PR]

Table continues on the next page...

Table 462. Port reset (continued)

Step	Purpose	Programming	Notes
			when a reset is in progress, the write is ignored and the reset continues until completion.

A port enable change occurs to notify the host controller that the device in now operational, and at this point, the port speed is determined.

51.3.4.4.2.2 Port speed detection

After the port change interrupt indicates that a port is enabled, the EHCI stack must determine the port speed. Unlike the EHCI implementation, which reassigns the port owner for any device that does not connect at HS, this host controller supports direct attachment of non-HS devices.

Consider the following differences for port-speed detection:

- PORTSC1[PO] is read-only and always reads 0.
- PORTSC1[PSPD] provides the current operating speed of the port to the host controller driver.

51.3.4.4.3 Port Test Control mode

This mode behaves exactly as described in EHCI. An alternate host controller driver procedure is neither necessary nor supported.

51.3.5 Device data structures

This section defines the interface data structures used to communicate control, status, and data between the device controller driver (DCD) software and device controller.

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device queue heads and transfer descriptors.

NOTE

You must ensure that no interface data structure reachable from the device controller spans a 4 KB page boundary.

The data structures defined in this chapter are, from the device controller's perspective, a mix of read-only (RO) and read/write (R/W) fields. The device controller must preserve the RO fields on all data structure writes.

Figure 174 shows the organization of the device endpoint queue head (dQH).

Endpoint queue heads are arranged in an array, in a continuous area of memory indicated by Endpoint List Address (ENDPTLISTADDR). The even-numbered device queue heads in the list support receive endpoints (Out/Setup) and the odd-numbered queue heads in the list are used for transmit endpoints (In/Interrupt). That is, the device queue heads in the list with an even offset support receiver endpoints (Out) and the queue heads in the list with an odd offset are used for the transmit endpoint (In). The device controller indexes into this array based on the endpoint number received from the USB bus. All information necessary to respond to transactions for all primed transfers is contained in this list so that the device controller can readily respond to incoming requests without having to traverse a linked list.

NOTE

The endpoint queue head list must be aligned to a 2 KB boundary.

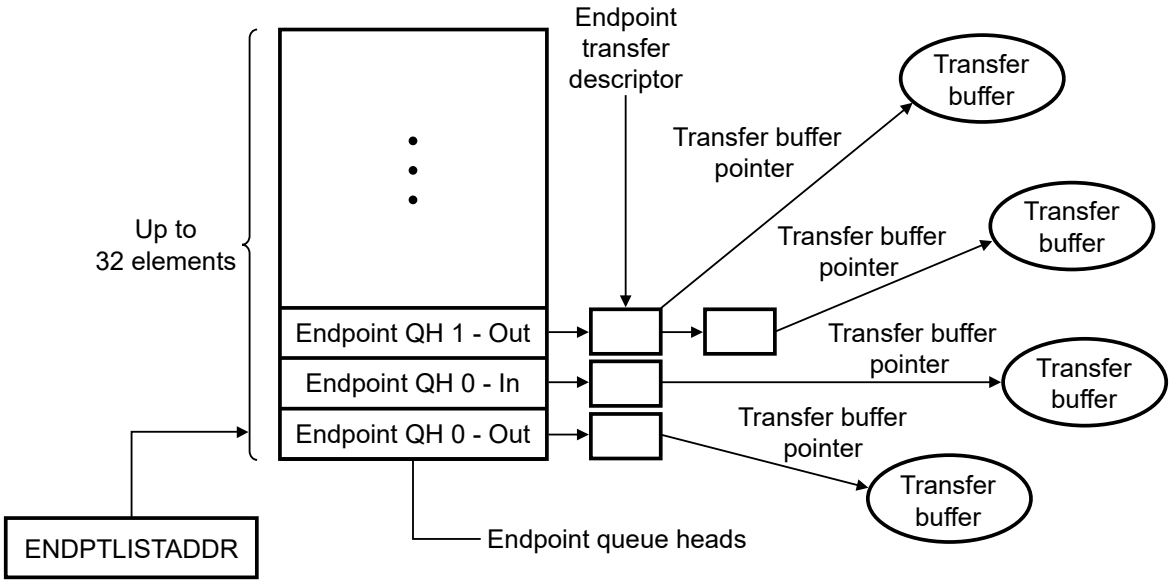


Figure 174. Endpoint queue head organization

51.3.5.1 Device endpoint queue head (dQH)

The dQH is where all transfers for a given endpoint are managed. It is a 48-byte data structure, but must be aligned on 64-byte boundaries.

During priming of an endpoint, the device transfer descriptor (dTD) is copied into the overlay area of the dQH, which starts at the next dTD pointer DWORD and continues through the end of the buffer pointer. After a transfer is complete, the dTD status DWORD is updated in the dTD indicated by the current dTD pointer. When a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the device controller can access the required information with minimal latency.

Table 463. dQH structure [1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MULT		ZLT	0		Maximum packet length											IOS	0														
Current dTD pointer																											0				
Next dTD pointer																											0				T [2]
0	Total bytes															IOC	0			MULTO		0		Status							
Buffer pointer (page 0)																				Current offset											

Table continues on the next page...

Table 463. dQH structure [1] (continued)

Buffer pointer (page 1)	Reserved
Buffer pointer (page 2)	Reserved
Buffer pointer (page 3)	Reserved
Buffer pointer (page 4)[2]	Reserved
Reserved	
Setup buffer bytes 3-0	
Setup buffer bytes 7-4	

1. The shaded cells indicate host controller read/write fields and unshaded cells indicate host controller read-only fields.
2. Transfer overlay starts at T and continues through the buffer pointer (page 4).

51.3.5.1.1 Endpoint capabilities and characteristics

This DWORD specifies static information about the endpoint; in other words, this information does not change over the lifetime of the endpoint. The device controller software must not attempt to modify this information when the corresponding endpoint is enabled.

Table 464. Endpoint capabilities and characteristics

Field	Function
31-30 MULT	Multiplier Indicates the number of packets executed per transaction description. • 00 – Execute *n* transactions as demonstrated by the USB variable length packet protocol, where *n* is computed using dQH and dTD. • 01 – Execute one transaction • 10 – Execute two transactions • 11 – Execute three transactions **NOTE** You must write 00b to this field for non-ISO endpoints, and 01b, 10b, or 11b, as needed, for ISO endpoints.
29 ZLT	Zero-Length Termination Select Enables a zero-length packet for use to terminate transfers, where the total transfer length is a multiple of the maximum packet length. This field is not relevant for isochronous transfer types. • 0 – Enable a zero-length packet to terminate transfers equal to a multiple of the maximum packet length (default) • 1 – Disable the zero-length packet on transfers that are equal in length to a multiple maximum packet length

Table continues on the next page...

Table 464. Endpoint capabilities and characteristics (continued)

Field	Function
28-27 Reserved	Reserved for future use; must be 0.
26-16 Maximum Packet Length	Maximum Packet Length Corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field can contain is 400h (1024).
15 IOS	Interrupt On Setup Used for control type endpoints to indicate whether USBINTR is configured in response to a setup being received.
14-0 Reserved	Reserved for future use; must be 0.

51.3.5.1.2 Transfer overlay-endpoint queue head

The seven DWORDS in the overlay area represent a transaction working space for the device controller.

The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, it does not read the associated endpoint.

After an endpoint is ready, the device controller copies the dTD into the queue head overlay area. Until a transfer is expired, you must not write to the queue head overlay area or the associated transfer descriptor. After the transfer is complete, the device controller writes the results back to the original transfer descriptor and advances the queue.

See Device endpoint transfer descriptor (dTD) for a description of the overlay fields.

51.3.5.1.3 Current dTD pointer

The device controller uses the current dTD pointer to locate the transfer in progress. This word is for device controller (hardware) use only, and the DCD software must not modify it.

The following table describes the dTD pointer.

Table 465. Current dTD pointer

Bit	Description
31-5 Current dTD	Current dTD Points to the dTD that is represented in the transfer overlay area. The device controller modifies this field to the next dTD pointer during endpoint priming or queue advance.
4-0 Reserved	Reserved for future use; must be set to 0.

51.3.5.1.4 Setup buffer

The setup buffer is a dedicated storage for the 8-byte data that follows a setup PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it, and only the RX queue head is used for receiving setup data packets.

Table 466. Setup buffer bytes

DWORD	Bits	Description
1	31-0	Setup Buffer 0 Contains bytes 3–0 of an incoming setup buffer packet. The device controller writes to it so you can read it.
2	31-0	Setup Buffer 1 Contains bytes 7–4 of an incoming setup buffer packet. The device controller writes to it so you can read it.

51.3.5.2 Device endpoint transfer descriptor (dTD)

The dTD informs the device controller of the location and quantity of data to be sent and received for a given transfer.

The DCD must not attempt to modify any field in an active dTD except the next like pointer, which you must modify only as described in Managing transfers with transfer descriptors.

Table 467 describes the dTD structure.

Table 467. dTD structure[1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next link pointer																											0				T
0	Total bytes															IOC	0			MULTO		0		Status							
Buffer pointer (page 0)																				Current offset											
Buffer pointer (page 1)																				0	Frame number										
Buffer pointer (page 2)																				Reserved											
Buffer pointer (page 3)																				Reserved											
Buffer pointer (page 4)																				Reserved											

1. The shaded cells indicate host controller read/write fields and unshaded cells indicate host controller read-only fields.

Table 468 describes the dTD pointer.

Table 468. Next dTD pointer

Field	Function
31-5 Next Transfer Element Pointer	Next Transfer Element Pointer Contains the physical memory address of the next dTD to be processed. This field corresponds to memory address signals [31:5], respectively.
4-1 Reserved	Reserved for future use; must be 0.
0 T	Terminate Indicates to the device controller whether there are valid entries in the queue. • 0 – Pointer is valid (points to a valid transfer element descriptor) • 1 – Pointer is invalid

Table 469 describes the dTD token.

Table 469. dTD token

Field	Function
31 Reserved	Reserved for future use; must be 0.
30-16 Total Bytes	Total Bytes Specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes moved during the transaction, after the successful completion of the transaction. The maximum value you can store in the field is 5 × 4KB (5000h). This is the maximum number of bytes that 5-page pointers can access. Although it is possible to create a transfer up to 20 KB, this assumes that the first offset into the first page is 0. When the offset cannot be predetermined, crossing past the fifth page can be guaranteed by limiting the total bytes to 16 KB. Therefore, the maximum recommended transfer is 16 KB (4000h). If the value of this field is 0 when the host controller fetches this transfer descriptor (and the active bit is 1), the device controller executes a zero-length transaction and retires the transfer descriptor. "In" transfers do not require that dTD be an even multiple of the Maximum Packet Length field in dQH. If you build such a transfer descriptor for an "In" transfer, the last transaction is always less than the value of the Maximum Packet Length field.
15 IOC	Interrupt On Complete Indicates whether USBINTR will be configured in response to the device controller being finished with this dTD.
14-12 Reserved	Reserved for future use; must be 0.
11-10 MULTO	Multiplier Override

Table continues on the next page...

Table 469. dTD token (continued)

Field	Function
	Can be used for transmit-ISO (that is, ISO-IN) to override the multiplier in the queue head. This field must be zero for all packet types that are not transmit-ISO. For example, if QH.multiplier = 3, the Maximum Packet Size field = 8, the Total Bytes field = 15, and the MULTO field = 0 (default), then three packets are sent: {Data2(8); Data1(7); Data0(0)}. If QH.multiplier = 3, the Maximum Packet Size field = 8, the Total Bytes field = 15, and the MULTO field = 2, then two packets are sent: {Data1(8); Data0(7)}. For maximal efficiency, you must calculate MULTO = greatest integer of (the Total Bytes field ÷ the Maximum Packet Size field). In cases where the Total Bytes field = 0, the MULTO field must be 1. For non-ISO and non-TX endpoints, you must write 0 to the MULTO field.
9-8 Reserved	Reserved for future use; must be 0.
7-0 Status	Status Used by the device controller to communicate individual command execution states back to the device controller software. This field contains the status of the last transaction performed on this dTD. The following are the bit encodings: • 0, 2, and 4 - Reserved • 3 - Transaction error • 5 - Data buffer error • 6 - Halted • 7 - Active

Table 470 describes the dTD buffer page pointer list.

Table 470. dTD buffer page pointer list

Field	Function
31-12 Buffer Pointer	Selects the page offset in memory for the packet buffer. Nonvirtual memory systems typically set the buffer pointers to a series of incrementing integers.
0, 11-0 Current Offset	Indicates the current offset that offsets into the 4 KB buffer where the packet will begin.
1, 10-0 Frame Number	Indicates the frame number in which a packet finishes. You typically use it to correlate relative completion times of packets on an ISO endpoint. The device controller writes to this field.

51.3.6 Device operational model

The function of the device operation is to transfer a request in the memory image to and from USB.

The device controller performs the data transfers (pointed to by the queue head) using a set of linked list transfer descriptors. The following sections explain the use of the device controller from the point-of-view of the DCD, and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

51.3.6.1 Port state and control

From a chip or system reset, the device controller enters the powered state. A transition from the powered state to the attach state occurs when USBCMD[RS] = 1.

After receiving a reset on the bus, the port enters the defaultFS or defaultHS state in accordance with the reset protocol described in Appendix C.2 of *USB 2.0 Specification Rev. 2.0*.

The following state diagram depicts the state of a USB 2.0 device.

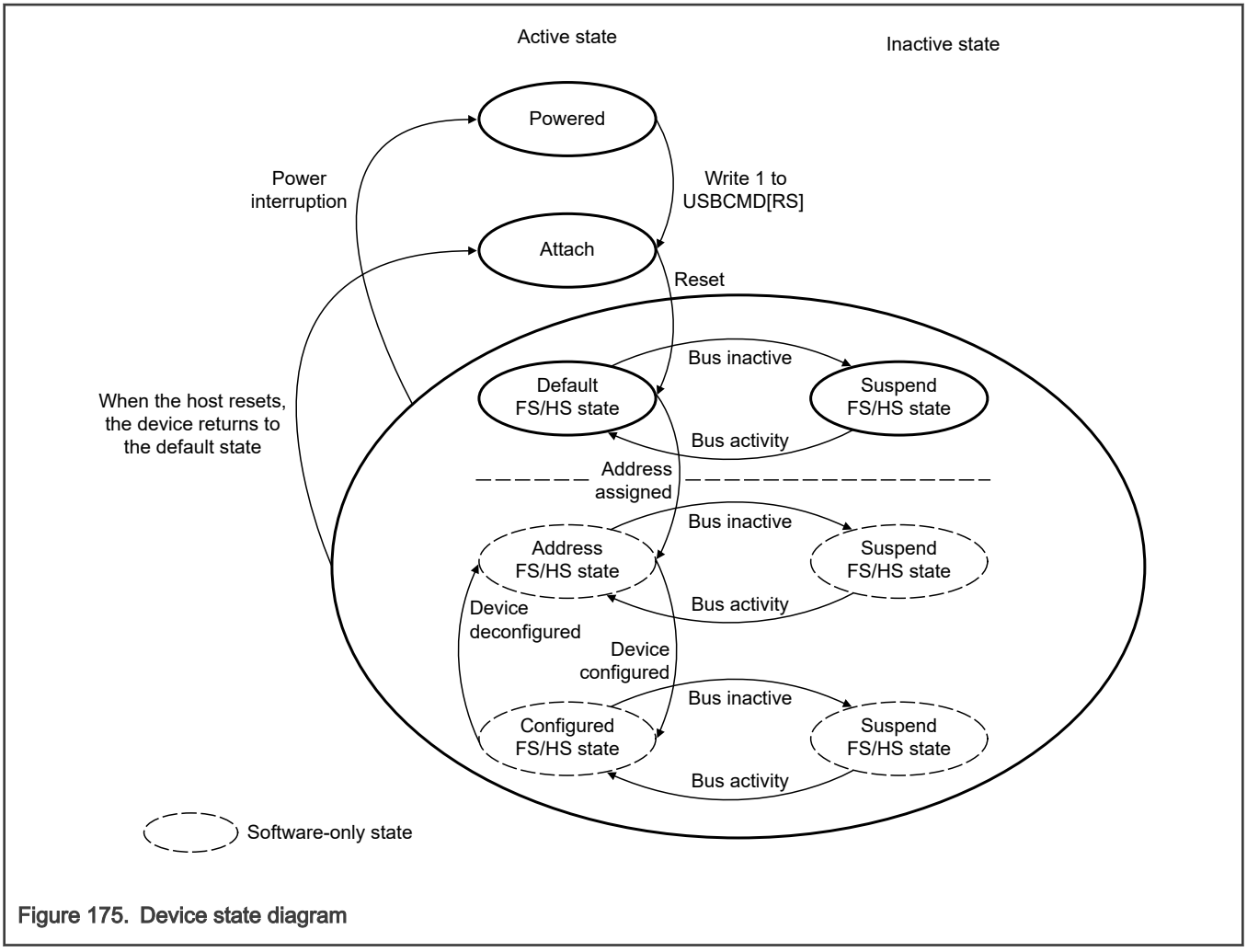


Figure 175. Device state diagram

The powered, attach, defaultFS, defaultHS, suspendFS, and suspendHS states are implemented in the device controller and communicated to the DCD using the following device controller state information fields:

- USBSTS[SLI]
- USBSTS[URI]
- USBSTS[PCI]
- PORTSC1[HSP]

It is the responsibility of the DCD to maintain a state variable to differentiate between the defaultFS and defaultHS states, and the Address and Configured states. A change of state from Default to Address or Configured is part of the enumeration process described in the device framework section of the USB 2.0 specification.

As a result of entering the Address state, the DCD must program DEVICEADDR.

Entry into the Configured state indicates that you have properly initialized all endpoints to be used in the operation of the device by programming the ENDPTCTRL*x* registers and initializing the associated queue heads.

51.3.6.1.1 Bus reset

The host uses a bus reset to initialize downstream devices.

When a bus reset is detected, the device controller renegotiates its attachment speed, resets the device address to 0, and notifies the DCD by interrupt (assuming USBINTR[URE] = 1). After a reset is received, all endpoints (except endpoint 0) are disabled, and the device controller cancels any primed transactions. The following table explains the concept of priming, and the tasks that DCD must perform when a reset is received.

Table 471. Bus reset

Step	Purpose	Programming
1	Clear all setup token semaphores.	Read Endpoint Status (ENDPTSTAT) and write the same value back to ENDPTSTAT.
2	Clear all the endpoint-complete status fields.	Read Endpoint Complete (ENDPTCOMPLETE) and write the same value back to ENDPTCOMPLETE.
3	Cancel primed status.	Wait until all fields in Endpoint Prime (ENDPTPRIME) become 0 and then write FFFFFFFFh to Endpoint Flush (ENDPTFLUSH).

Read PORTSC1[PR] and make sure that it is still active. A USB reset occurs for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before the end of the reset occurs; otherwise, a hardware reset of the device controller is recommended. This is a rare scenario.

You can perform a hardware reset by writing 1 to USBCMD[RST]. By writing 0 to USBCMD[RS], a hardware reset causes the device to detach from the bus. Thus, the DCD must completely reinitialize the device controller after a hardware reset.

Additionally, you must free up all allocated dTDs because the device controller will no longer execute them. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs are allocated.

At this time, the DCD can release the control back to the OS because no further changes to the device controller are permitted until a port change detect is indicated.

After a port change detect, the device reaches the default state and the DCD can read Port Status and Control (PORTSC1) to determine whether the device is operating in FS or HS mode. At this time, the device controller reaches its normal operating mode and DCD can begin enumeration according to *USB 2.0 specification chapter 9: Device Framework*.

NOTE

The device DCD may use the FS or HS mode information to determine the bandwidth mode of the device.

In some applications, it may not be possible to enable one or more pipes when in FS mode. There is no difference in DCD operations in FS and HS modes, except for the data rate issue.

51.3.6.1.2 Suspended state and Resume signal

Table 472. Operational models for Suspended state and Resume signal

Suspended	Resume
To conserve power, USB devices automatically enter Suspended state when the device observes no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. The attached devices must be prepared to suspend at any time when they are powered, regardless of whether they are assigned a nondefault address, are configured, or neither of the two conditions exist. Bus activity may cease because of the host entering a suspend mode of its own. In addition, a USB device must enter the Suspended state when the attached hub port is disabled. A USB device exits Suspend mode when there is bus activity. It may also request the host to exit Suspend mode or selectively suspend by using electrical signaling to indicate remote wake-up. The ability of a device to signal remote wake-up is optional. If the USB device is capable of remote wake-up signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wake-up signaling must be disabled. The device controller moves into the Suspended state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the Suspended state, an interrupt notifies the DCD (assuming that PORTSC1[SUSP] is 1, which means, the device controller is suspended). The DCD response, when the device controller is suspended, is application-specific and may involve switching to a low-power operation. See the USB 2.0 specification for information about bus power limits in Suspended state.	If the device controller is suspended, its operation is resumed when an active signal is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port by writing 1 to PORTSC1[FPR] when the device is in the Suspended state. Sending resume signal to an upstream port must cause the host to issue resume signaling and bring the suspended bus segment (one more device) back to the active condition. **NOTE** Before using resume signaling, the host must enable it via the set feature command defined in the device framework (chapter 9 of the USB 2.0 specification).

51.3.6.2 Managing endpoints

The USB 2.0 specification defines an endpoint (also called a device endpoint or an address endpoint) as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device.

The combination of the endpoint number and endpoint direction specifies the endpoint address.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a control type data channel used for device discovery and enumeration. Other types of endpoints that USB supports include bulk, interrupt, and isochronous. Each endpoint type has a specific behavior related to packet response and error handling.

The USB device controller hardware supports up to eight endpoint numbers.

Each endpoint direction is essentially independent, and you can configure it with differing behaviors in each direction. For example, the DCD can configure endpoint "1-In" as a bulk endpoint and endpoint "1-Out" as an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, always a control endpoint and uses both directions.

Each endpoint direction requires a queue head allocated in memory. To support the eight endpoint numbers, 16 queue heads are required. The operation of an endpoint and use of queue heads are described later in this document.

See the USB 2.0 specification for more information about endpoint operations.

51.3.6.2.1 Endpoint initialization

After hardware reset, all endpoints except endpoint 0 are uninitialized and disabled. The DCD must configure and enable each endpoint by configuring ENDPTCTRL*x*.

Each 32-bit ENDPTCTRL*x* is split into an upper and a lower half. The lower half of ENDPTCTRL*x* is used to configure the receive or "Out" endpoint, and the upper half is used to configure the corresponding transmit or "In" endpoint. You must configure the control endpoints in the same way for both the upper and lower halves of ENDPTCTRL*x*; otherwise, the behavior is undefined. The following table shows the fields and corresponding values for device controller endpoint initialization.

Table 473. Device controller endpoint initialization: configuration settings

Step	Purpose	Programming
1	Reset the PID sequence.	Write 1 to ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR].
2	Configure receive data toggle sequencing.	Write 0 to ENDPTCTRL*x*[TXI] and ENDPTCTRL*x*[RXI].
3	Select the endpoint type.	Configure ENDPTCTRL*x*[TXT] and ENDPTCTRL*x*[RXT].
4	Configure the endpoint stall condition.	Write 0 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS].

51.3.6.2.2 Stalling

The following are the occasions in which the device controller may need to return a STALL response to the host:

- In case of a functional stall: It is a condition that DCD sets, as described in the USB 2.0 device framework. This kind of a stall is only used on noncontrol endpoints, and can be enabled in the device controller by writing 1 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS], associated with the given endpoint and the given direction. In a functional stall condition, the device controller continues to return STALL responses to all transactions occurring on the respective endpoint and direction until DCD writes 0 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS].
- In case of a protocol stall: Unlike a function stall, it is used on control endpoints. The device controller automatically clears it at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD must enable ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS] (both directions) as a pair. A single write to ENDPTCTRL*x* can ensure that both ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS] become 1 at the same instant.

NOTE

Any write to ENDPTCTRL*x* in Operational mode must preserve ENDPTCTRL*x*[TXT] and ENDPTCTRL*x*[RXT], which means you must perform a read-modify-write operation.

Table 474. Device controller stall response matrix

USB packet	Value of ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS]	Effect on ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS]	USB response
Setup packet that a noncontrol endpoint receives	N/A	None	STALL
In, Out, or Ping packet that a noncontrol endpoint receives	1	None	STALL
In, Out, or Ping packet that a noncontrol endpoint receives	0	None	ACK, NAK, or NYET
Setup packet that a control endpoint receives	N/A	Cleared	ACK
In, Out, or Ping packet that a control endpoint receives	1	None	STALL
In, Out, or Ping packet that a control endpoint receives	0	None	ACK, NAK, or NYET

51.3.6.2.3 Data toggle

Data toggle is a mechanism to maintain data coherency between the host and device for a given data pipe.

For more information about data toggle, see the USB 2.0 specification.

51.3.6.2.3.1 Data toggle reset

The DCD can reset ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR] and cause the data toggle sequence to reset in the device controller by writing 1 to ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR].

This is necessary only when configuring or initializing an endpoint, or returning from a STALL condition.

51.3.6.2.3.2 Data toggle inhibit

Writing 1 to ENDPTCTRL*x*[TXI] and ENDPTCTRL*x*[RXI] causes the device controller to ignore the data toggle pattern that is normally sent, and accepts all incoming data packets regardless of the data toggle state.

During normal operation, the device controller checks the DATA0 and DATA1 fields against the data toggle to determine whether the packet is valid. If the data PID does not match either ENDPTCTRL*x*[TXR] or ENDPTCTRL*x*[RXR] (that the device controller maintains for that endpoint), the data toggle is considered invalid. In this case, the device controller assumes that the packet is already received and discards the packet without reporting it to the DCD. To prevent the host controller from resending the same packet, the device controller responds to the error packet by acknowledging it with either an ACK or NYET response.

NOTE

This feature is for test purposes only; you must never use it during a normal device controller operation.

51.3.6.2.3.3 Priming transmit (TX) endpoints

Priming a transmit endpoint causes the device controller to fetch the device transfer descriptor (dTD) for the transaction that the dQH points to.

After the dTD is fetched, it is stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host, without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint, with the maximum number of endpoints configured at the device synthesis time.

After a priming request is complete, Endpoint Status (ENDPTSTAT) indicates an endpoint state of primed transmit. For this, the device controller can respond to an "In" request from the host and meet the stringent bus turnaround time of HS USB.

Because only the leading data is stored in the device controller FIFO, the device controller must begin filling in leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that the system memory bus can incur.

51.3.6.2.3.4 Priming receive (RX) endpoints

From the point of view of the DCD, priming receive endpoints is identical to priming transmit endpoints. Considering the device controller, the major difference in the operational model is that there is no data movement of the leading packet data because the data will be received from the host.

As part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the receive FIFO does not scale with the number of endpoints.

51.3.6.3 Operational model for packet transfers

The host initiates all transactions on the USB bus, and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 specification.

With USB 1.1 FS and LS rates, this turnaround time was significant and the USB 1.1 device controllers were architected in a way that enabled the device controller to access the main memory or interrupt a host protocol processor in order to respond to the USB 1.1 transaction. The architecture of the USB 2.0 device controller is different because the same methods do not meet the USB 2.0 HS turnaround time requirements from the increased clock rate.

A USB host sends requests to the device controller in an order that cannot be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable if the endpoint number and direction are considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then the host is expected to send "In" requests to that endpoint. This device controller is architected in a way that enables it to prepare packets for each endpoint or direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host-initiated transactions on the bus is referred to as priming the endpoint. This term is used throughout the chapter to describe the device controller operation so you can use priming to appropriately architect the DCD. Furthermore, the term flushing is used to describe the action of clearing a packet that is queued for execution.

51.3.6.3.1 Interrupt and bulk endpoint operational model

The behaviors of the device controller for interrupt and bulk endpoints are identical.

All valid "In" and "Out" transactions to bulk pipes handshake with a NAK unless the endpoint is primed. After the endpoint is primed, data delivery commences.

The device controller retires a dTD after the packets described in the transfer descriptor are completed. Each dTD describes "N" packets to be transferred according to the USB variable length transfer protocol. Equation 5, Table 475, and Table 476 describe how the device controller computes the number and length of packets that USB must send and receive. These vary according to the total number of bytes and maximum packet length. Also, Table 477 describes when TX-dTD and RX-dTD are considered complete.

With ZLT = 0, N = INT (number of bytes ÷ maximum packet length) + 1

With ZLT = 1, N = MAXINT (number of bytes ÷ maximum packet length)

N = MAXINT (number of bytes ÷ maximum packet length)

Equation 5. Formula for N

Table 475. Variable length transfer protocol example (ZLT = 0)

Bytes (dTD)	Maximum packet length (dQH)	N	P1	P2	P3
511	256	2	256	255	—
512	256	3	256	256	0
512	512	2	512	0	—

Table 476. Variable length transfer protocol example (ZLT = 1)

Bytes (dTD)	Maximum packet length (dQH)	N	P1	P2	P3
511	256	2	256	255	—
512	256	2	256	256	—
512	512	1	512	—	—

NOTE

You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.

Table 477. TX-dTD and RX-dTD completion status

TX-dTD is considered complete when:	RX-dTD is considered complete when:
All packets described in dTD are successfully transmitted. When this occurs, the total bytes in dTD become zero.	• All packets described in dTD are successfully received. When this occurs, the total bytes in dTD become zero. • A short packet (number of bytes < maximum packet length) is received. This is a successful transfer completion. DCD must check the total bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received. • A long packet is received: (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). This is an error condition. The device controller discards the remaining packet and sets the Data Buffer Error bit in the Status field of the dTD (see Device endpoint transfer descriptor (dTD)). In addition, the endpoint is flushed and the USBERR interrupt becomes active.

After the successful completion of the packet(s) that dTD describes, the Active bit in the Status field of the dTD is cleared and the next pointer is followed when the Terminate field in dTD is clear. When the Terminate field is set, the device controller flushes the endpoint or direction and ceases operations for that endpoint or direction.

After an unsuccessful completion of a packet (in the case of a long packet), the dQH points to the dTD in error. To recover from this error condition, the DCD must properly reinitialize the dQH by clearing the Active bit and updating the next dTD pointer before attempting to reprime the endpoint.

NOTE

The device controller automatically retires all packet level errors, such as a missing handshake or CRC error. An interaction with the DCD is not required for handling such errors.

51.3.6.3.1.1 Interrupt and bulk endpoint bus response matrix

The following table shows the device controller's response to packets on interrupt and bulk endpoint, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- NYET or ACK: If the transfer descriptor has packets remaining (according to the USB variable length protocol), the response is ACK. Otherwise, the response is NYET.

A system error (SYSERR) should not occur when the latency FIFOs are correctly sized and the DCD is responsive.

Table 478. Interrupt and bulk endpoint bus response matrix

Token type	Stall	Not primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	Bus error	N/A
Out	STALL	NAK	Receive + NYET or ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

51.3.6.3.2 Control endpoint operation model

51.3.6.3.2.1 Setup phase

All requests to a control endpoint begin with a setup phase, followed by an optional data phase and a required status phase. The device controller always accepts the setup phase unless the setup lockout is engaged.

The setup lockout engages so that future setup packets are ignored. Lockout of setup packets ensures that when software is reading the setup packet stored in the queue head, that data is not written at the same time because this can potentially cause an invalid setup packet.

You can disable the setup lockout mechanism only one time at initialization (by writing 1 to USBMODE[SLOM]), and a new tripwire type semaphore ensures that the setup packet payload is extracted from the queue head without being corrupted by an incoming setup packet. Setup lockout is not necessary when using the tripwire. This is the preferred behavior because ignoring repeated setup packets because of long software interrupt latency leads to a compliance issue.

After receiving an interrupt and inspecting Endpoint Setup Status (ENDPTSETUPSTAT) to determine that a setup packet is received on a particular pipe, perform the following procedure.

Table 479. Setup phase: associated procedure

Step	Purpose	Programming	Notes
1	Clear the corresponding bit.	Write 1 to ENDPTSETUPSTAT[ENDPTSETUPSTAT].	—
2	—	Write 1 to USBCMD[SUTW].	—
3	Duplicate contents of the Setup Buffer field (in dTD) into a local software byte array.	—	—
4	—	Read USBCMD[SUTW].	If the field is 1, continue to the next step; if the field is 0, go back to step 2.
5	—	Write 0 to USBCMD[SUTW].	—
6	Process the setup packet using a local software byte array copy, and execute the status and handshake phases.	—	After receiving a new setup packet, the status and handshake phases may still be pending from a previous control sequence. These must be flushed and deallocated before linking a new status or handshake dTD for the most recent setup packet.

51.3.6.3.2.2 Data phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must immediately verify (by reading Endpoint Setup Status (ENDPTSETUPSTAT)) that a new setup packet has not been received. A prime completes when the associated field in Endpoint Prime (ENDPTPRIME) is 0 and the associated field in Endpoint Status (ENDPTSTAT) is 1. If the associated field in Endpoint Prime (ENDPTPRIME) becomes 0 and the associated field in Endpoint Status (ENDPTSTAT) has not become 1, prime fails. This is only due to improper setup of the dQH, the dTD, or a setup arriving during the prime operation. If a new setup packet is indicated after the associated field in Endpoint Prime (ENDPTPRIME) becomes 0, then the transfer descriptor is freed and the DCD must reinterpret the setup packet.

If a setup arrives after the data stage is primed, the device controller automatically clears the prime status (Endpoint Status (ENDPTSTAT)) to enforce data coherency with the setup packet.

NOTE

- You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is performed in the same way as bulk packets.

51.3.6.3.2.3 Status phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal to 0) and prime the endpoint for the status phase.

The DCD must also perform the same checks associated with ENDPTSETUPSTAT[ENDPTSETUPSTAT], as described in the data phase.

NOTE

- You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is performed in the same way as bulk packets.

51.3.6.3.2.4 Control endpoint bus response matrix

The following table shows the device controller's response to packets on a control endpoint, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- NYET or ACK: If the transfer descriptor has packets remaining (according to the USB variable length protocol), the response is ACK. Otherwise, the response is NYET.
- System error (SYSERR) must never occur when the latency FIFOs are correctly sized and the DCD is responsive.

Table 480. Control endpoint bus response matrix

Token type	Endpoint state					Setup lockout
	Stall	Not primed	Primed	Underflow	Overflow	
Setup	ACK	ACK	ACK	N/A	SYSERR	—
In	STALL	NAK	Transmit	Bus error	N/A	N/A
Out	STALL	NAK	Receive + NYET or ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

51.3.6.3.3 Isochronous endpoint operational model

Isochronous endpoints are used for real-time scheduled delivery of data, and their operational model is significantly different from the host throttled bulk, interrupt, and control data pipes.

The device controller accomplishes real-time delivery by using the following techniques:

- Exact MULT packets per microframe are transmitted and received. MULT is a two-bit field in the dQH (see Device endpoint queue head (dQH)). The variable length packet protocol is not used for isochronous endpoints.
- NAK responses are not used. Instead, zero-length packets are sent in response to an "In" request for unprimed endpoints. For unprimed RX endpoints, the response to an "Out" transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next microframe. If the ISO-dTD is still active after that frame, then the ISO-dTD is considered ready until the DCD executes or cancels it.

An EHCI-compatible host controller uses the periodic frame list to schedule data exchanges to isochronous endpoints. The operational model for Device mode does not use such a data structure. Instead, the same dTD used for control, bulk, and interrupt endpoints is also used for isochronous endpoints. The difference between bulk and ISO endpoints is in the handling of the dTD:

- Priming an ISO endpoint is a delayed operation so that an endpoint is primed only after an SOF is received. After the DCD writes to the prime bit, the bit is cleared as usual to indicate to software that the device controller completed priming the dTD for transfer. Internal to the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific microframe.
- With isochronous endpoints, the transaction must fully complete within one microframe. After an ISO transaction is started in a microframe, it retires the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

  The Transaction Error bit in the Status field of the dTD indicates a fulfillment error condition. When a fulfillment error occurs on the frame after the transfer fails to complete, the device controller force retires the ISO-dTD and moves to the next ISO-dTD.

NOTE

Partially completed packets cause fulfillment errors. If no activity occurs on a primed ISO-dTD, the transaction stays primed indefinitely. This means it is up to you to discard transmit ISO-dTDs that pile up from a failure of the host or to move the data.

- The data level error handling for ISO packets is different. When a CRC error occurs on a received packet, the packet is not retried in the same way as bulk and control endpoints. Instead, the CRC error is noted by setting the Transaction Error bit in the Status field of the dTD, and the data is stored as usual for the application software to manage. The following table shows how a retired ISO packet is managed.

Table 481. Conditions that cause transmit and receive packets to retire

Conditions for transmit packet retirement (all conditions must be fulfilled)	Conditions for receive packet retirement (all conditions must be fulfilled)
• The MULT counter reaches zero. • A fulfillment error occurs. The Transaction Error bit is set in the Status field of the dTD: number of packets occurred > 0 AND number of packets occurred < MULT. **NOTE** For TX-ISO, the MULT counter can be loaded with a lesser value in the dTD MULTO field in hardware versions 2.3 and later. If MULTO is 0, the MULT counter is initialized to the multiplier in the queue head.	• The MULT counter reaches zero. • A non-MDATA PID is received. The exit criteria is only valid in hardware version 2.3 or later. Before this version, any PID sequence that did not exactly match the MULTO field was flagged as a transaction error because of a PID mismatch or fulfillment error. • An overflow error occurs. This happens when: — The packet received is greater than the maximum packet length (the Data Buffer Error bit is set in the Status field of the dTD). — The packet received exceeds the total bytes allocated in the dTD (the Data Buffer Error bit is set in the Status field of the dTD). • A fulfillment error occurs (the Transaction Error bit is set in the Status field of the dTD): number of packets occurred > 0 AND number of packets occurred < MULT. • A CRC error occurs (the Transaction Error bit is set in the Status field of the dTD).

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For a continuous microframe-to-microframe operation, the DCD must ensure that the dTD linked-list is ahead of the device controller by at least two microframes.

51.3.6.3.3.1 Isochronous pipe synchronization

When it is necessary to synchronize an isochronous data pipe to the host, the microframe number (specified by FRINDEX) can be used as a marker.

To cause a packet transfer to occur at a specific microframe number ("N"), the DCD must interrupt on SOF during frame N-1. If FRINDEX = N-1, the DCD must write to the prime bit. The device controller primes the isochronous endpoint in microframe N-1 so that the device controller is able to execute delivery during microframe N.

NOTE

Priming an endpoint towards the end of microframe N-1 does not guarantee delivery in microframe N. The delivery may actually occur in microframe N+1 if the device controller does not have enough time to complete the prime before the SOF for packet N is received.

51.3.6.3.3.2 Isochronous endpoint bus response matrix

The following table shows the device controller's response to packets on isochronous endpoints, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- Null packet refers to zero-length packet.

Table 482. Isochronous endpoint bus response matrix

Token type	Stall	Not primed	Primed	Underflow	Overflow
Setup	Stall	Stall	Stall	N/A	N/A
In	Null packet	Null packet	Transmit	Bus error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

51.3.6.4 Managing queue heads

The dQH points to the linked list of transfer tasks; the dTD depicts each of these tasks. An area of memory that ENDPTLISTADDR points to contains a group of all dQHs in a sequential list, as shown in Figure 176. The even elements in the list of dQHs are used for receive endpoints (Out and Setup) and the odd elements are used for transmit endpoints (In and Interrupt). Device transfer descriptors are linked, head to tail, starting at the queue head and ending at the Terminate bit of the dTD. After the dTD retires, it is no longer a part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers do not exist within the queue head after the dTD is retired (see Software link pointers).

In addition to the current and next pointers, and the dTD overlay examined in Operational model for packet transfers, the dQH also contains the following parameters for the associated endpoint: Multiplier (MULT), Maximum Packet Length, and Interrupt On Setup (IOS). The next section discusses the complete initialization of the dQH, along with these fields.

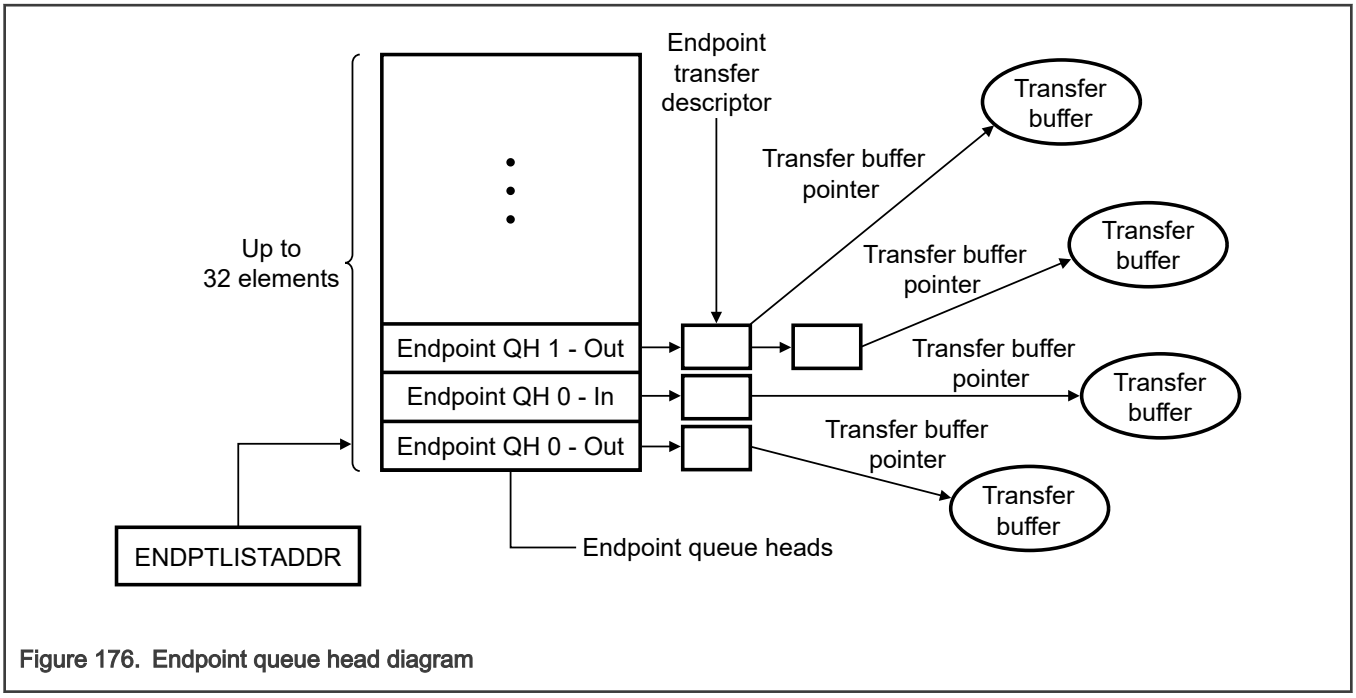


Figure 176. Endpoint queue head diagram

51.3.6.4.1 Queue head initialization

You must initialize a device queue head for each active endpoint. Perform the following procedure to initialize it.

Table 483. Queue head initialization

Step	Programming	Notes
1	Write to the wMaxPacketSize field as either the USB chapter 9 or application-specific protocol requires.	—
2	Write 0 to the MULT field in dQH for control, bulk, and interrupt endpoints.	Set the multiplier to 1, 2, or 3, for ISO endpoints, as the required bandwidth and in conjunction with the USB chapter 9 protocol. In FS mode, the MULT field in dQH can only be 1 for ISO endpoints.
3	Write 1 to the next Terminate field in dTD.	—
4	Write 0 to the Active bit in the Status field of dTD.	—
5	Write 0 to the Halt bit in the Status field of dTD.	—

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTDs.

51.3.6.4.2 Operational model for setup transfers

As discussed in Control endpoint operation model, the DCD gives special treatment to setup transfer. It does not use a dTD, but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

After receiving a notification of the setup packet, the DCD must manage the setup transfer as provided in the following procedure.

Table 484. Setup transfers: associated procedure

Step	Purpose	Programming	Notes
1	Copy setup buffer contents from dQH–RX to software buffer.	—	—
2	Acknowledge setup backup.	Write 1 to the corresponding bit in ENDPTSETUPSTAT[ENDPTSETUPSTAT].	• The acknowledgement must occur before continuing to process the setup packet. • After the acknowledgement has occurred, the DCD must not attempt to access the setup buffer in the dQH–RX. Only the local software copy must be examined.
3	Check for pending data or status dTDs from previous control transfers and flush, if any exist.	See Flushing and de-priming an endpoint.	—
4	—	Decode the setup packet, and prepare data phase (optional) and status phase transfers as required by either the USB chapter 9 or application-specific protocol.	—

NOTE

The device controller can receive setup packets before the completion of previous control transfers. Existing control packets in progress must be flushed and the new control packet must be completed.

51.3.6.5 Managing transfers with transfer descriptors

51.3.6.5.1 Software link pointers

The DCD software must maintain head and tail pointers to the linked list of dTDs for each respective queue head.

This is necessary because the dQH only maintains pointers to the currently working dTD and the next dTD to be executed. The operations described in the next section for managing the dTD assume that the DCD can use the head and tail of the dTD linked list. The following figure shows the software link pointers.

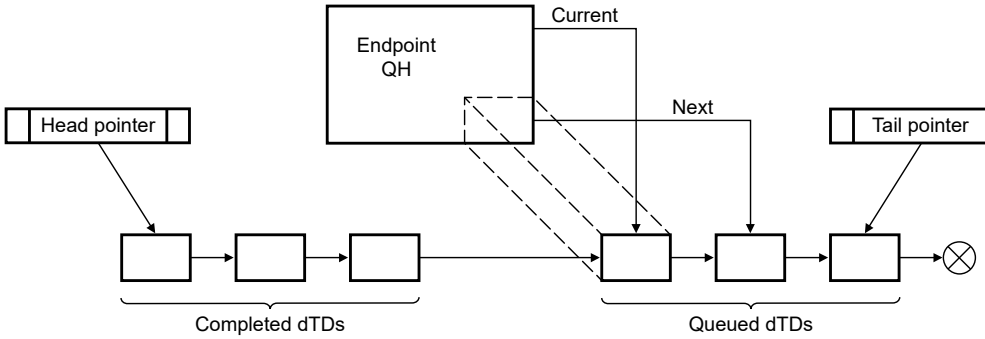


Figure 177. Software link pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the head and tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

51.3.6.5.2 Building a transfer descriptor

Before executing a transfer from the linked list, you must build a dTD to describe the transfer.

As a prerequisite, allocate 8-DWORD dTD block of memory aligned to 8-DWORD boundaries. For example, bit address 4:0 is equal to 00000. Then perform the following procedure to build a dTD.

Table 485. Building a transfer descriptor: associated procedure

Step	Purpose	Programming
1	—	Write 0 to the first seven DWords.
2	—	Write 1 to the Terminate field in dTD.
3	—	Write the transfer size to the Total Bytes field in dQH.
4	—	Write 1 to the Interrupt On Complete (IOC) field in dTD, if desired.
5	Initialize the Status field.	Write 1 to the Active bit of the Status field and write 0 to all remaining Status bits.
6	Point to the start of the data buffer.	Fill in buffer pointer page 0 and the current offset.
7	—	Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointers.

51.3.6.5.3 Executing a transfer descriptor

To safely add a dTD, the DCD must perform the procedures listed in Table 486, based on the case. These procedures manage the event in which the device controller reaches the end of the dTD list at the same time a new dTD is added to the end of the list.

Determine whether the link list is empty by checking the DCD driver. This evaluates whether the pipe is empty (the internal representation of the linked list must indicate whether any packets are outstanding).

Table 486. Executing a transfer descriptor

Step	Case 1: If the link list is empty	Step	Case 2: If the link list is not empty
1	Write to the dQH next pointer and write 0 to the dQH Terminate bit as a single DWORD operation.	1	Add dTD to the end of linked list.
2	Clear the active and halt bits in dQH (in case they are set from a previous error).	2	Read the appropriate field in ENDPTPRIME. If it is 1, it is done; otherwise, continue to read the field.
3	Prime the endpoint by writing 1 to the appropriate field in ENDPTPRIME. You must check that the dTD and dQH content in external memory is updated before writing 1 to the field in ENDPTPRIME.	3	Write 1 to USBCMD[ATDTW].
—	—	4	Read the appropriate status field in ENDPTSTAT (store in a temporary variable for later use).
—	—	5	Read USBCMD[ATDTW]: • If the field is 0, go back to step 3. • If the field is 1, move forward to step 6.
—	—	6	Write 0 to USBCMD[ATDTW]: • If the status field read at step 4 is 1, the procedure is complete. • If the status field read at step 4 is 0, then go to step 1 of case 1.

51.3.6.5.4 Transfer completion

After you initialize a dTD and prime the associated endpoint, the device controller executes the transfer on the host-initiated request. The DCD is notified with a USB interrupt if the Interrupt On Complete (IOC) bit in the dTD is set. Alternately, the DCD can poll ENDPTCOMPLETE to find when the dTD was executed. After a dTD is executed, the DCD can also check the associated status bits to determine execution success or failure.

NOTE

Multiple dTDs can be completed in a single endpoint-complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that are complete (the Active bit in the dTD Status field is cleared).

By reading the status fields of the completed dTDs, the DCD can determine whether the transfers completed successfully. The following combination of status bits determine the transfer completion success:

- Active = 0
- Halted = 0
- Transaction Error = 0
- Data Buffer Error = 0

If any combination other than the aforementioned one exists, the DCD must take proper action. Device error matrix indicates the transfer failure mechanisms.

In addition to checking the status bit, the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the total bytes transferred is decremented by the actual bytes transferred. For transmit packets, a packet is considered complete only after the actual byte count is zero. For receive packets, the host can send fewer bytes in the transfer according to the USB variable length packet protocol.

51.3.6.5.5 Flushing and de-priming an endpoint

The DCD must flush to de-prime another endpoint on a USB device reset or during a broken control transfer.

Some application-specific requirements may exist to stop transfers in progress. The DCD can perform the following procedure to stop a transfer in progress:

1. Write 1 to the corresponding field(s) in ENDPTFLUSH.
2. Wait until all fields in ENDPTFLUSH become 0.

   NOTE

   This operation may take quite a long time, depending on the USB bus activity. You must not have this wait loop within an interrupt service routine.

3. Read ENDPTSTAT to ensure that the corresponding fields are now 0 for all endpoints commanded to be flushed. The following is the reason for a flush failure, if the corresponding bits are still 1 after completion of step 2:

   In very rare cases, a packet is already in progress to a particular endpoint when the previous endpoint commands the flush operation by writing to ENDPTFLUSH. A safeguard is in place to refuse the flush, to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush by repeating steps 1–3 until each endpoint is successfully flushed.

51.3.6.5.6 Device error matrix

Table 487 summarizes packet errors that the device controller does not manage automatically.

Table 487. Device error matrix

Error	Direction	Packet type	Data buffer error bit	Transaction error bit
Overflow	RX	Any	1	0
ISO packet error	RX	ISO	0	1
ISO fulfillment error	Both	ISO	0	1

The device controller manages all errors on bulk, control, and interrupt endpoints, except for a data buffer overflow. However, for ISO endpoints, error packets are not retried and errors are tagged as indicated in Table 488.

Table 488. Error descriptions

Error	Description
Overflow	Indicates that the number of bytes received exceeded the maximum packet size or total buffer length. This error also sets the Halt bit in the dQH, and if there are dTDs remaining in the linked list for the endpoint, then those are not executed.
ISO packet error	Indicates a CRC error on the received ISO packet. Contents are not guaranteed to be correct.

Table continues on the next page...

Table 488. Error descriptions (continued)

Error	Description
ISO fulfillment error	Indicates that the host failed to complete the number of packets, defined in the dQH MULT field, within the given microframe. For scheduled data delivery, the DCD may need to readjust the data queue because a fulfillment error causes the device controller to cease data transfers on the pipe for one microframe. During the dead microframe, the device controller reports an error on the pipe and primes for the following frame.

51.3.7 Modes of operation

USB has two main modes of operation.

Table 489. Modes of operation

Mode	Description
Normal	USB can operate in Host mode and Device (Peripheral) mode.
Low-Power	USB supports Suspend mode and Sleep mode to reduce power consumption. It has an associated power control subblock that is clocked on a low-power, low-frequency clock. When the USB bus is idle, the transceiver can be placed in Low-Power mode, after which you can stop the USB clocks. This low-power clock must remain active because it is needed for wake-up detection. **NOTE** See the chip-specific USB information for the low-power clock in Suspend mode and Sleep mode. As the USB 2.0 specification describes, the device can go into the Suspend state after it sees a constant Idle state on the upstream-facing port (UFP). USB enters Suspend mode after 3 ms of inactivity on the port when it is in Device Operation mode. The host controller does not suspend automatically; you can place it in Suspend mode. Either the local Arm platform, the remote USB host, or a peripheral can initiate a wake-up sequence to resume USB communication. For more information on Suspend and Resume states, see Power control.

51.3.7.1 Power control

USB supports suspend and wake-up functionalities.

The power control block allows you to place the transceiver in USB Low-Power mode when the USB bus is in the Idle state, and supports local and remote wake-up to bring the transceiver out of the USB Low-Power mode, when needed. Additionally, the power control block can wake up the Arm platform by generating an interrupt.

51.3.7.1.1 Entering a low-power suspend (L2) mode

In Device Operation mode, after the host drive is in the Idle state for 3 ms, an SLI interrupt is issued (USBSTS[SLI]). After SLI is received, perform the steps listed in the last column of the following table.

Low-Power Suspend mode is always entered under the control of the driver software by writing 1 to the appropriate field in PORTSC1. After the controller is suspended, the clocks to the USB controller can be stopped.

For entering a low-power suspend (L2) mode, the device can be in Low-Power mode.

Table 490. Entering a low-power suspend (L2) mode

Step	When in Host Operation mode:	Step	When in Device Operation mode (after SLI is received):
1	Write 0 to USBCMD[ASE] and USBCMD[PSE], and wait until USBSTS[AS] and USBSTS[PS] become 0.	1	Write 1 to PORTSC1[PHCD].
2	Write 1 to PORTSC1[SUSP].	2	Select the wake-up sources.
3	For an HS operation, wait for 4 ms.	3	Wait for 3 µs.
4	Write 1 to PORTSC1[PHCD].	—	—
5	Select the wake-up sources.	—	—
6	Wait for 3 µs.	—	—

51.3.7.1.2 Entering a low-power sleep (L1) mode

For entering a low-power sleep (L1) mode, the device can be in Low-Power mode but the low-power clock must be available for wake-up from L1.

Table 491. Entering a low-power sleep (L1) mode

Step	When in Host Operation mode:	Step	When in Device Operation mode:
1	Write 0 to USBCMD[ASE] and USBCMD[PSE], and wait until USBSTS[AS] and USBSTS[PS] become 0.	1	Write 1 to LPM_CSR0[LPM_EN].
2	Write 1 to LPM_CSR0[LPM_EN].	2	Write 1 to the interrupt enable fields, such as USBINTR[PCE], USBINTR[SLE], USBINTR[LPM_L1_ENTRYIE], and USBINTR[LPM_L1_EXITIE].
3	Write 1 to LPM_CSR0[LPM_ERRATA_EN] to support *Errata for USB 2.0 ECN: Link Power Management (LPM) 7/2007*.	3	Write 1 to LPM_CSR0[LPM_ERRATA_EN] to support *Errata for USB 2.0 ECN: Link Power Management (LPM) 7/2007*.
4	Configure LPM_CSR2[LPM_HST_DEVADD], LPM_CSR2[LPM_HST_BESL], and LPM_CSR2[LPM_HST_RWKEN]. These fields are sent via the LPM token to the device.	4	Configure LPM_CSR1[LPM_DEV_BESLTHRES] and LPM_CSR1[LPM_DEV_RES].
5	Write 1 to the interrupt enable fields, such as USBINTR[PCE], USBINTR[LPM_L1_ENTRYIE], and USBINTR[LPM_L1_EXITIE].	5	After the host sends the LPM token, the device automatically replies with field values ACK, NYET, or STALL, according to the current situation. The received LPM token content is recorded in LPM_CSR1[LPM_DEV_BESLRCVD], LPM_CSR1[LPM_DEV_LNKSTRCVD], or LPM_CSR1[LPM_DEV_RWKENRCVD]. The device's response is recorded in LPM_CSR1[LPM_DEV_RSPSTS] too.
6	Write 1 to LPM_CSR2[LPM_HST_SEND] to send the LPM token to the device. This field automatically becomes 0 after the LPM transaction is completed.	6	USBSTS[LPM_DEV_RCVDI] interrupt can be issued.

Table continues on the next page...

Table 491. Entering a low-power sleep (L1) mode (continued)

Step	When in Host Operation mode:	Step	When in Device Operation mode:
	USBSTS[LPM_HST_COMPI] becomes 1 after receiving the device's response to the LPM token. The response is recorded in LPM_CSR2[LPM_HST_STSRCVD].		
7	PORTSC1[SUSP] automatically becomes 1 after the ACK response is received and LPM_CSR0[LPM_L1_ACTIVE] is 1. The host is in the L1 state.	7	After the TL1TokenRetry time, the device starts transitioning to L1. After the TL1Residency time, the device is in L1.
8	PORTSC1[PHCD] automatically becomes 1 after (TL1TokenRetry + TL1Residency) . The host is in Low-Power L1 mode.	8	PORTSC1[PHCD] automatically becomes 1 and the device is in Low-Power Sleep mode.
9	USBSTS[LPM_L1_ENTRYI] interrupt can be issued.	9	USBSTS[SLI] or USBSTS[LPM_L1_ENTRYI] interrupts can be issued.
10	Select the wake up sources.	10	Select the wake-up sources.
11	Wait for 3 µs.	11	Wait for 3 µs.

51.3.7.1.3 Wake-up events

The power control block monitors the USB bus when the USB core is in USB Suspend or Sleep state.

Depending on whether the controller is in Host or Device mode, a number of wake-up conditions are monitored. If the related wake-up interrupt enable field is 1, an asynchronous interrupt is generated to the Arm platform upon detection of a wake-up condition.

51.3.7.1.3.1 Host mode events

The host controller wakes up after any of the following events:

- Remote wake-up request: A peripheral can request the host to reactivate the bus by driving wake-up signaling on the DM or DP lines. The power control block sends a wake-up request to the USB core when a J-K transition on the DM or DP line is detected.
- Wake-up on overcurrent: If a wake-up on overcurrent is enabled (PORTSC1[WKOC] = 1), the power control block sends a wake-up request to the USB core when an overcurrent event is detected.

NOTE

See the chip-specific USB information to know whether the controller supports wake-up on overcurrent.

- Wake-up on disconnect: If wake-up on disconnect is enabled (PORTSC1[WKDC] = 1), the power control block sends a wake-up request to the USB core when a disconnection event is detected (a J-SE0 or K-SE0 transition on the DM or DP line).
- Wake-up on connect: If a wake-up on connect is enabled (PORTSC1[WKCN] = 1), the power control block sends a wake-up request to the USB core when a connection event is detected (an SE0-J or SE0-K transition on the DM or DP line).

51.3.8 Clocking

Table 492. USB clock sources

Clock	Description
AHB clock	AHB bus clock
UTMI clock	UTMI clock from USBPHY
Low-power, low-speed clock	Used for wake up from Low-Power mode. See the chip-specific USB information for low-power clock information.

See the "Clock Controller (CLKCTL)" chapter for clock setting, configuration, and gating information.

51.3.9 Interrupts

51.3.9.1 USB core interrupts

The USB core uses a dedicated vector in the interrupt table.

With the exception of wake-up interrupts, the USB core controls all interrupt sources. See Interrupt Enable (USBINTR) for more information.

51.3.9.2 USB wake-up interrupts

The USB core has an associated wake-up interrupt. These interrupts are generated outside the USB core, but use the same vector as the corresponding USB core interrupt.

The power control blocks generate these interrupts, which run on a low-power, low-speed clock. The wake-up interrupt is designed to work even when the USB and Arm platform clocks are disabled so that a wake-up condition on the USB bus can reactivate the Arm platform clocks.

Because the wake-up interrupt is generated and cleared on a low-power, low-speed clock, this interrupt request responds very slowly to clear actions. For this reason, you must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously because it is clocked by the Arm platform clock. You must wait for at least three low-power clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is used only during USB low-power modes, it is sufficient to enable the wake-up interrupt just prior to entering the USB Suspend mode.

51.3.9.3 Interrupts: host operational model

For detailed information about interrupt capability that the EHCI host controller hardware provides, see the "Interrupts" section in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

51.3.9.4 Servicing interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency, and error operations, and consider the order of interrupts accordingly.

51.3.9.4.1 High-frequency interrupts

You must manage high-frequency interrupts in the order shown in the following table. The most important of these is listed first because the DCD must acknowledge a setup buffer as promptly as possible.

Table 493. High-frequency interrupt events

Execution order	Interrupt	Action
1a	USB interrupt (see ENDPTSETUPSTAT)	Copy the contents of the setup buffer and acknowledge the setup packet (as indicated in Figure 176). Process the setup packet according to USB 2.0 chapter 9 or the application-specific protocol.
1b	USB interrupt[1] (see ENDPTCOMPLETE)	Handle the completion of dTD as indicated in Figure 176.
2	SOF interrupt	Perform the action that the application considers necessary. This interrupt may not be used in all applications.

1. It is common for multiple interrupts to stack up on any call to and during the interrupt service routine.

51.3.9.4.2 Low-frequency interrupts

Low-frequency interrupts do not occur often compared to high-frequency interrupts. You can handle low-frequency interrupts in any order.

Table 494. Low-frequency interrupt events

Interrupt	Action
Port change	Change software state information.
Sleep enable (suspend)	Change software state information. Perform low-power handling as necessary.
Reset received	Change software state information. Abort pending transfers.

51.3.9.4.3 Error interrupts

Error interrupts are infrequent. You must place them last in the interrupt service routine.

Table 495. Error interrupt events

Interrupt	Action
USB error interrupt	This error is redundant because it combines a USB interrupt and an error status in the dTD. The DCD handles packet-level errors more appropriately by checking the dTD status field upon receipt of the USB interrupt (by using ENDPTCOMPLETE).
System error	Unrecoverable error that requires immediate reset of core; you must free up transfer buffers in progress and restart the DCD.

51.4 External signals

Table 496. USB external signals

Signal	Description	Direction
OC	External input for VBUS overcurrent detection from GPIO	I
PWR	Used to control PMIC to supply VBUS voltage	O

See the chip-specific USB information for the external signals that this chip supports.

51.5 Initialization

This section provides the information required to properly initialize USB.

51.5.1 Host controller initialization

For detailed information, see the "Host Controller Initialization" section in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

51.5.2 Device controller initialization

After hardware reset, the device is disabled until you write 1 to USBCMD[RS]. In the Disabled state, the pull-up on USB D+ is not active, and that prevents an attach event from occurring. The minimum requirement is to have the queue heads setup for endpoint 0 before the device attach occurs.

Soon after you enable the device, a USB reset occurs followed by a setup packet arriving at endpoint 0. A queue head must be prepared so that the device controller can store the incoming setup packet.

51.5.2.1 Prerequisite for initializing a device

You must reset the device controller for transitioning from Host mode to Device mode, before modifying USBMODE.

You must also prime endpoint 0 initially because the first packet received is always a setup packet. The contents of the first setup packet require a response in accordance with the USB device framework (chapter 9) command set.

51.5.2.2 Procedure to initialize a device

Table 497. Device controller initialization: associated procedure

Step	Purpose	Programming	Notes
1	Specify the controller mode.	Write 10b to USBMODE[CM].	—
2	Allocate and initialize device queue heads in the system memory. At a minimum, you must initialize device queue heads 0 TX and 0 RX.	—	You must initialize all device queue heads for control endpoints before the endpoint is enabled, and noncontrol device queue heads before you can use the endpoint. For information about device queue heads, see Device data structures.
3	Specify the endpoint list pointer.	Configure ENDPTLISTADDR.	—
4	Enable the microprocessor interrupt associated with the USB core.	NXP recommends that you enable all device interrupts including USBSTS[UI], USBSTS[UEI], USBSTS[PCI], USBSTS[URI], and USBSTS[SLI].	For a list of available interrupts, see Interrupt Enable (USBINTR) and USB Status (USBSTS).
5	Specify whether the controller is executing the operation.	Write 1 to USBCMD[RS].	After this is done and the device is connected to a host, the host downstream port issues a bus reset. The DCD must monitor the reset event and adjust the software state as described in Bus reset.

NOTE

Endpoint 0 is designed as a control endpoint only; you do not need to configure it using Endpoint Control 0 (ENDPTCTRL0).

51.6 Application information

51.6.1 eUSBPHY configuration

Perform the following procedure to configure eUSBPHY. You must complete this sequence first to enable the controller.

Table 498. eUSBPHY configuration: associated procedure

Step	Purpose	Programming	Notes
0	Configure PHY analog.	• Write 3'b011 to EUSB_CTRL1[EUSB_SQ]. • Write 2'b01 to EUSB_CTRL1[EUSB_FSTX_OPT].	—
1	Select work modes.	Use EUSB_CTRL0[NATIVE_MODE].	—
2	Select the HS receiver termination mode.	Use EUSB_CTRL0[UN_TERMINATED_MODE].	This field is only used for the host HS native mode.
3	Enable the eUSBPHY PLL reference clock and wait until it is stable.	—	—
4	Enable the CM.RAP feature, if needed.	Write 1 to EUSB_RAP[CM_RAP_INIT_EN].	—
5	Enable eUSBPHY.	Write 1 to EUSB_CTRL0[PONRST].	—
6	Confirm that the UTMI clock from the USB PHY is valid.	Poll CTRL2[UTMI_CLK_VLD] until it becomes 1.	—

51.6.2 eUSB CM.RAP operation

The following is the control sequence for the register access protocol (CM.RAP). Perform the following procedure to control CM.RAP.

Table 499. Controlling the CM.RAP feature

Step	Purpose	Programming	Notes
1	Enable the CM.RAP feature.	Write 1 to EUSB_RAP[CM_RAP_INIT_EN].	—
2	Confirm that the internal state is ready for CM.RAP transmission.	Poll EUSB_RAP[CM_RAP_INIT_READY] until it returns 1.	—
3	Specify the CM.RAP command type.	Write to EUSB_RAP[CM_RAP_OP], EUSB_RAP[CM_RAP_ADDR], and EUSB_RAP[CM_RAP_WRDATA].	Table 500 shows the details of the CM.RAP command and Table 501 shows the CM.RAP address and data format.

Table continues on the next page...

Table 499. Controlling the CM.RAP feature (continued)

Step	Purpose	Programming	Notes
4	Send the command and communicate with CM.	Write 1 to EUSB_RAP[CM_RAP_START].	—
5	Confirm that the CM.RAP command completed successfully.	Poll EUSB_RAP[CM_RAP_DONE] or EUSB_RAP[CM_RAP_FAIL] until they return 1 and 0, respectively.	—
6	Retrieve the data.	Read EUSB_RAP[CM_RAP_RDDATA] (read command only).	—
7	Complete the command execution.	Write 0 to EUSB_RAP[CM_RAP_START] after EUSB_RAP[CM_RAP_DONE] or EUSB_RAP[CM_RAP_FAIL] becomes 1.	—
8	Perform the next CM.RAP command operation, until all required commands are executed.	Repeat steps 3 to 7.	—
9	Disable the CM.RAP feature.	Write 0 to EUSB_RAP[CM_RAP_INIT_EN].	—

Table 500. CM.RAP command description

CMD	Operation	Description
00b	Write	Data is written to the register address.
01b	Read	Data is read from the register address.
10b	Clear	Active-high, bitwise clear with the data on the register address.
11b	Set	Bit-wise OR with the data on the register address.

Table 501. CM.RAP format of address and data

Address						Data							
a0	a1	a2	a3	a4	a5	d0	d1	d2	d3	d4	d5	d6	d7

The following is an example of the CM.RAP write command. EUSB_RAP[CM_RAP_ADDR] and EUSB_RAP[CM_RAP_WRDATA] are randomly used in this example. For a read command, you can access EUSB_RAP[CM_RAP_RDDATA] after EUSB_RAP[CM_RAP_DONE] becomes 1. Perform the following procedure to use the CM.RAP command.

Table 502. Using the CM.RAP command

Step	Purpose	Programming
1	Enable the CM.RAP feature.	Write 1 to EUSB_RAP[CM_RAP_INIT_EN].
2	Confirm that the internal state is ready for CM.RAP transmission.	Poll EUSB_RAP[CM_RAP_INIT_READY] until it returns 1.

Table continues on the next page...

Table 502. Using the CM.RAP command (continued)

Step	Purpose	Programming
3	Configure the write command content.	1. Write 00b to EUSB_RAP[CM_RAP_OP]. 2. Write 6'd30 to EUSB_RAP[CM_RAP_ADDR]. 3. Write 8'd113 to EUSB_RAP[CM_RAP_WRDATA].
4	Enable CM.RAP start.	Write 1 to EUSB_RAP[CM_RAP_START].
5	Confirm that the CM.RAP command completed successfully.	Poll EUSB_RAP[CM_RAP_DONE] and EUSB_RAP[CM_RAP_FAIL] until they return 1 and 0, respectively.
6	Complete the command execution.	Write 0 to EUSB_RAP[CM_RAP_START].
7	Disable CM.RAP.	Write 0 to EUSB_RAP[CM_RAP_INIT_EN].

51.7 USB noncore memory map and register definitions

51.7.1 USBNC register descriptions

USB supports two types of registers: core and noncore. The USB core registers are used to control USB core functions and are more independent of USB features. The USB controller core has its own core registers. The USB noncore registers are additional to USB core registers and are more dependent on USB features. The USB noncore registers support the 32-bit, 16-bit, and 8-bit register accesses.

This section describes only the USB noncore registers. For detailed descriptions of the USB core registers, see Register interface.

NOTE

For reserved bits, preserve the value when writing (read its reset value, then write this value back).

51.7.1.1 USBNC memory map

USBNC0 base address: 4041_8200h

USBNC1 base address: 4041_9200h

Offset	Register	Width (In bits)	Access	Reset value
0h	USB Control 1 (CTRL1)	32	RW	3000_1000h
4h	USB Control 2 (CTRL2)	32	RW	DF00_0000h
2Ch	HSIC DLL Configuration 4 (HSIC_DLL_CFG4)	32	RW	0000_0040h
A0h	USB LPM Control and Status 0 (LPM_CSR0)	32	RW	0000_0008h
A4h	USB LPM Control and Status 1 (LPM_CSR1)	32	RW	0000_0000h
A8h	USB LPM Control and Status 2 (LPM_CSR2)	32	RW	0000_0000h
100h	eUSB Control 0 (EUSB_CTRL0)	32	RW	8000_0094h
104h	eUSB Control 1 (EUSB_CTRL1)	32	RW	0000_A000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
10Ch	eUSB Control 3 (EUSB_CTRL3)	32	RW	0000_0007h
148h	eUSB RAP Control and Status (EUSB_RAP)	32	RW	0000_0000h

51.7.1.2 USB Control 1 (CTRL1)

Offset

Register	Offset
CTRL1	0h

Function

Controls the integration-specific features of the USB module. These features are not directly related to USB functionality, but they control special features, interfacing on the USB ports, and power control and wake-up functionality.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WIR	Reserved	WKUP_DP...	REMOTE_...	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WKUP_VB...	Reserved
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WKUP_SW	WKUP_SW...	Reserved	Reserved	Reserved	WIE	PWR_POL	OVER_CU...	OVER_CU...	Reserved						Reserved
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 WIR	Wake-Up Interrupt Request Indicates whether a wake-up interrupt request is received. This field becomes 0 by disabling the wake-up interrupt (writing 0 to CTRL1[WIE]). 0b - Not received 1b - Received
30	This bit must retain its reset value.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>29
WKUP_DPDM_EN</td><td>Wake-Up After DP or DM Change Enable
Enables wake-up after DP or DM change. For Device mode only.
NOTE
The descriptions of the field settings vary by module instance.
<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>USBNC0</td><td>0b - Disable only when VBUS is invalid
1b - Enable (default)</td></tr><tr><td>USBNC1</td><td>0b - Disable
1b - Enable (default)</td></tr></table></td></tr>
<tr><td>28
REMOTE_WAKEUP_EN</td><td>Remote Wake-Up Enable
Enables remote wake-up. For Host mode only.
0b - Disables
1b - Enables</td></tr>
<tr><td>27
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>26
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>25
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>24
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>23
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>22
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>21
—</td><td>This bit must retain its reset value.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>20
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>19
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>18
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>17
WKUP_VBUS_EN</td><td>Wake-Up After VBUS Change Enable
Enables wake-up after VBUS change.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>USBNC0</td><td>CTRL1</td><td>—</td></tr><tr><td>USBNC1</td><td>—</td><td>CTRL1</td></tr></table>
0b - Disables
1b - Enables</td></tr>
<tr><td>16
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>15
WKUP_SW</td><td>Software Wake-Up
Forces software wake-up.
0b - Inactive
1b - Force wake-up</td></tr>
<tr><td>14
WKUP_SW_EN</td><td>Software Wake-Up Enable
Enables software wake-up.
0b - Disables
1b - Enables</td></tr>
<tr><td>13
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>12</td><td>This bit must retain its reset value.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>11
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>10
WIE</td><td>Wake-Up Interrupt Enable
Enables the wake-up interrupt. Disabling the interrupt also clears CTRL1[WIR]. Prior to entering Suspend mode, you must turn off the wake-up interrupt enable after receiving a wake-up interrupt and turn it on again.
0b - Disables
1b - Enables</td></tr>
<tr><td>9
PWR_POL</td><td>Power Polarity
Specifies the power polarity of the PMIC power pin.
0b - Active low
1b - Active high</td></tr>
<tr><td>8
OVER_CUR_POL</td><td>Overcurrent Polarity
Specifies the polarity of the port overcurrent event, representing an overcurrent condition.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>USBNC0</td><td>CTRL1</td><td>—</td></tr><tr><td>USBNC1</td><td>—</td><td>CTRL1</td></tr></table>
0b - Active high
1b - Active low</td></tr>
<tr><td>7
OVER_CUR_DIS</td><td>Overcurrent Disable
Disables overcurrent detection.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>USBNC0</td><td>CTRL1</td><td>—</td></tr></table></td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>USBNC1</td><td>—</td><td>CTRL1</td></tr></table> 0b - Enables 1b - Disables
6-1 —	This bit must retain its reset value.
0 —	This bit must retain its reset value.

51.7.1.3 USB Control 2 (CTRL2)

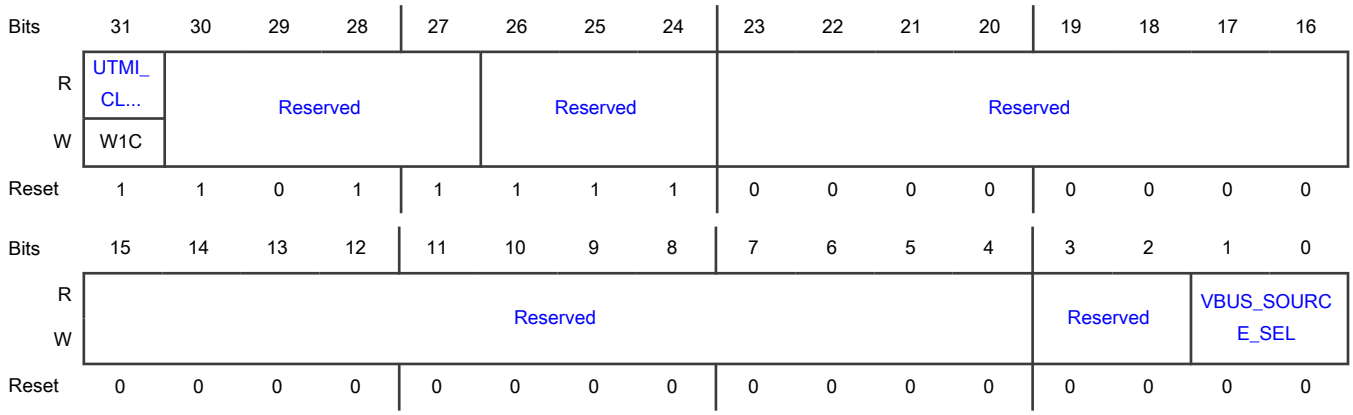
Offset

Register	Offset
CTRL2	4h

Function

Controls the integration-specific features of the USB module. These features are not directly related to USB functionality, but they control special features, interfacing on the USB ports, and power control and wake-up functionality.

Diagram



Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31
UTMI_CLK_VLD</td><td>UTMI Clock Valid Flag
Indicates whether the UTMI clock from the USB PHY is valid.
NOTE
This field behaves differently for register reads and writes.
When reading
0b - Not valid
1b - Valid
When writing
0b - No effect
1b - Clear the flag</td></tr>
<tr><td>30-27
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>26-24
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>23-4
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>3-2
—</td><td>This bit must retain its reset value.</td></tr>
<tr><td>1-0
VBUS_SOURCE_SEL</td><td>VBUS Source Select
Specifies VBUS source select when detecting VBUS wake-up event; it is for the UTMI PHY only.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>USBNC0</td><td>CTRL2</td><td>—</td></tr><tr><td>USBNC1</td><td>—</td><td>CTRL2</td></tr></table>00b - vbus_valid
01b - sess_valid
10b - sess_valid
11b - sess_valid</td></tr>
</table>

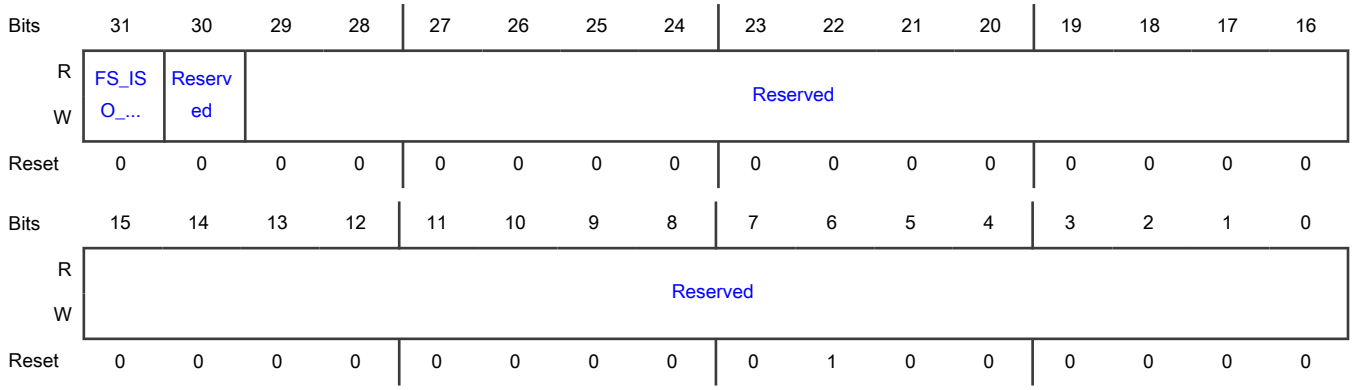
51.7.1.4 HSIC DLL Configuration 4 (HSIC_DLL_CFG4)

Offset

Register	Offset
HSIC_DLL_CFG4	2Ch

Function
Holds the internal control register bits of the PHY.

Diagram



Fields

Field	Function
31 FS_ISO_B2B_FIXEN	FS Isochronous Back-to-Back Transfer Enable Enables two connected FS isochronous transfers when the first length is longer than about 280 bytes in case the USB controller directly connects to an FS device. Such transfers are not supported by default. **NOTE** This bit must not be set in other cases, for example, when the FS device is connected to the USB controller through a hub. 0b - Disables 1b - Enables
30 —	This field must retain its reset value.
29-0 —	This field must retain its reset value.

51.7.1.5 USB LPM Control and Status 0 (LPM_CSR0)

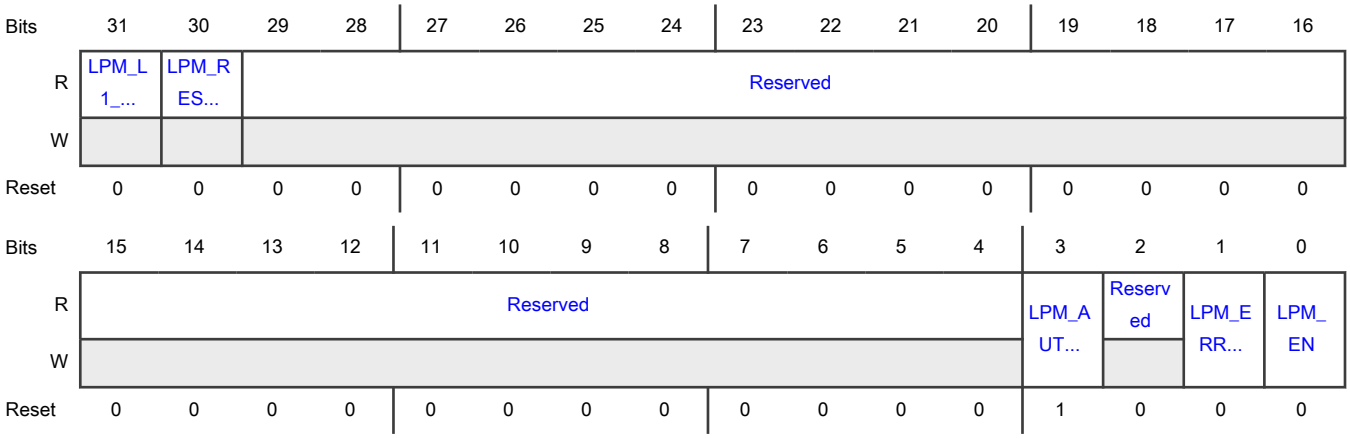
Offset

Register	Offset
LPM_CSR0	A0h

Function

Includes the common control and status fields for the host or device controller's LPM mode.

Diagram



Fields

Field	Function
31 LPM_L1_ACTIVE	LPM Active Indicates whether L1 is active. This flag means the device or host is in the L1 state: • For Device mode, this flag is set after the device sends an ACK response to the host. • For Host mode, this flag is set when the host receives an ACK response for the LPM extension transaction. 0b - Inactive 1b - Active
30 LPM_RESUME_OK	LPM Resume OK Indicates whether LPM can initiate a resume or accept a resume from host: • For Device mode, this flag means LPM can initiate a resume or accept a resume from the host. • For Host mode, this flag means LPM can initiate a resume or accept a resume from the device. • For HS Device mode, the controller waits for TL1TokenRetry to start the transition to FS mode after sending the LPM ACK response. Then, after TL1Residency, this flag is set.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• For FS Device mode, the controller waits for TL1TokenRetry after sending the LPM ACK response. Then, this flag is set. • For HS Host mode, the controller starts the transition to FS mode after receiving the LPM ACK response. Then, after (TL1TokenRetry + TL1Residency), this flag is set. • For FS or LS Host mode, this flag is set after receiving the LPM ACK response. 0b - Cannot resume 1b - Can resume
29-4 —	Reserved
3 LPM_AUTO_PHCD	Auto Low-Power Mode Enables automatic Low-Power mode. The PHY is put in Low-Power mode when in L1. PORTSC1[PHCD] becomes 1 automatically in L1. 0b - Disables 1b - Enables
2 —	Reserved
1 LPM_ERRATA_EN	Link Power Management ECN Errata Feature Enable Enables the link power management ECN errata feature. The host or device supports *Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007*. If this field is 0, the host-initiated resume duration is defined by this errata: the HIRD or BESL value (0000b) equals 50 µs and each additional increment adds 75 µs. If this field is 1, the host-initiated resume duration is defined by table X-X1 in the errata. 0b - Disables 1b - Enables
0 LPM_EN	Link Power Management Feature Enable Enables the link power management feature. The host or device supports the USB 2.0 link power management addendum. 0b - Disables 1b - Enables

51.7.1.6 USB LPM Control and Status 1 (LPM_CSR1)

Offset

Register	Offset
LPM_CSR1	A4h

Function

Includes the control and status fields for the device controller's LPM mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LPM_DEV_BESLRCVD				LPM_DEV_LNKSTRCVD				LPM_DEV...	Reserved	LPM_DEV_RSPSTS		Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										LPM_DEV...	LPM_DEV...	LPM_DEV_BESLTHRES			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 LPM_DEV_BESLRCVD	LPM Device Received BESL Indicates the device received BESL field from the LPM transaction. See LPM_CSR1[LPM_DEV_BESLTHRES] for the timing definition.
27-24 LPM_DEV_LNKSTRCVD	LPM Device Received bLinkState Indicates the value of the device received bLinkState field from the LPM transaction. 0000b - Reserved for future use 0001b - L1 (Sleep mode) 0010b-1111b - Reserved for future use
23 LPM_DEV_RWKENRCVD	LPM Device Received bRemoteWake Indicates the value of the bRemoteWake field in the received LPM transaction. You can issue a remote wake-up only when this field is 1. 0b - 0 1b - 1
22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
21-20 LPM_DEV_RSPSTS	LPM Device Response Status Indicates the device's response to the host in LPM transactions. 00b - Invalid 01b - ACK 10b - NYET 11b - STALL
19-6 —	Reserved
5 LPM_DEV_DP	LPM Device Data Pending Specifies whether LPM device data is pending. If this field is 0, the device returns a handshake on every LPM token that it receives, based on LPM_CSR1[LPM_DEV_RES]. If this field is 1, the device returns an NYET handshake after every LPM token it receives. 0b - Not pending 1b - Pending
4 LPM_DEV_RES	LPM Device Response Determines whether the additional fourth condition, mentioned below, is needed. This field is effective only when LPM_CSR1[LPM_DEV_DP] is 0. The controller responds with an ACK response when at least the first three of the following conditions are satisfied: 1. There are no PID checks or CRC5 errors in both the EXT and LPM tokens (otherwise, a response timeout error occurs). 2. Received bLinkState is 0001b (L1) (otherwise, a response STALL error occurs). 3. Received BESL or HIRD value is larger than or equal to that of LPM_CSR1[LPM_DEV_BESLTHRES] (otherwise, a response STALL error occurs). 4. No data is pending in the transmit FIFO (otherwise, a response NYET error occurs). 0b - Fourth condition not needed 1b - Fourth condition needed
3-0 LPM_DEV_BESLTHRES	Device Required Host Initiated Resume Duration Specifies the time duration that the device requires to come out of LPM-initiated suspend state after receiving the host-initiated LPM resume signal. If the received value of LPM_CSR1[LPM_DEV_BESLRCVD] is less than the value of this field, the device sends a STALL response. 0000b - 75 µs, if LPM_ERRATA_EN = 1; 50 µs, if LPM_ERRATA_EN = 0

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0001b - 100 µs, if LPM_ERRATA_EN = 1; 125 µs, if LPM_ERRATA_EN = 0 0010b - 150 µs, if LPM_ERRATA_EN = 1; 200 µs, if LPM_ERRATA_EN = 0 0011b - 250 µs, if LPM_ERRATA_EN = 1; 275 µs, if LPM_ERRATA_EN = 0 0100b - 350 µs, if LPM_ERRATA_EN = 1; 350 µs, if LPM_ERRATA_EN = 0 0101b - 450 µs, if LPM_ERRATA_EN = 1; 425 µs, if LPM_ERRATA_EN = 0 0110b - 950 µs, if LPM_ERRATA_EN = 1; 500 µs, if LPM_ERRATA_EN = 0 0111b - 1950 µs, if LPM_ERRATA_EN = 1; 575 µs, if LPM_ERRATA_EN = 0 1000b - 2950 µs, if LPM_ERRATA_EN = 1; 650 µs, if LPM_ERRATA_EN = 0 1001b - 3950 µs, if LPM_ERRATA_EN = 1; 725 µs, if LPM_ERRATA_EN = 0 1010b - 4950 µs, if LPM_ERRATA_EN = 1; 800 µs, if LPM_ERRATA_EN = 0 1011b - 5950 µs, if LPM_ERRATA_EN = 1; 875 µs, if LPM_ERRATA_EN = 0 1100b - 6950 µs, if LPM_ERRATA_EN = 1; 950 µs, if LPM_ERRATA_EN = 0 1101b - 7950 µs, if LPM_ERRATA_EN = 1; 1025 µs, if LPM_ERRATA_EN = 0 1110b - 8950 µs, if LPM_ERRATA_EN = 1; 1100 µs, if LPM_ERRATA_EN = 0 1111b - 9950 µs, if LPM_ERRATA_EN = 1; 1175 µs, if LPM_ERRATA_EN = 0

51.7.1.7 USB LPM Control and Status 2 (LPM_CSR2)

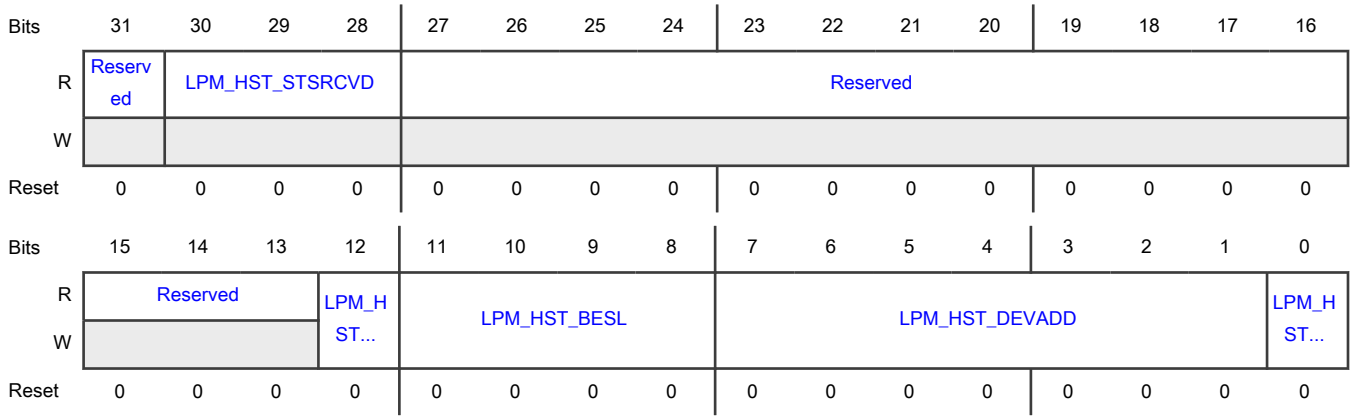
Offset

Register	Offset
LPM_CSR2	A8h

Function

Includes the control and status fields for the host controller's LPM mode.

Diagram



Fields

Field	Function
31 ——	Reserved
30-28 LPM_HST_STS RCVD	LPM Host Response Status from the Device Indicates the device's response to the host in an LPM transaction. 000b - Invalid 001b - ACK 010b - NYET 011b - STALL 100b - Timeout 101b - ERR 110b - Reserved 111b - Reserved
27-13 ——	Reserved
12 LPM_HST_RW KEN	LPM Host Extension Token's bRemoteWake Enables the extension token's bRemoteWake field. 0b - Disables 1b - Enables
11-8 LPM_HST_BES L	LPM Host Extension Token's BESL or HIRD Is carried in the extension token's BESL or HIRD field.
7-1 LPM_HST_DEV ADD	LPM Host Extension Token's Device Address Is carried in the extension token's device address field.
0 LPM_HST_SEN D	LPM Host Send Extension Token Specifies whether the LPM transaction is ongoing. Writing 1 to this field causes the host to send an extension token to the device for L1. This field becomes 0 automatically when the transaction is completed. This field can become 1 only in the L0 state. 0b - LPM transaction did not happen or is complete 1b - LPM transaction is ongoing

51.7.1.8 eUSB Control 0 (EUSB_CTRL0)

Offset

Register	Offset
EUSB_CTRL0	100h

Function

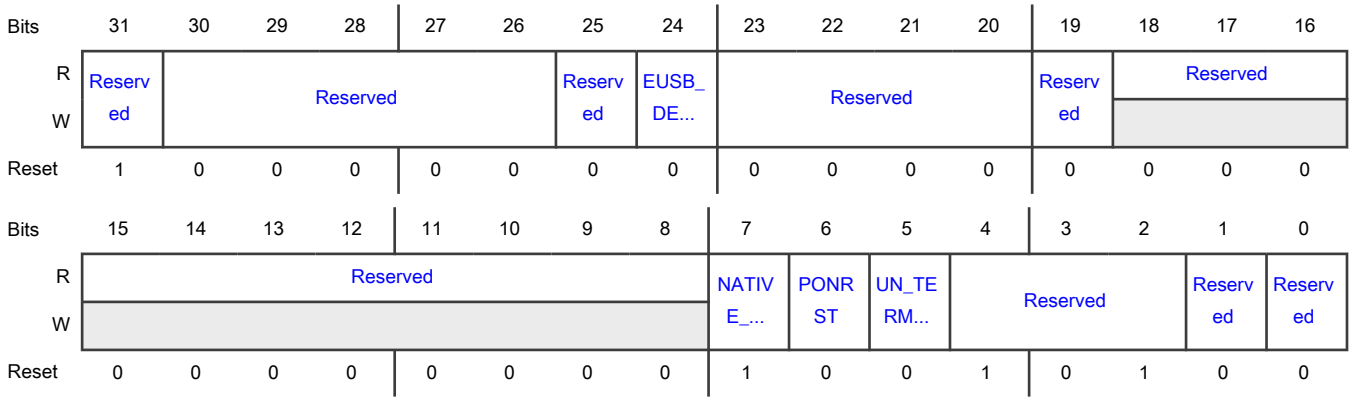
Includes some control fields for eUSBPHY.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
USBNC0	—	EUSB_CTRL0
USBNC1	EUSB_CTRL0	—

Diagram



Fields

Field	Function
31 —	Reserved
30-26 —	Reserved
25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 EUSB_DEV_PORT_RST	Trigger Device Issue Port Reset Triggers a device issue port reset. If this field is 1, device issue port reset is triggered. The hi-active pulse should be > 2T 60 MHz. 0b - No action 1b - Trigger device issue port reset
23-20 —	Reserved
19 —	Reserved
18-8 —	Reserved
7 NATIVE_MODE	Mode Select Selects the eUSBPHY mode. 0b - Repeater mode 1b - Native mode
6 PONRST	Power-On Reset Resets eUSBPHY. 0b - Reset 1b - Operate normally
5 UN_TERMINATED_MODE	HS Receiver Termination Option Controls HS receiver termination in HS native host mode. 0b - Enables 1b - Disables
4-2 —	Reserved
1 —	Reserved
0 —	Reserved

51.7.1.9 eUSB Control 1 (EUSB_CTRL1)

Offset

Register	Offset
EUSB_CTRL1	104h

Function

Includes some control bits for the eUSB PHY. User needs to toggle PONRST to configure a new value.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
USBNC0	—	EUSB_CTRL1
USBNC1	EUSB_CTRL1	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											EUSB_FSTX_OPT	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EUSB_SQ			Reserved												
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-21 —	Reserved
20-19 EUSB_FSTX_OPT	eUSB Full-Speed TX Option Adjust Full-Speed TX rise/fall time. 00b - Default 01b - Faster 10% 10b - Slower 20%

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Slower 10%
18-16 —	Reserved
15-13 EUSB_SQ	eUSB Squelch Adjust reference level of squelch. 000b - 66.6mV 001b - 63.2mV 010b - 59.7mV 011b - 56mV 100b - 80.6mV 101b - 76.9mV 110b - 73.5mV 111b - 70mV
12-0 —	Reserved

51.7.1.10 eUSB Control 3 (EUSB_CTRL3)

Offset

Register	Offset
EUSB_CTRL3	10Ch

Function

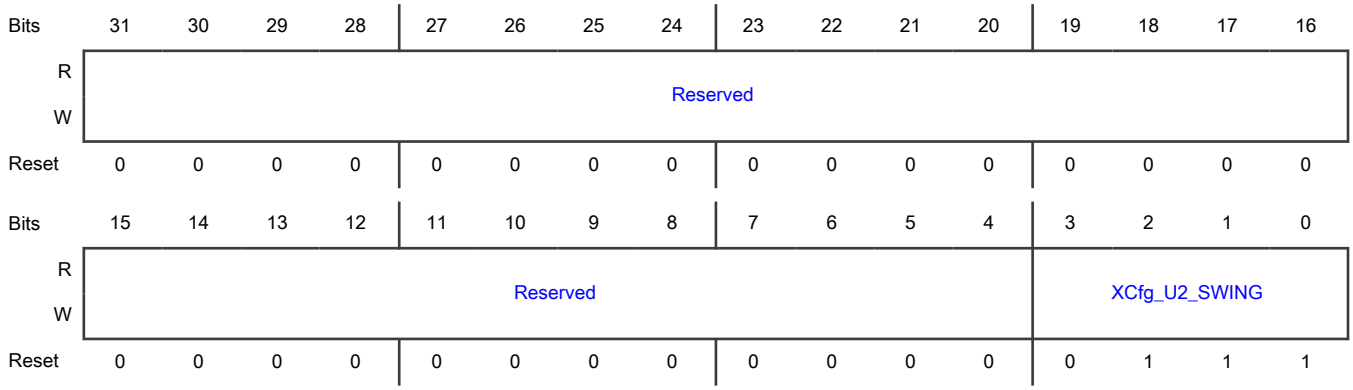
Includes some control bits for the eUSB PHY. User needs to toggle PONRST to configure a new value.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
USBNC0	—	EUSB_CTRL3
USBNC1	EUSB_CTRL3	—

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 XCfg_U2_SWING	HSTX swing Adjust the level of HSTX swing (terminated). 0000b - 200mV 0001b - 195.9mV 0010b - 191.7mV 0011b - 187.5mV 0100b - 216.7mV 0101b - 212.5mV 0110b - 208.4mV 0111b - 204.2mV 1000b - 233.4mV 1001b - 229.2mV 1010b - 225mV 1011b - 220.9mV 1100b - 250mV 1101b - 245.9mV 1110b - 241.7mV 1111b - 237.5mV

51.7.1.11 eUSB RAP Control and Status (EUSB_RAP)

Offset

Register	Offset
EUSB_RAP	148h

Function

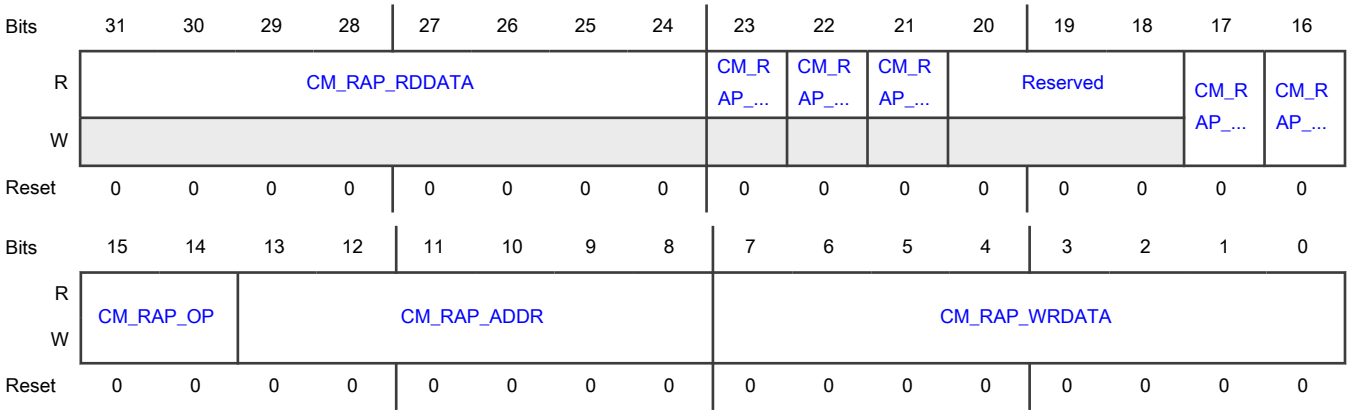
Includes RAP-related control and status fields for eUSBPHY.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
USBNC0	—	EUSB_RAP
USBNC1	EUSB_RAP	—

Diagram



Fields

Field	Function
31-24 CM_RAP_RDDATA	CM.RAP Read Data Indicates the data for the CM.RAP read command.
23 CM_RAP_INIT_READY	Host and Device Entering CM.RAP State Indicates whether the host and device are ready to transmit CM.RAP. You can program EUSB_RAP[CM_RAP_OP], EUSB_RAP[CM_RAP_ADDR], and EUSB_RAP[CM_RAP_START] to access the link partner's CSR through CM.RAP.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not ready 1b - Ready
22 CM_RAP_DONE	CM.RAP Command Done Indicates whether the CM.RAP command completed successfully. 0b - Not complete 1b - Complete
21 CM_RAP_FAIL	CM.RAP Command Fail Indicates whether the CM.RAP command finished with failure. 0b - Not failed 1b - Failed
20-18 —	Reserved
17 CM_RAP_START	CM.RAP Start Enables CM_RAP_START to send a command and communicate with CM. 0b - Disables 1b - Enables
16 CM_RAP_INIT_EN	Enable CM.RAP Feature Enables the CM.RAP feature. You must initiate RAP upon power-up and before port and repeater configuration. After this field becomes 1, and after the Port.Reset command is complete, the PHY must stay in the CM_RAP state and then write 1 to EUSB_RAP[CM_RAP_INIT_READY] to signal that it is ready to transmit CM.RAP. 0b - Disables 1b - Enables
15-14 CM_RAP_OP	CM.RAP Operation Code Specifies the CM.RAP command type. 00b - Write; data is written to the register address 01b - Read; data is read from the register address 10b - Clear; active-high, bitwise clear with the data on the register address 11b - Set; bitwise OR with the data on the register address
13-8 CM_RAP_ADDR	CM.RAP Address Specifies data for CM.RAP read or write address.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 CM_RAP_WRDATA	CM.RAP Write Data Specifies data for CM.RAP write command.

51.8 USB core memory map and register definitions

51.8.1 USBC register descriptions

51.8.1.1 USBC memory map

USB0 base address: 4041_8000h

USB1 base address: 4041_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Identification (ID)	32	R	E4A1_FA05h
4h	Hardware General (HWGENERAL)	32	R	0000_1015h
8h	Host Hardware Parameters (HWHOST)	32	R	1002_0001h
Ch	Device Hardware Parameters (HWDEVICE)	32	R	0000_0011h
10h	TX Buffer Hardware Parameters (HWTXBUF)	32	R	8008_0B08h
14h	RX Buffer Hardware Parameters (HWRXBUF)	32	R	0000_0808h
80h	General Purpose Timer 0 Load (GPTIMER0LD)	32	RW	0000_0000h
84h	General Purpose Timer 0 Controller (GPTIMER0CTRL)	32	RW	0000_0000h
88h	General Purpose Timer 1 Load (GPTIMER1LD)	32	RW	0000_0000h
8Ch	General Purpose Timer 1 Controller (GPTIMER1CTRL)	32	RW	0000_0000h
90h	System Bus Configuration (SBUSCFG)	32	RW	0000_0002h
100h	Capability Registers Length (CAPLENGTH)	8	R	40h
102h	Host Controller Interface Version (HCIVERSION)	16	R	0100h
104h	Host Controller Structural Parameters (HCSPARAMS)	32	R	0001_0011h
108h	Host Controller Capability Parameters (HCCPARAMS)	32	R	0000_0006h
120h	Device Controller Interface Version (DCIVERSION)	16	R	0001h
124h	Device Controller Capability Parameters (DCCPARAMS)	32	R	0000_0188h
140h	USB Command (USBCMD)	32	RW	0008_0000h
144h	USB Status (USBSTS)	32	RW	See section

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
148h	Interrupt Enable (USBINTR)	32	RW	0000_0000h
14Ch	USB Frame Index (FRINDEX)	32	RW	0000_0000h
154h	Device Address (DEVICEADDR)	32	RW	0000_0000h
154h	Frame List Base Address (PERIODICLISTBASE)	32	RW	0000_0000h
158h	Next Asynchronous Address (ASYNCLISTADDR)	32	RW	0000_0000h
158h	Endpoint List Address (ENDPTLISTADDR)	32	RW	0000_0000h
160h	Programmable Burst Size (BURSTSIZE)	32	RW	0000_0808h
164h	TX FIFO Fill Tuning (TXFILLTUNING)	32	RW	0000_0000h
178h	Endpoint NAK (ENDPTNAK)	32	RW	0000_0000h
17Ch	Endpoint NAK Enable (ENDPTNAKEN)	32	RW	0000_0000h
180h	Configure Flag (CONFIGFLAG)	32	R	0000_0001h
184h	Port Status and Control (PORTSC1)	32	RW	1C00_0004h
1A4h	On-The-Go Status and Control (OTGSC)	32	RW	See section
1A8h	USB Device Mode (USBMODE)	32	RW	0000_5000h
1ACh	Endpoint Setup Status (ENDPTSETUPSTAT)	32	RW	0000_0000h
1B0h	Endpoint Prime (ENDPTPRIME)	32	RW	0000_0000h
1B4h	Endpoint Flush (ENDPTFLUSH)	32	RW	0000_0000h
1B8h	Endpoint Status (ENDPTSTAT)	32	R	0000_0000h
1BCh	Endpoint Complete (ENDPTCOMPLETE)	32	RW	0000_0000h
1C0h	Endpoint Control 0 (ENDPTCTRL0)	32	RW	0080_0080h
1C4h	Endpoint Control 1 (ENDPTCTRL1)	32	RW	0000_0000h
1C8h	Endpoint Control 2 (ENDPTCTRL2)	32	RW	0000_0000h
1CCh	Endpoint Control 3 (ENDPTCTRL3)	32	RW	0000_0000h
1D0h	Endpoint Control 4 (ENDPTCTRL4)	32	RW	0000_0000h
1D4h	Endpoint Control 5 (ENDPTCTRL5)	32	RW	0000_0000h
1D8h	Endpoint Control 6 (ENDPTCTRL6)	32	RW	0000_0000h
1DCh	Endpoint Control 7 (ENDPTCTRL7)	32	RW	0000_0000h

51.8.1.2 Identification (ID)

Offset

Register	Offset
ID	0h

Function

Identifies the USB 2.0 HS core and its revision.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								REVISION							
W																
Reset	1	1	1	0	0	1	0	0	1	0	1	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NID						Reserved		ID					
W																
Reset	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1

Fields

Field	Function
31-24 —	Reserved
23-16 REVISION	Revision Number Indicates the revision number of the controller core.
15-14 —	Reserved
13-8 NID	Complement Version Indicates the complement version of the ID.
7-6 —	Reserved
5-0 ID	Configuration Number Indicates the configuration number (05h) and signifies that the peripheral is USB 2.0 HS core.

51.8.1.3 Hardware General (HWGENERAL)

Offset

Register	Offset
HWGENERAL	4h

Function

Indicates general hardware parameters as defined in system-level issues and core configuration.

NOTE

The reset value can vary from instance to instance. Therefore, for more accurate information, see field descriptions and ignore their reset values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			LPM	SM		PHYM				PHYW		Reserved			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1

Fields

Field	Function
31-13 —	Reserved
12 LPM	Link Power Management Capability Indicates whether USB 2.0 link power management ECN is supported. 0b - Not supported 1b - Supported
11-10 SM	Serial Interface Mode Capability Indicates serial interface mode capability. 00b - No serial engine; always use parallel signaling 01b - Serial engine present; always use serial signaling for FS and LS

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Software programmable; reset to use parallel signaling for FS and LS 11b - Software programmable; reset to use serial signaling for FS and LS
9-6 PHYM	Transceiver Type Indicates the transceiver type. 0000b - UTMI/UMTI+ 0001b - ULPI DDR 0010b - ULPI 0011b - Serial only 0100b - Software programmable: reset to UTMI/UTMI+ 0101b - Software programmable: reset to ULPI DDR 0110b - Software programmable: reset to ULPI 0111b - Software programmable: reset to Serial 1000b - IC-USB 1001b - Software programmable: reset to IC-USB 1010b - HSIC 1011b - Software programmable: reset to HSIC 1100b-1111b - Reserved
5-4 PHYW	PHY Width Indicates the data width of the transceiver connected to the controller core. 00b - 8-bit wide data bus (software nonprogrammable) 01b - 16-bit wide data bus (software nonprogrammable) 10b - Reset to 8-bit wide data bus (software programmable) 11b - Reset to 16-bit wide data bus (software programmable)
3-0 —	Reserved

51.8.1.4 Host Hardware Parameters (HWHOST)

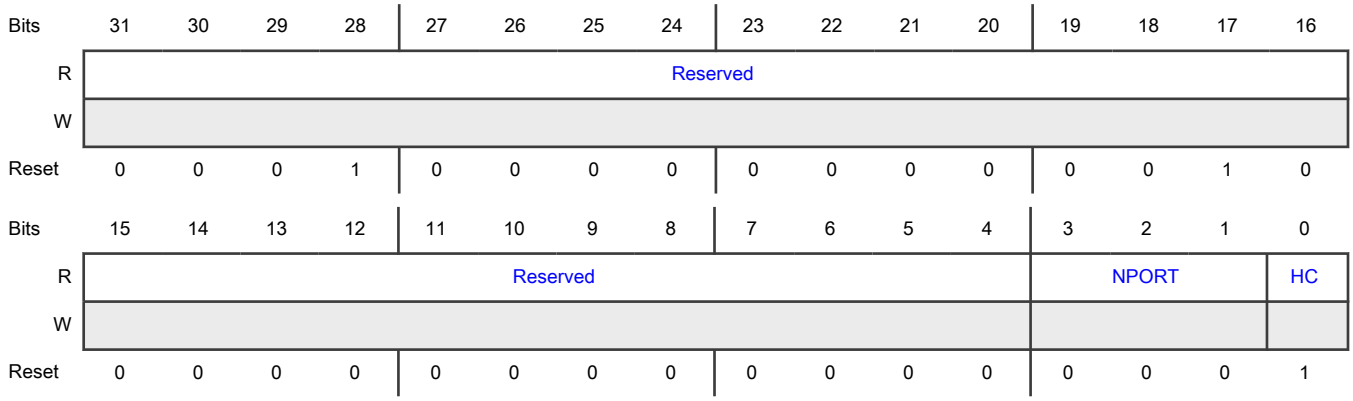
Offset

Register	Offset
HWHOST	8h

Function

Indicates the host hardware parameters, the number of ports, and the mode supported.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-1 NPORT	Number of Ports Indicates the number of downstream ports supported by the host controller. It is defined as NPORT + 1. **NOTE** A value of 000 indicates a single-port host controller.
0 HC	Host Capable Indicates whether Host Operation mode is supported. 0b - Not supported 1b - Supported

51.8.1.5 Device Hardware Parameters (HWDEVICE)

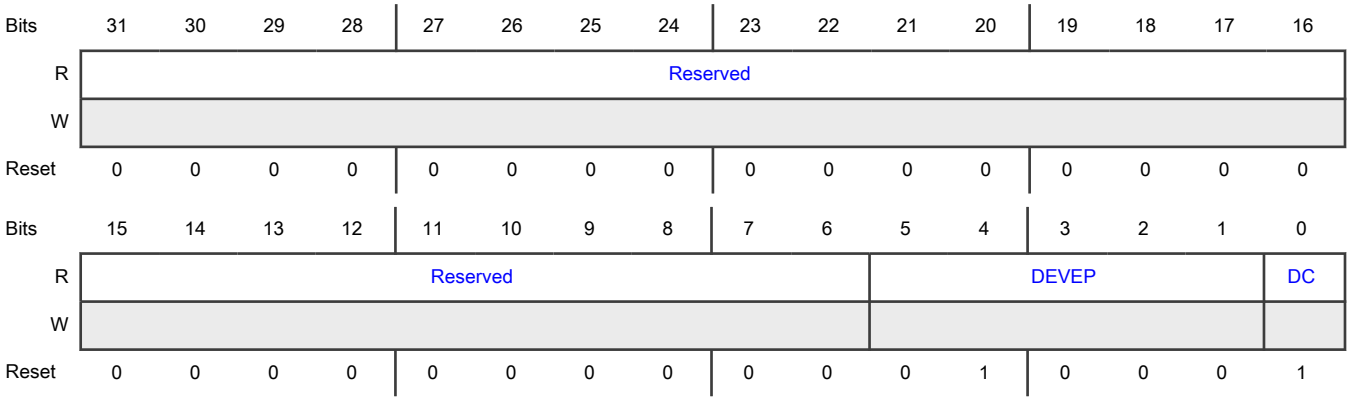
Offset

Register	Offset
HWDEVICE	Ch

Function

Indicates the device hardware parameters such as the device endpoint number and the mode supported.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-1 DEVEP	Device Endpoint Number Indicates the device endpoint number.
0 DC	Device Capable Indicates whether Device Operation mode is supported. 0b - Not supported 1b - Supported

51.8.1.6 TX Buffer Hardware Parameters (HWTXBUF)

Offset

Register	Offset
HWTXBUF	10h

Function

Indicates TX buffer hardware parameters, such as the default burst size and total buffer size.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXCHANADD							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TXBURST							
W																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-16 TXCHANADD	TX Channel Add Indicates the TX FIFO buffer size. The TX FIFO buffer size is (2^TXCHANADD) × 4 bytes. This field becomes 08h, meaning the buffer size is 256 × 4 bytes. If USB is in Device mode, this is the FIFO buffer size per endpoint. Because the controller has eight TX endpoints, there are eight of these buffers. For the USB controller operating in Host mode, or for host-only controller, the entire buffer memory operates as a single TX buffer. Therefore, there is only one.
15-8 —	Reserved
7-0 TXBURST	TX Burst Indicates the default burst size for memory to TX buffer transfer. This is the reset value of BURSTSIZE[TXPBURST].

51.8.1.7 RX Buffer Hardware Parameters (HWRXBUF)

Offset

Register	Offset
HWRXBUF	14h

Function

Indicates RX buffer hardware parameters, such as the default burst size and total buffer size.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXADD								RXBURST							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-8 RXADD	RX Add Indicates the RX FIFO buffer size. The total buffer size for all receive endpoints is: (2^RXADD). RX buffer size is: (2^RXADD) × 4 bytes. This field becomes 08h, meaning the buffer size is 256 × 4 bytes. There is a single receive FIFO buffer in the USB controller. The buffer is shared for all endpoints when USB is in Device mode.
7-0 RXBURST	RX Burst Indicates the default burst size for memory to RX buffer transfer. This is the reset value of BURSTSIZE[RXPBURST].

51.8.1.8 General Purpose Timer 0 Load (GPTIMER0LD)

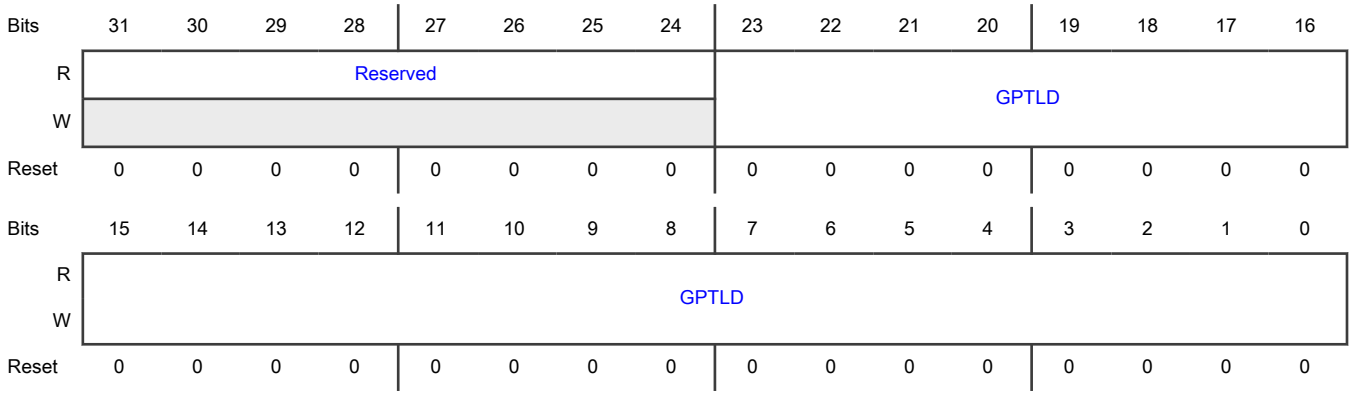
Offset

Register	Offset
GPTIMER0LD	80h

Function

Controls the load value of the count timer in General Purpose Timer 0 Controller (GPTIMER0CTRL).

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 GPTLD	General Purpose Timer Load Value Indicates the general purpose timer load value. This value represents the time, in microseconds minus 1, of the timer duration. The value of this field is loaded to GPTIMER0CTRL[GPTCNT], if GPTIMER0CTRL[GPTRST] is 1. For example, for a one millisecond timer, load 1000 - 1 = 999 (or 0003E7h). **NOTE** The maximum value is FFFFFFh or 16.777215 seconds.

51.8.1.9 General Purpose Timer 0 Controller (GPTIMER0CTRL)

Offset

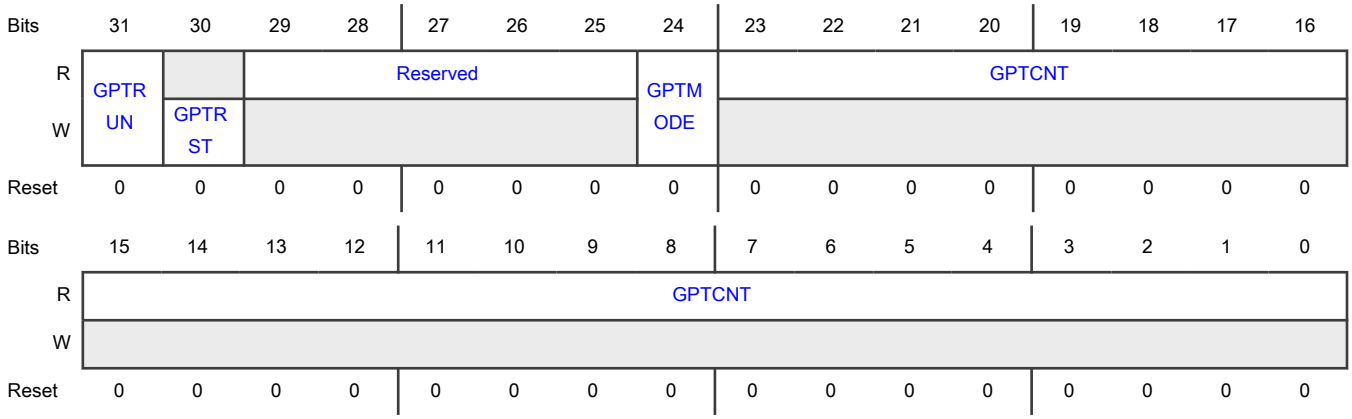
Register	Offset
GPTIMER0CTRL	84h

Function

Contains the control for the general purpose countdown timer 0. A data field can be queried to determine the running count value. This timer has granularity on 1 µs and you can program it to a little over 16 seconds. There are two counter modes that are described in this register. When the timer counter value transitions to zero, an interrupt may be generated if it is enabled.

USBSTS[TI0] is the interrupt status field, and USBINTR[TIE0] is the interrupt enable field.

Diagram



Fields

Field	Function
31 GPTRUN	General Purpose Timer Run Specifies whether the general purpose timer is running or has stopped counting. GPTIMER0CTRL[GPTCNT] is not impacted when writing 1 or 0 to this field. 0b - Stopped counting 1b - Running
30 GPTRST	General Purpose Timer Reset Loads counter value from GPTIMER0LD[GPTLD]. 0b - No action 1b - Load counter value
29-25 —	Reserved
24 GPTMODE	General Purpose Timer Mode Specifies the general purpose timer mode. In One Shot mode, the timer counts down to zero, generates an interrupt, and halts until you reset the counter. In Repeat mode, the timer counts down to zero, generates an interrupt, and automatically reloads the counter value from General Purpose Timer 1 Load (GPTIMER1LD) to start again. 0b - One Shot mode 1b - Repeat mode
23-0 GPTCNT	General Purpose Timer Counter Indicates the count value of the countdown timer.

51.8.1.10 General Purpose Timer 1 Load (GPTIMER1LD)

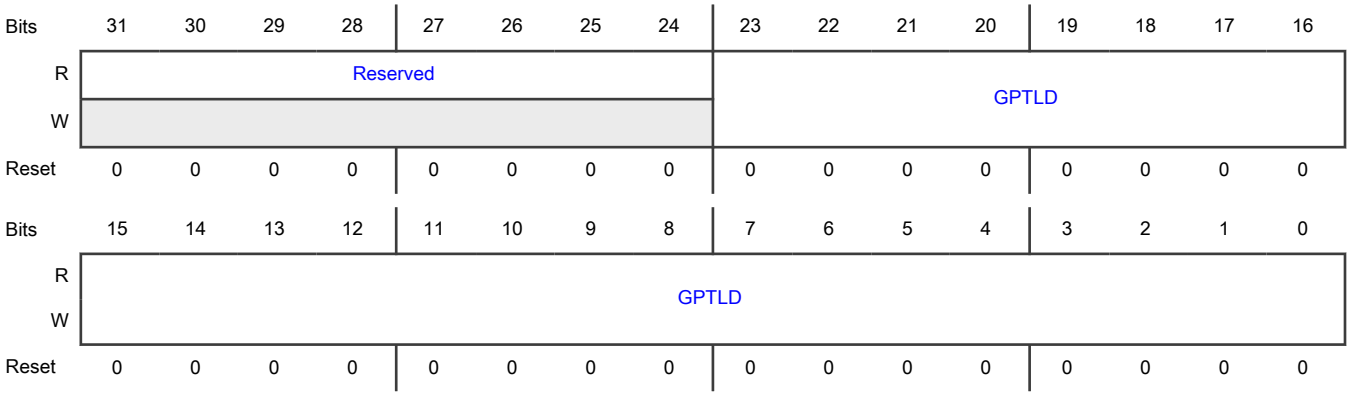
Offset

Register	Offset
GPTIMER1LD	88h

Function

Controls the load value of the count timer in General Purpose Timer 1 Controller (GPTIMER1CTRL).

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 GPTLD	General Purpose Timer Load Value Indicates the general purpose timer load value. The value of this field is loaded to GPTIMER0CTRL[GPTCNT], if GPTIMER0CTRL[GPTRST] is 1. This value represents the time, in microseconds minus 1, of the timer duration. For example, for a 1 ms timer, load 1000 - 1 = 999 (or 0003E7h). **NOTE** The maximum value is FFFFFFh or 16.777215 seconds.

51.8.1.11 General Purpose Timer 1 Controller (GPTIMER1CTRL)

Offset

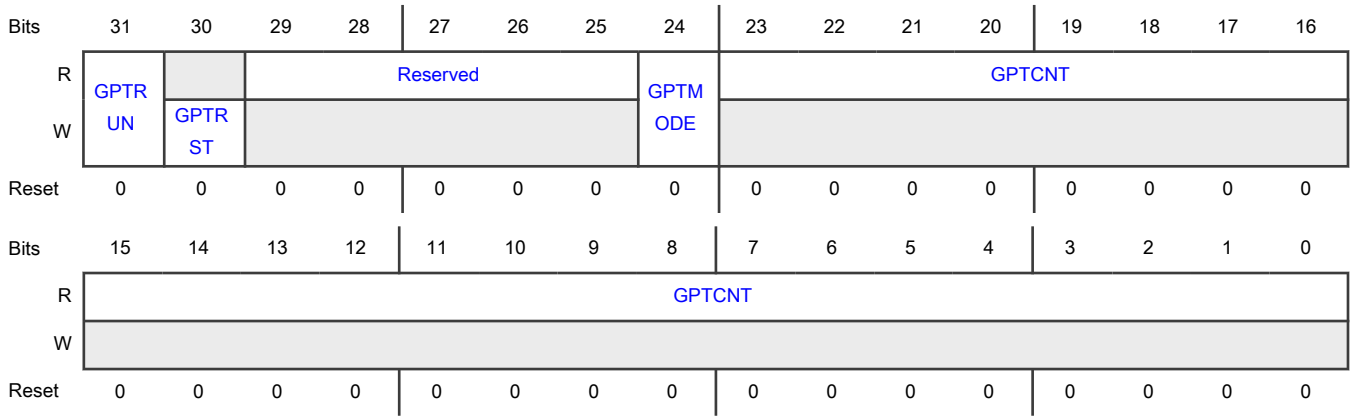
Register	Offset
GPTIMER1CTRL	8Ch

Function

Contains the control for the general purpose countdown timer 1. A data field can be queried to determine the running count value. This timer has granularity on 1 µs and you can program it to a little over 16 seconds. There are two counter modes that are described in this register. When the timer counter value transitions to zero, an interrupt may be generated if it is enabled.

USBSTS[TI1] is the interrupt status field, and USBINTR[TIE1] is the interrupt enable field.

Diagram



Fields

Field	Function
31 GPTRUN	General Purpose Timer Run Specifies whether the general purpose timer is running or has stopped counting. GPTIMER0CTRL[GPTCNT] is not impacted when writing 1 or 0 to this field. 0b - Stopped counting 1b - Running
30 GPTRST	General Purpose Timer Reset Loads counter value from GPTIMER0LD[GPTLD]. 0b - No action 1b - Load counter value
29-25 —	Reserved
24 GPTMODE	General Purpose Timer Mode Indicates the general purpose timer mode. In One Shot mode, the timer counts down to zero, generates an interrupt, and stops until you reset the counter. In Repeat mode, the timer counts down to zero, generates an interrupt, and automatically reloads the counter value from GPTIMER0LD[GPTLD] to start again.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - One Shot mode 1b - Repeat mode
23-0 GPTCNT	General Purpose Timer Counter Indicates the count value of the countdown timer.

51.8.1.12 System Bus Configuration (SBUSCFG)

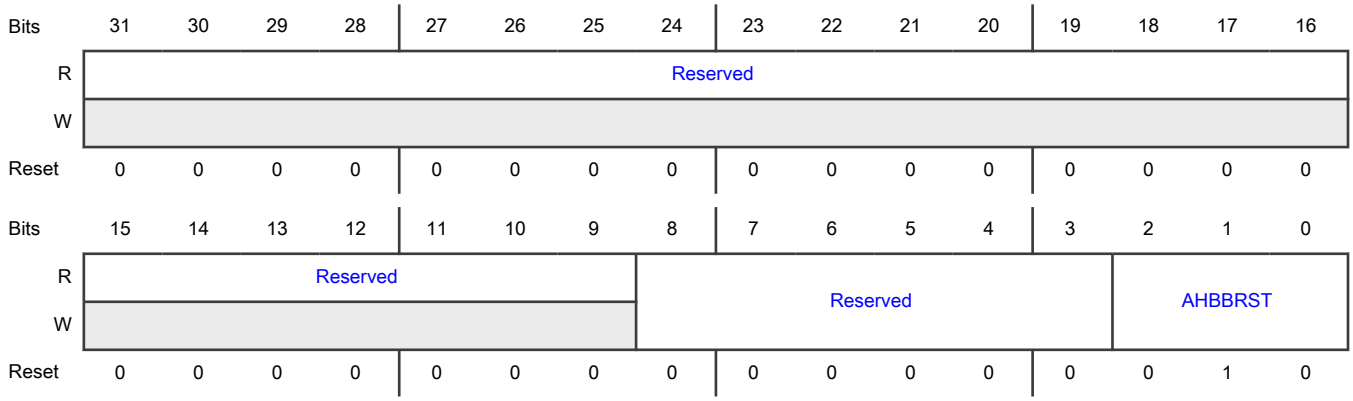
Offset

Register	Offset
SBUSCFG	90h

Function

Defines the system bus configuration.

Diagram



Fields

Field	Function
31-9 —	Reserved
8-3 —	Reserved
2-0 AHBBRST	AHB Manager Interface Burst Configuration Controls the AHB manager transfer type sequence (or priority).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This register overrides Programmable Burst Size (BURSTSIZE) when the value of AHBBRST is not zero. 000b - Incremental burst of unspecified length only 001b - INCR4 burst, then single transfer 010b - INCR8 burst, INCR4 burst, then single transfer 011b - INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100b - Reserved, do not use 101b - INCR4 burst, then incremental burst of unspecified length 110b - INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111b - INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

51.8.1.13 Capability Registers Length (CAPLENGTH)

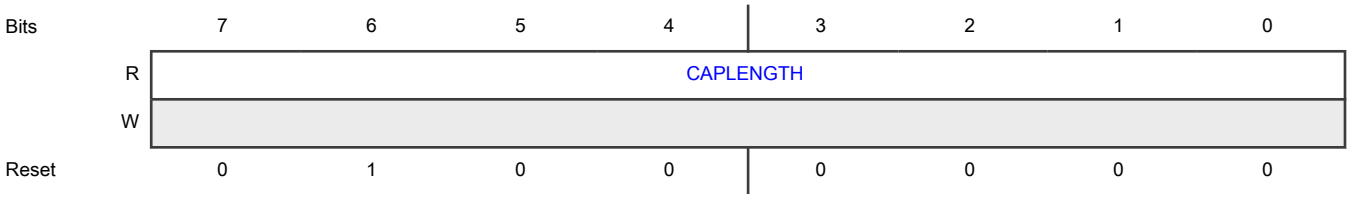
Offset

Register	Offset
CAPLENGTH	100h

Function

Contains the address offset of the operational registers relative to this register.

Diagram



Fields

Field	Function
7-0 CAPLENGTH	Capability Length Indicates the capability length. This field is used as an offset to add to the register base to find the beginning of the operational register. The default value is 40h.

51.8.1.14 Host Controller Interface Version (HCIVERSION)

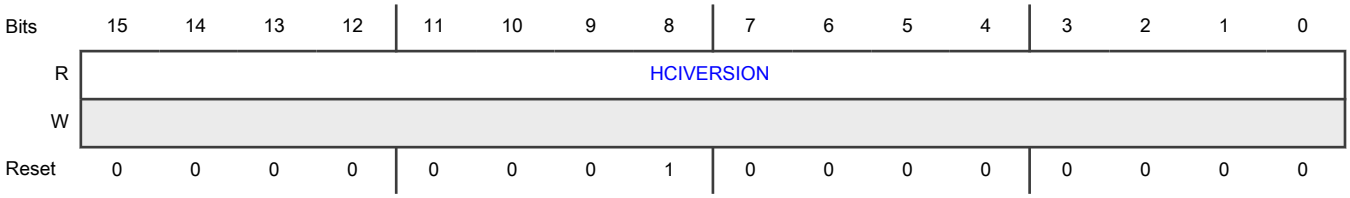
Offset

Register	Offset
HCIVERSION	102h

Function

Contains a BCD encoding of the EHCI revision number that this host controller supports. The most significant byte of this register represents a major revision and the least significant byte indicates the minor revision.

Diagram



Fields

Field	Function
15-0 HCIVERSION	Host Controller Interface Version Number Indicates the host controller interface version number. The default value is 10h, which specifies EHCI rev1.0.

51.8.1.15 Host Controller Structural Parameters (HCSPARAMS)

Offset

Register	Offset
HCSPARAMS	104h

Function

Indicates the port steering logic capabilities of the host control structural parameters (HCSPARAMS).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				N_TT				N_PTT				Reserved			PI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N_CC				N_PCC				Reserved			PPC	N_PORTS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Fields

Field	Function
31-28 —	Reserved
27-24 N_TT	Number of Transaction Translators (N_TT) Indicates the number of embedded transaction translators associated with the USB 2.0 host controller. The value of this field is 0001b for multi-port host, and 0000b for single-port host. The default value of this field is 0000b.
23-20 N_PTT	Number of Ports per Transaction Translator (N_PTT) Indicates the number of ports assigned to each transaction translator within the USB 2.0 host controller. The value of this field is equal to that of HCSPARAMS[N_PORTS] for a multi-port host, and 0000b for a single-port host. The default value of this field is 0000b.
19-17 —	Reserved
16 PI	Port Indicators (P_INDICATOR) Indicates whether the ports support port indicator control. If this field is 1, the port status and control registers include an R/W field for controlling the state of the port indicator. The value of this field is 1b in all controller cores.
15-12 N_CC	Number of Companion Controller (N_CC) Indicates the number of companion controllers associated with the USB 2.0 host controller. The value of this field is 0000b in all controller cores. If this field is 0, there is no internal companion controller and port-ownership hand-off is not supported. If this field is 1, internal companion controllers exist and port-ownership hand-offs are supported.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000b - No internal companion controller exists 0001b - Internal companion controllers exist
11-8 N_PCC	Number of Ports per Companion Controller Indicates the number of ports supported per internal companion controller. This field is used to indicate the port routing configuration to the system software. For example, if HCSPARAMS[N_PORTS] has a value of 6 and HCSPARAMS[N_CC] has a value of 2, then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, and so on. In the previous example, N_PCC could have been 4, where the first four are routed to companion controller 1 and the last two are routed to companion controller 2. The value of this field must be consistent with that of HCSPARAMS[N_PORTS] and HCSPARAMS[N_CC]. The value of this field is 0000b in all controller cores.
7-5 —	Reserved
4 PPC	Port Power Control Indicates whether the host controller implementation includes port power control. The value of this field affects the functionality of PORTSC1[PP] in each port status and control register. If this field is 1, it indicates that the ports have port power switches. If this field is 0, it indicates that the ports do not have port power switches. 0b - No port power switches 1b - Port power switches exist
3-0 N_PORTS	Number of Ports Indicates the number of downstream ports. This field indicates the number of physical downstream ports implemented on this host controller. The value of this field determines the number of port registers that are addressable in the operational register. The range of valid values is 1h–Fh. A zero in this field is undefined. The value of this field is always 0001b because all controller cores are single-port hosts.

51.8.1.16 Host Controller Capability Parameters (HCCPARAMS)

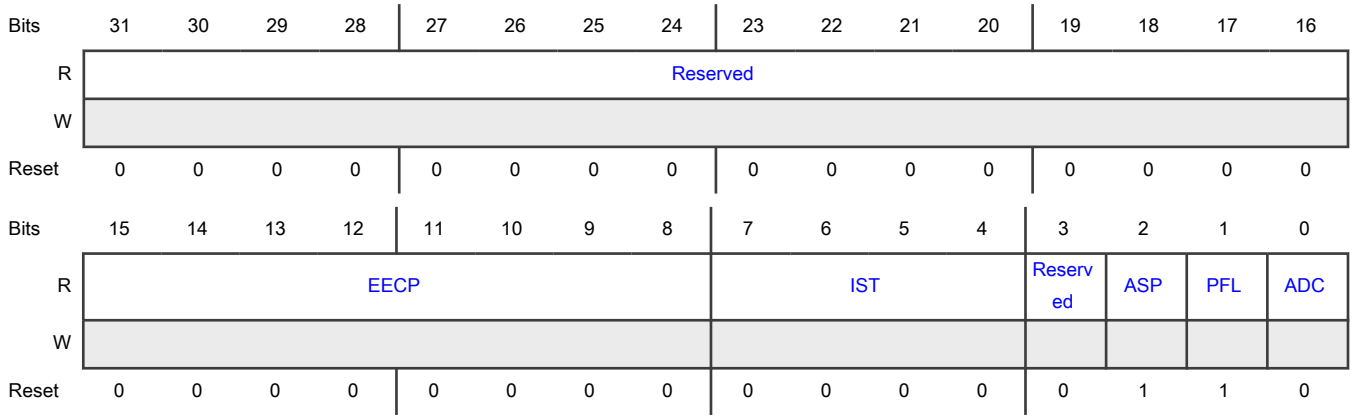
Offset

Register	Offset
HCCPARAMS	108h

Function

Identifies multiple mode control (time-based bit functionality) and addressing capability.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 EECP	EHCI Extended Capabilities Pointer Indicates the existence of a capabilities list. A value of 00h indicates that no extended capabilities are implemented. A nonzero value in this field indicates the offset of the first EHCI extended capability in the PCI configuration space. The pointer value must be 40h or greater (if implemented) to maintain the consistency of the PCI header defined for this class of device. The value of this field is 00h in all controller cores.
7-4 IST	Isochronous Scheduling Threshold Indicates, relative to the current position of the executing host controller, where you can reliably update the isochronous schedule. When bit 7 of this field is 0, the value of the least significant three bits indicates the number of microframes during which a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When the value of bit 7 is 1, then the host software assumes that the host controller can cache an isochronous data structure for an entire frame. The value of this field is 00h in all controller cores.
3 —	Reserved
2 ASP	Asynchronous Schedule Park Capability Indicates whether the park feature for HS queue heads is supported. If this field is 1, the host controller supports the park feature for HS queue heads in the asynchronous schedule. You can enable or disable this feature and set it to a specific level by using USBCMD[ASPE] and USBCMD[ASP].

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 PFL	Programmable Frame List Flag Indicates the frame list length. If this flag is 0, you must use a frame list length of 1024 elements with this host controller. You must write 0 to USBCMD[FS_2] and USBCMD[FS_1]. If this flag is 1, you can specify and use a smaller frame list and configure the host controller via USBCMD[FS_2] and USBCMD[FS_1]. The frame list must always be aligned on a 4 KB page boundary. This requirement ensures that the frame list is always physically contiguous. The value of this flag is 1 in all controller cores.
0 ADC	Addressing Capability Indicates 64-bit addressing capability. This field is 0b in all controller cores, and 64-bit addressing capability is not supported.

51.8.1.17 Device Controller Interface Version (DCIVERSION)

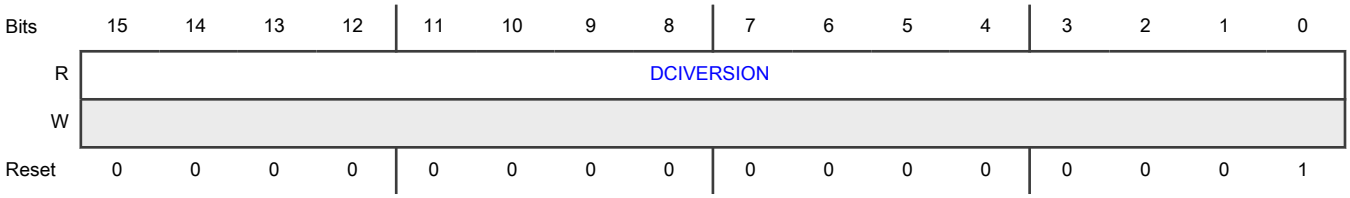
Offset

Register	Offset
DCIVERSION	120h

Function

Indicates the 2-byte BCD encoding of the device controller interface version number.

Diagram



Fields

Field	Function
15-0 DCIVERSION	Device Controller Interface Version Number Indicates the device controller interface version number. The default value of this field is 01h, which specifies rev 0.1.

51.8.1.18 Device Controller Capability Parameters (DCCPARAMS)

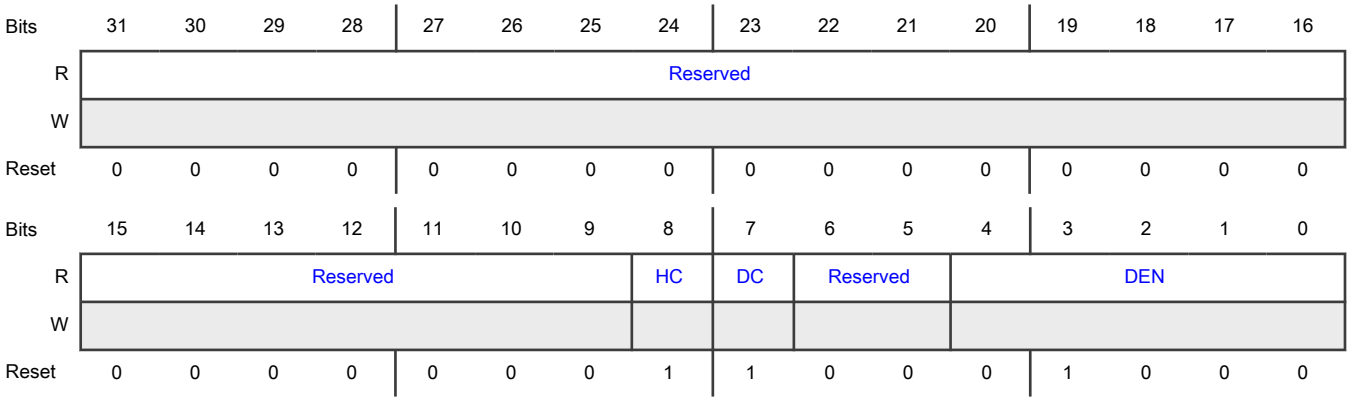
Offset

Register	Offset
DCCPARAMS	124h

Function

Indicates the overall device capability of the controller.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 HC	Host Capable Indicates whether the controller is capable of operation as a host controller. If this field is 1, this controller is capable of operating as an EHCI-compatible USB 2.0 host controller. 0b - Not host capable 1b - Host capable
7 DC	Device Capable Indicates whether the controller is capable of operation. If this field is 1, the controller is capable of operating as a USB 2.0 device. 0b - Not device capable 1b - Device capable
6-5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
4-0 DEN	Device Endpoint Number Indicates the number of endpoints built into the device controller. If this controller is not device capable, then this field is 0. Valid values are 0–15. 0_0000b - Not device capable 0_0001b - Device capable

51.8.1.19 USB Command (USBCMD)

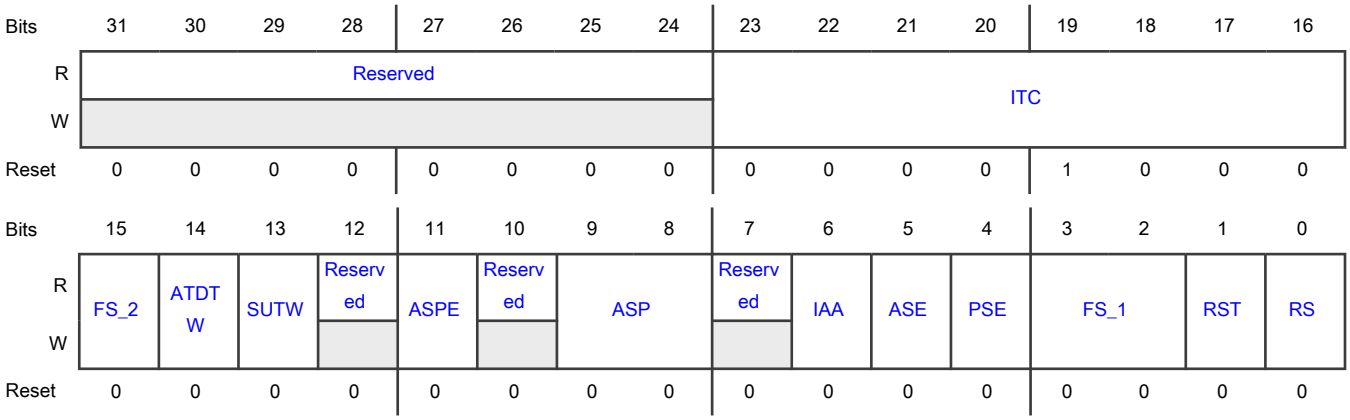
Offset

Register	Offset
USBCMD	140h

Function

Indicates the command that the serial bus host or device controllers must execute. Writing to this register causes a command to be executed.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 ITC	Interrupt Threshold Control Sets the maximum rate at which the host or device controller issues interrupts. This field contains the maximum interrupt interval measured in microframes. Valid values are shown below.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The following values are the valid maximum interrupt intervals: 0000_0000b - Immediate (no threshold) 0000_0001b - 1 microframe 0000_0010b - 2 microframes 0000_0100b - 4 microframes 0000_1000b - 8 microframes 0001_0000b - 16 microframes 0010_0000b - 32 microframes 0100_0000b - 64 microframes
15 FS_2	Frame List Size 2 (Host mode only) Specifies the size of the frame list that controls which fields in FRINDEX you must use for the current frame list index. This field is both readable and writable only if HCCPARAMS[PFL] is 1, otherwise it is only readable. **NOTE** Frame List Size[2:0] = {FS_2, FS_1[1:0]} 000b - 1024 elements (4096 bytes) 001b - 512 elements (2048 bytes) 010b - 256 elements (1024 bytes) 011b - 128 elements (512 bytes) 100b - 64 elements (256 bytes) 101b - 32 elements (128 bytes) 110b - 16 elements (64 bytes) 111b - 8 elements (32 bytes)
14 ATDTW	Add dTD Trip Wire (Device mode only) Works as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. The hardware can also write 0 to this field when state machine is in hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 SUTW	Setup Trip Wire (Device mode only) Works as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a queue head by the DCD without being corrupted. If the setup lockout mode is off (USBMODE[SLOM] is 1), then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the queue head of a previous setup packet. The hardware can also write 0 to this field when a hazard is detected.

Table continues on the next page...

Table continued from the previous page...

Field	Function
12 —	Reserved
11 ASPE	Asynchronous Schedule Park Mode Enable Enables Park mode. If HCCPARAMS[ASP] is 1, then this field defaults to 1h with RW as its access type. Otherwise, the field is 0 with access type RO. If this field is 1, Park mode is enabled, and if this field is 0, Park mode is disabled. 0b - Disables 1b - Enables
10 —	Reserved
9-8 ASP	Asynchronous Schedule Park Mode Count Contains a count of the number of successive transactions that the host controller is allowed to execute from an HS queue head on the asynchronous schedule, before continuing traversal of the asynchronous schedule. If HCCPARAMS[ASP] is 1, then this field defaults to 3h, with R/W as the access type. Otherwise, the field defaults to 0, with access type RO. Valid values are 1h–3h. You must not write 0 to this field when USBCMD[ASPE] is 1. Doing so leads to undefined behavior. This field is 3h for all controller cores.
7 —	Reserved
6 IAA	Interrupt on Async Advance Doorbell Acts as a doorbell for the host controller to issue an interrupt the next time it advances an asynchronous schedule. You must write 1 to this field to ring the doorbell. After the host controller evicts all appropriate cached schedule states, it writes 1 to USBSTS[AAI]. If USBINTR[AAE] is 1, then the host controller asserts an interrupt at the next interrupt threshold. The host controller writes 0 to this field after it writes 1 to USBSTS[AAI]. You must not write 1 to this field when the asynchronous schedule is inactive. Doing so yields undefined results. This field is only used in Host mode. Writing 1 to this field in Device mode leads to undefined results.
5 ASE	Asynchronous Schedule Enable Determines whether the host controller skips processing the asynchronous schedule. If this field is 1, Next Asynchronous Address (ASYNCLISTADDR) is used to access the periodic schedule. Only the host controller uses this field. 0b - Do not process the asynchronous schedule

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Access the asynchronous schedule
4 PSE	Periodic Schedule Enable Determines whether the host controller skips processing the periodic schedule. If this field is 1, Frame List Base Address (PERIODICLISTBASE) is used to access the periodic schedule. Only the host controller uses this field. 0b - Do not process the periodic schedule 1b - Process the periodic schedule
3-2 FS_1	Frame List Size 1 See the description of USBCMD[FS_2].
1 RST	Controller Reset Resets the controller. The host or device controller writes 0 to this field after the reset process is complete. You cannot terminate the reset process early by writing 0 to this field. In Host operation mode, the controller resets its internal pipelines, timers, counters, state machines, and so on, to their initial values when you write 1 to this field. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. You must not write 1 to this field when USBSTS[HCH] is 0. Attempting to reset an actively running host controller results in undefined behavior. In Device operation mode, the controller resets its internal pipelines, timers, counters, state machines, and so on, to their initial values when you write 1 to this field. Writing 1 to this field when the chip is in the attached state is not recommended because the effect on an attached host is undefined. To ensure that the chip is not in an attached state before initiating a device controller reset, all primed endpoints must be flushed and USBCMD[RS] must be 0.
0 RS	Run/Stop Specifies whether the controller is executing the operation. In Host operation mode, the controller proceeds with the execution of the schedule if the value of this field is 1b. The controller continues the execution as long as this field is 1. When this field becomes 0, the host controller completes the current transaction on the USB and then halts. The Halted bit in the Status field of the dTD token indicates when the controller has finished the transaction and has entered the stopped state. You must not write 1 to this field unless the controller is in the halted state (that is, USBSTS[HCH] is 1). In Device operation mode, writing 1 to this field causes the controller to enable a pull-up on D+ and initiate an attach event. This control field is not directly connected to the pull-up enable because the pull-up becomes disabled after transitioning into HS mode. You must use this field to prevent an attach event before the controller is properly initialized. Writing 0 to this field leads to a detach event. 0b - Stopped executing 1b - Running

51.8.1.20 USB Status (USBSTS)

Offset

Register	Offset
USBSTS	144h

Function

Indicates various states of the host and device controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LPM_HST...	LPM_DEV...	LPM_L1_...	LPM_L1_...	Reserved		TI1	TI0	Reserved				UPI	UAI	Reserved	NAKI
W	W1C	W1C	W1C	W1C			W1C	W1C					W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AS	PS	RCL	HCH	Reserved	ULPII	Reserved	SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
W						W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	u	0	0	0	0	0	0	0

Fields

Field	Function
31 LPM_HST_COMPI	Host Completes the LPM Transaction Interrupt Flag Indicates when the host receives a handshake or timeout for the LPM transaction. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
30 LPM_DEV_RCVDI	Device Received Extension Token Interrupt Flag Indicates when the device received the extension token.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
29 LPM_L1_ENTRYI	L1 Entry Interrupt Flag Indicates when the device or host transitions to the L1 state and the PHY is put in Low-Power mode. The field is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
28 LPM_L1_EXITI	L1 Exit Interrupt Flag Indicates when the device or host exits the L1 state. The field is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
27-26 —	Reserved
25	General Purpose Timer Interrupt 1 (GPTINT1) Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
TI1	Indicates when the counter in General Purpose Timer 1 Controller (GPTIMER1CTRL) transitions to 0. This field is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
24 TI0	General Purpose Timer Interrupt 0 (GPTINT0) Flag Indicates when the counter in General Purpose Timer 0 Controller (GPTIMER0CTRL) transitions to 0. This field is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
23-20 —	Reserved
19 UPI	USB Host Periodic Interrupt Flag Indicates the cause of a periodic interrupt and whether a short packet is detected. The host controller sets this field when the cause of an interrupt is the completion of a USB transaction, where the transfer descriptor (TD) has the IOC field set and the TD is from a periodic schedule. The host controller also sets this field when a short packet is detected (the actual number of bytes received is less than expected) and the packet is on a periodic schedule. The device controller does not use this field; this field is always 0. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
18 UAI	USB Host Asynchronous Interrupt Flag Indicates the cause of an asynchronous interrupt and whether a short packet is detected. The host controller sets this field when the cause of an interrupt is the completion of a USB transaction, where the transfer descriptor (TD) has the IOC field set and the TD is from an asynchronous schedule. The host controller also sets this field when a short packet is detected (the actual number of bytes received is less than expected) and the packet is on an asynchronous schedule. The device controller does not use this field; this field is always 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
17 —	Reserved
16 NAKI	NAK Interrupt Indicates whether a NAK interrupt is detected. Hardware sets this field, when ENDPTNAK[EPTN] and ENDPTNAK[EPRN] are both set for a particular endpoint, and their corresponding fields (ENDPTNAKEN[EPTNE] and ENDPTNAKEN[EPRNE]) are 1. When all enabled TX and RX endpoint NAK fields are cleared, this field is cleared automatically.
15 AS	Asynchronous Schedule Status Reports the current real status of the asynchronous schedule. When this field is 0, the asynchronous schedule status is disabled, and when this field is 1, the status is enabled. The host controller is not required to immediately disable or enable the asynchronous schedule when you transition USBCMD[ASE]. When this field and USBCMD[ASE] are the same value, the asynchronous schedule is either enabled (1) or disabled (0). This field is only used in Host Operation mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disabled 1b - Enabled
14 PS	Periodic Schedule Status Reports the current real status of the periodic schedule. If this field is 0, the periodic schedule is disabled, and if this field is 1, the status is enabled. The host controller is not required to immediately disable or enable the periodic schedule when you transition USBCMD[PSE]. When this field and USBCMD[PSE] are the same value, the periodic schedule is either enabled (1) or disabled (0). This field is only used in Host Operation mode. 0b - Disabled 1b - Enabled
13 RCL	Reclamation Detects an empty asynchronous schedule. This field is only used in Host Operation mode. 0b - Does not detect 1b - Detects
12 HCH	HC Halted Indicates whether the host controller is halted. This field is 0 whenever USBCMD[RS] is 1. The controller writes 1 to this field when you or the controller hardware write 0 to USBCMD[RS] after it has stopped executing (for example, in case of an internal error). This field is only used in Host Operation mode.
11 —	Reserved
10 ULPII	ULPI Interrupt Flag Indicates an event completion in the ULPI viewport. Hardware sets this field to 1b when there is an event completion in the ULPI viewport. You can only use this field if the controller supports UPLI interface mode. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Event completion did not occur 1b - Event completion occurred When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
9 —	Reserved
8 SLI	Device Controller Suspend Flag Indicates whether the device controller entered a suspended state. When a device controller enters a suspended state from an active state, this field is set to 1. This field is only used in Device Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Did not enter Suspended state 1b - Entered Suspended state When writing 0b - No effect 1b - Clear the flag
7 SRI	SOF Received Flag Indicates whether a start of (micro) frame (SOF) is received. When the device controller detects an SOF, this field is set to 1. When the SOF is extremely late, the device controller automatically sets this field to indicate that an SOF was expected. Therefore, this field is set roughly every 1 ms in device FS mode, and every 125 µs in HS mode. It is synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this field sets at an interval of 1 ms during the prelude to connect and chirp. In Host mode, this bit sets every 125 µs; the host controller driver can use it as a time base. **NOTE** This field behaves differently for register reads and writes. When reading 0b - SOF not received 1b - SOF received When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
6 URI	USB Reset Received Indicates whether a USB reset is received. When the device controller detects a USB reset and enters the default state, this field is set to 1. This field is used only in Device Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0b - USB reset not received 1b - USB reset received When writing 0b - No effect 1b - Clear the flag
5 AAI	Interrupt on Asynchronous Advance Flag Indicates the assertion of the interrupt source. You can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing 1 to USBCMD[IAA]. This field is used only in Device Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
4 SEI	System Error Flag Indicates an error response to a read on the system interface. This field is set to 1b when there is an error response to a read on the system interface. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Error response did not occur

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Error response occurred When writing 0b - No effect 1b - Clear the flag
3 FRI	Frame List Rollover Flag Indicates the frame list index has rolled over. The host controller sets this field to 1 when the frame list index rolls over from its maximum value to 0. The exact value at which the rollover occurs depends on the frame list size. For example, if USBCMD[FS_2] and USBCMD[FS_1] are 000b, USB Frame Index (FRINDEX) rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the host controller sets this field to 1 every time FRINDEX[12] toggles. This field is used only in Host Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Frame list index did not roll over 1b - Frame list index rolled over When writing 0b - No effect 1b - Clear the flag
2 PCI	Port Change Detect Flag Indicates port change detection. The host controller sets this field to 1 when a connect status occurs on any port, a port enable or disable change occurs, or PORTSC1[FPR] becomes 1 as the result of a J-K transition on the suspended port. The device controller sets this field to 1 when the port controller enters the FS or HS operational state. When the port controller exits the FS or HS operation state because of reset or suspend events, the notification mechanisms are controlled by USBSTS[URI] and PORTSC1[SUSP], respectively. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Port change not detected 1b - Port change detected When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 UEI	USB Error Interrupt (USBERRINT) Flag Indicates an error condition after the completion of a USB transaction. The host or device controller sets this field after the completion of a USB transaction results in an error condition. This field is set along with USBSTS[UI], if the TD on which the error interrupt occurred also has its IOC field set in the dTD token. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag
0 UI	USB Interrupt (USBINT) Flag Indicates the cause of an interrupt. The host or device controller sets this field when the cause of an interrupt is the completion of a USB transaction where the transfer descriptor (TD) also has its IOC field set in the dTD token. The host or device controller also sets this field when a short packet is detected (the actual number of bytes received is less than the expected number of bytes). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Interrupt did not occur 1b - Interrupt occurred When writing 0b - No effect 1b - Clear the flag

51.8.1.21 Interrupt Enable (USBINTR)

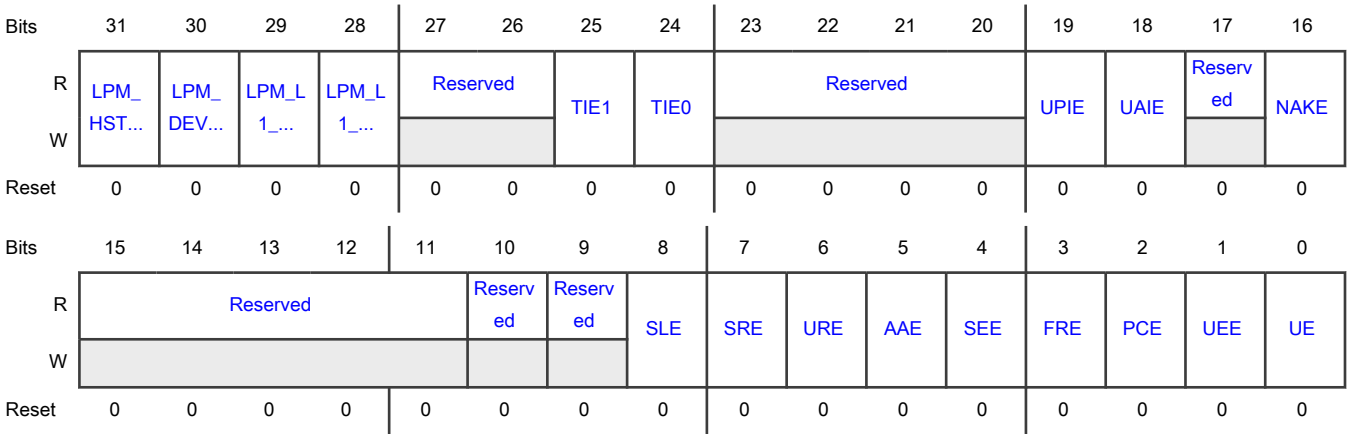
Offset

Register	Offset
USBINTR	148h

Function

Enables interrupts to software. An interrupt is generated when a field in this register is 1 and the corresponding interrupt source is active. USB Status (USBSTS) still shows interrupt sources even if they are disabled by USBINTR, allowing you to poll the interrupt events.

Diagram



Fields

Field	Function
31 LPM_HST_COMPIE	Host Completed LPM Transaction Interrupt Enable Enables LPM transaction interrupt. When this field is 1 and USBSTS[LPM_HST_COMPI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0b - Disables 1b - Enables
30 LPM_DEV_RCVDIE	Device Received Extension Token Interrupt Enable Enables device received extension token interrupt. When this field is 1 and USBSTS[LPM_DEV_RCVDI] is 1 too, the controller issues an interrupt. This field is used only in Device Operation mode. 0b - Disables 1b - Enables
29 LPM_L1_ENTRYIE	L1 Entry Interrupt Enable Enables L1 entry interrupt. When this field is 1 and USBSTS[LPM_L1_ENTRYI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
28	L1 Exit Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
LPM_L1_EXITIE	Enables L1 exit interrupt. When this field is 1 and USBSTS[LPM_L1_EXITI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
27-26 —	Reserved
25 TIE1	General Purpose Timer 1 Interrupt Enable Enables general purpose timer 1. When this field is 1 and USBSTS[TI1] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
24 TIE0	General Purpose Timer 0 Interrupt Enable Enables general purpose timer 0. When this field is 1 and USBSTS[TI0] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
23-20 —	Reserved
19 UPIE	USB Host Periodic Interrupt Enable Enables USB host periodic interrupt. When this field is 1 and USBSTS[UPI] is 1 too, the host controller issues an interrupt at the next interrupt threshold. 0b - Disables 1b - Enables
18 UAIE	USB Host Asynchronous Interrupt Enable Enables USB host asynchronous interrupt. When this field is 1 and USBSTS[UAI] is 1 too, the host controller issues an interrupt at the next interrupt threshold. 0b - Disables 1b - Enables
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 NAKE	NAK Interrupt Enable Enables NAK interrupt. When this field is 1 and USBSTS[NAKI] is 1 too, the controller issues an interrupt. This field is used only in Device Operation mode. 0b - Disables 1b - Enables
15-11 —	Reserved
10 —	Reserved
9 —	Reserved
8 SLE	Sleep Interrupt Enable Enables sleep interrupt. When this field is 1 and USBSTS[SLI] is 1 too, the controller issues an interrupt. This field is used only in Device Operation mode. 0b - Disables 1b - Enables
7 SRE	SOF Received Interrupt Enable Enables SOF received interrupt. When this field is 1 and USBSTS[SRI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
6 URE	USB Reset Interrupt Enable Enables the USB reset interrupt. When this field is 1 and USBSTS[URI] is 1 too, the controller issues an interrupt. This field is used only in Device Operation mode. 0b - Disables 1b - Enables
5 AAE	Asynchronous Advance Interrupt Enable Enables asynchronous advance interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When this field is 1 and USBSTS[AAI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0b - Disables 1b - Enables
4 SEE	System Error Interrupt Enable Enables system error interrupt. When this field is 1 and USBSTS[SEI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0b - Disables 1b - Enables
3 FRE	Frame List Rollover Interrupt Enable Enables the frame list rollover interrupt. When this field is 1 and USBSTS[FRI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0b - Disables 1b - Enables
2 PCE	Port Change Detect Interrupt Enable Enables port change detect interrupt. When this field is 1 and USBSTS[PCI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
1 UEE	USB Error Interrupt Enable Enables USB error interrupt. When this field is 1 and USBSTS[UEI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables
0 UE	USB Interrupt Enable Enables USB interrupt. When this field is 1 and USBSTS[UI] is 1 too, the controller issues an interrupt. 0b - Disables 1b - Enables

51.8.1.22 USB Frame Index (FRINDEX)

Offset

Register	Offset
FRINDEX	14Ch

Function

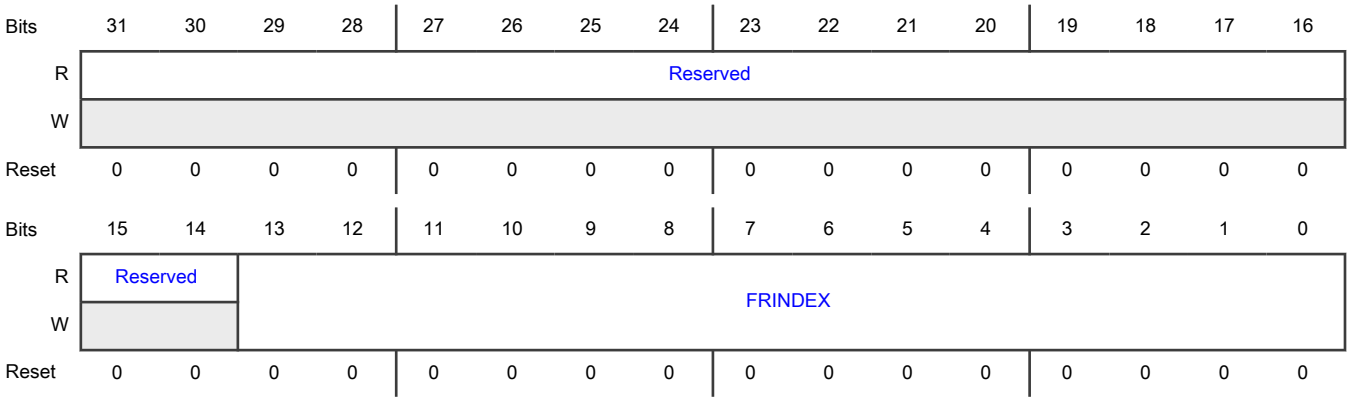
Indexes the periodic frame list.

The host controller uses this field to index the periodic frame list. The register updates every 125 microseconds (once each microframe). Bits[*n*:3] are used to select a particular entry in the periodic frame list during periodic schedule execution. The number of bits used for the index depends on the size of the frame list that you define by using USBCMD[FS_2] and USBCMD[FS_1].

You must write to this register as a DWord. Byte writes produce undefined results. You cannot write to this register unless the host controller is in the Halted state, as indicated by USBSTS[HCH]. A write to this register, when USBCMD[RS] is 1, produces undefined results. Writes to this register also affect the SOF value.

In Device mode, this register is read-only and the device controller updates FRINDEX [13:3] from the frame number indicated by the SOF marker. Whenever the USB bus receives an SOF, FRINDEX [13:3] is checked against the SOF marker. If FRINDEX [13:3] is different from the SOF marker, FRINDEX [13:3] is set to the SOF value and FRINDEX [2:0] is 0 (that is, SOF for 1 ms frame). If FRINDEX [13:3] is equal to the SOF value, FRINDEX [2:0] increments (that is, SOF for 125 µs microframe).

Diagram



Fields

Field	Function
31-14 —	Reserved
13-0 FRINDEX	Frame Index Increments at the end of each time frame (microframe). Bits[*n*:3] are used for the frame list current index. This means that each location of the frame list is accessed eight times (frames or microframes) before moving to the next index. In Host mode, the following values of *n* are based on the values of USBCMD[FS_2] and USBCMD[FS_1].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	In Device mode, the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode, bits 2:0 indicate the current microframe. The following values are represented as USBCMD (frame list size) number elements *n*. 00_0000_0000_0000b - (1024) 12 00_0000_0000_0001b - (512) 11 00_0000_0000_0010b - (256) 10 00_0000_0000_0011b - (128) 9 00_0000_0000_0100b - (64) 8 00_0000_0000_0101b - (32) 7 00_0000_0000_0110b - (16) 6 00_0000_0000_0111b - (8) 5

51.8.1.23 Device Address (DEVICEADDR)

Offset

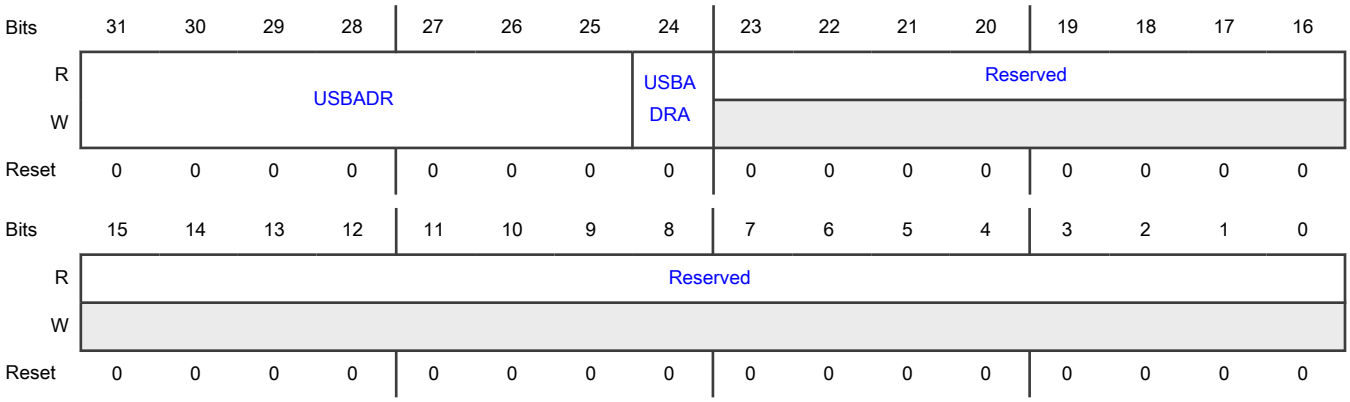
Register	Offset
DEVICEADDR	154h

Function

Contains the device address. After a controller or USB reset, the device address is set to the default address (0) that matches all incoming addresses. You must reprogram the address after receiving a SET_ADDRESS descriptor.

Only the device controller uses this register.

Diagram



Fields

Field	Function
31-25 USBADR	Device Address Represents the USB device address.
24 USBADRA	Device Address Advance Accelerates the SET_ADDRESS sequence by allowing the DCD to preset DEVICEADDR[USBADR] before the status phase of the SET_ADDRESS descriptor. When this field is 0, any writes to USBADR are instantaneous. If you write 1 to this field at the same time or before writing to USBADR, a write to USBADR is staged and held in a hidden register. After an "In" transaction occurs on endpoint 0 and is acknowledged, USBADR loads from the holding register. Hardware writes 0 to this field when all these conditions occur: • The "In" transaction is acknowledged to endpoint 0 (USBADR is updated from the staging register) • "Out" or "Setup" occur to endpoint 0 (USBADR is not updated) • Device reset occurs (USBADR is reset to 0) **NOTE** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism ensures that this specification is met when the DCD is busy and cannot write to this field within 2 ms from the SET_ADDRESS status phase. If the DCD writes to the USBADR field when USBADRA = 1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR field is programmed instantly at the correct time and meets the 2 ms USB requirement.
23-0 —	Reserved

51.8.1.24 Frame List Base Address (PERIODICLISTBASE)

Offset

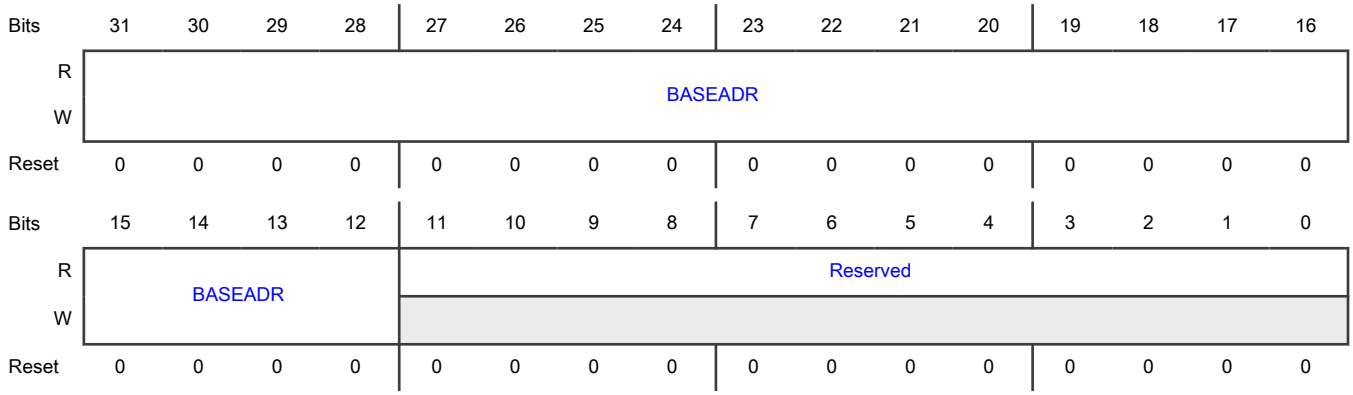
Register	Offset
PERIODICLISTBASE	154h

Function

Contains the beginning address of the periodic frame list in the system memory.

The host controller driver (HCD) loads this register before the host controller starts the schedule execution. The memory structure referenced by this physical memory pointer is assumed to be 4-KB aligned. The contents of this register are combined with USB Frame Index (FRINDEX) to enable the host controller to step through the periodic frame list in sequence.

Diagram



Fields

Field	Function
31-12 BASEADR	Base Address (Low) Specifies the base address. The bits of this field correspond to memory address signals [31:12], respectively. Only the host controller uses this field.
11-0 —	Reserved

51.8.1.25 Next Asynchronous Address (ASYNCLISTADDR)

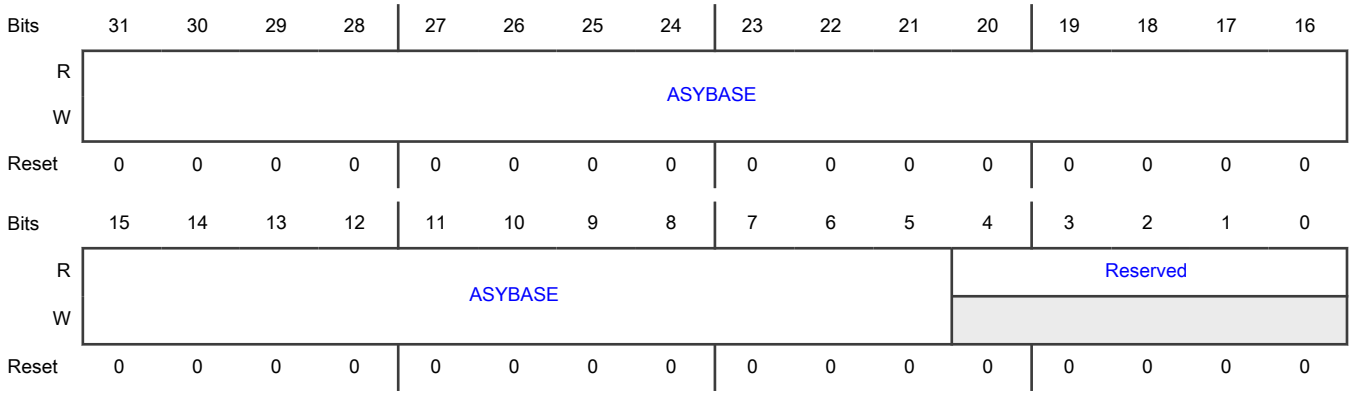
Offset

Register	Offset
ASYNCLISTADDR	158h

Function

Contains the address of the next asynchronous queue head to be executed by the host. You cannot modify bits [4:0] of this register; they always return a 0 when read.

Diagram



Fields

Field	Function
31-5 ASYBASE	Link Pointer Low (LPL) Specifies the link pointer. The bits of this field correspond to memory address signals [31:5], respectively. This field can only reference a queue head. Only the host controller uses this field.
4-0 —	Reserved

51.8.1.26 Endpoint List Address (ENDPTLISTADDR)

Offset

Register	Offset
ENDPTLISTADDR	158h

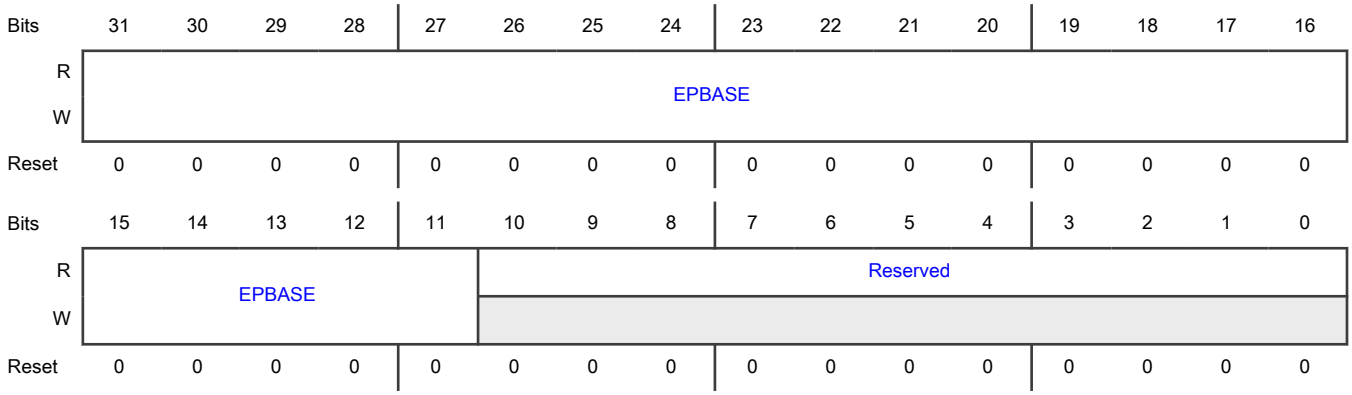
Function

Contains the address of the top of the endpoint list in system memory when in Device mode.

You cannot modify bits[10:0] of this register; they always return a 0 when read.

The memory structure referenced by this physical memory pointer is assumed to be 64 bytes.

Diagram



Fields

Field	Function
31-11 EPBASE	Endpoint List Pointer (Low) Specifies the endpoint list pointer. The bits of this field correspond to memory address signals [31:11], respectively. The field references a list of up to 32 queue heads (that is, one queue head per endpoint and direction). Only the device controller uses this field.
10-0 —	Reserved

51.8.1.27 Programmable Burst Size (BURSTSIZE)

Offset

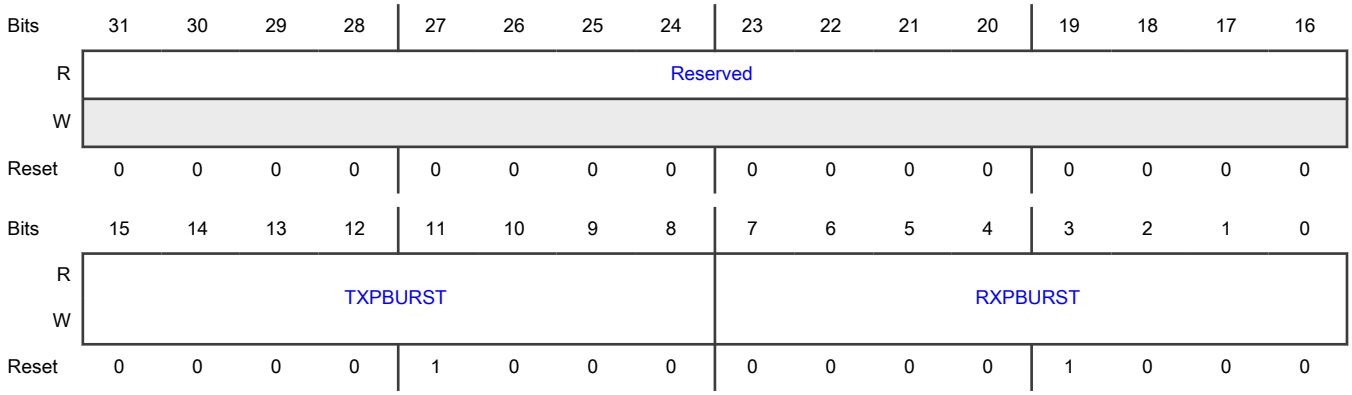
Register	Offset
BURSTSIZE	160h

Function

Controls the burst size used during data movement on the AHB manager interface.

This register is ignored if SBUSCFG[AHBBRST] has a nonzero value.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 TXPBURST	Programmable TX Burst Size Represents the maximum length of a burst, in 32-bit words, when moving data from system memory to the USB bus. HWTXBUF[TXBURST] determines the default value of this field.
7-0 RXPBURST	Programmable RX Burst Size Represents the maximum length of a burst, in 32-bit words, when moving data from the USB bus to system memory. HWRXBUF[RXBURST] determines the default value of this field.

51.8.1.28 TX FIFO Fill Tuning (TXFILLTUNING)

Offset

Register	Offset
TXFILLTUNING	164h

Function

Controls performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus.

The specific areas of performance include how much data to post into the FIFO and an estimate for how long that operation must take in the target system.

Following are some related definitions:

- T_0 = Standard packet overhead
- T_1 = Time to send data payload

- T_{ff} = Time to fetch packet into TX FIFO up to the specified level
- T_s = Total packet flight time (send-only) packet
- $T_s = T_0 + T_1$
- T_p = Total packet time (fetch and send) packet
- $T_p = T_{ff} + T_0 + T_1$

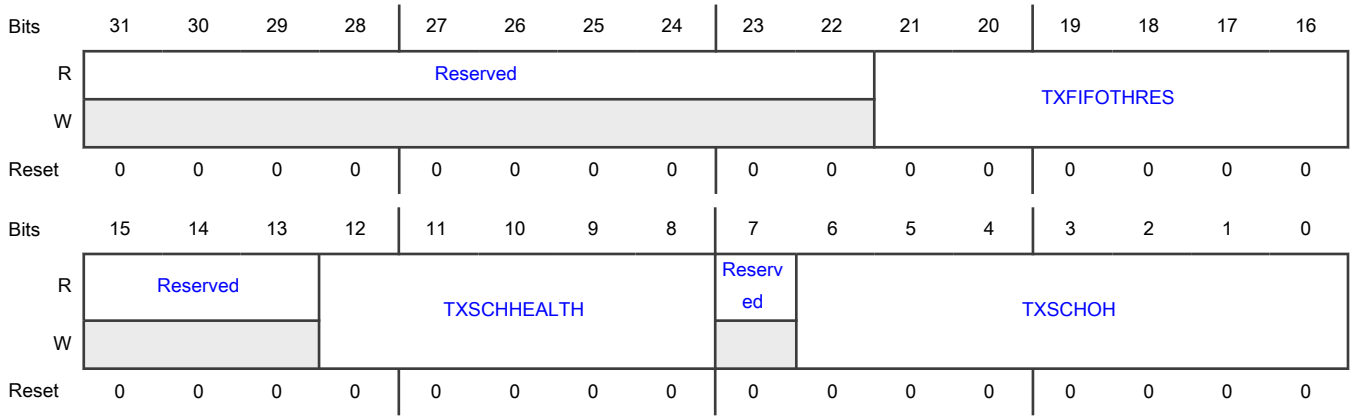
After the discovery of a transmit (Out or Setup) packet in the data structures, the host controller checks to ensure that T_p remains before the end of the microframe. If so, it proceeds to prefill the TX FIFO. If the remaining time in the frame of microframe is < T_s at any time during this operation, then the packet attempt ceases and the packet is tried at a later time.

Although this is not an error condition and the host controller eventually recovers, a mark is created in the scheduler health counter to note the occurrence of a back-off event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that begins after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and should therefore be minimized (although not necessarily eliminated). Back-off events can be minimized by using TXFILLTUNING[TXSCHHEALTH] (T_{ff}).

NOTE

The reset value can vary from instance to instance. In that case, refer to the field descriptions of this register.

Diagram



Fields

Field	Function
31-22 —	Reserved
21-16 TXFIFOTHRES	FIFO Burst Threshold Controls the number of data bursts that are posted to the TX latency FIFO in Host mode before the packet begins on to the bus. This value must be as low as possible to maximize USB performance. The minimum value is two. A higher value can be used in systems with unpredictable latency and insufficient bandwidth, where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from the system memory. This value is ignored if USBMODE[SDIS] is 1.
15-13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
12-8 TXSCHHEALTH	Scheduler Health Counter Increments when the host controller fails to fill the TX latency FIFO to the level that you program in TXFILLTUNING[TXFIFOTHRES] before running out of time to send the packet before the next SOF. This health counter measures the number of times this occurs to provide feedback for selecting a proper scheduler overhead (TXFILLTUNING[TXSCHOH]). Writing to this register clears this field. The maximum value of the field is 31.
7 —	Reserved
6-0 TXSCHOH	Scheduler Overhead Adds an additional fixed offset to the schedule time estimator described above as T_{ff}. As an approximation, the value selected for this register must limit the number of back-off events captured in TXFILLTUNING[TXSCHHEALTH] to less than 10 per second in a highly utilized bus. Selecting a value that is too high for this register is not recommended because it can needlessly reduce USB utilization. The time unit represented in this register is 1.267 µs when a device is connected in HS mode and is 6.333 µs when a device is connected in LS or FS mode.

51.8.1.29 Endpoint NAK (ENDPTNAK)

Offset

Register	Offset
ENDPTNAK	178h

Function
Configures RX and TX endpoint NAKs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								EPTN							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								EPRN							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-16 EPTN	TX Endpoint NAK Flag Indicates a transmit endpoint NAK. Each TX endpoint has one bit in this field, and that bit sets when the device sends a NAK handshake on a received "In" token for the corresponding endpoint. Bit [*n*] - Endpoint [*n*], *n* is 0-7. This field is used only in Device mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000b - No NACK 0000_0001b - NACK When writing 0000_0000b - No effect 0000_0001b - Clear the flag
15-8 —	Reserved
7-0 EPRN	RX Endpoint NAK Flag Indicates a receive endpoint NAK. Each RX endpoint has one bit in this field, and that bit sets when the device sends a NAK handshake on a received "Out" or "Ping" token for the corresponding endpoint. Bit [*n*] - Endpoint [*n*], *n* is 0-7 This field is used only in Device mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000b - No NACK 0000_0001b - NACK When writing 0000_0000b - No effect 0000_0001b - Clear the flag

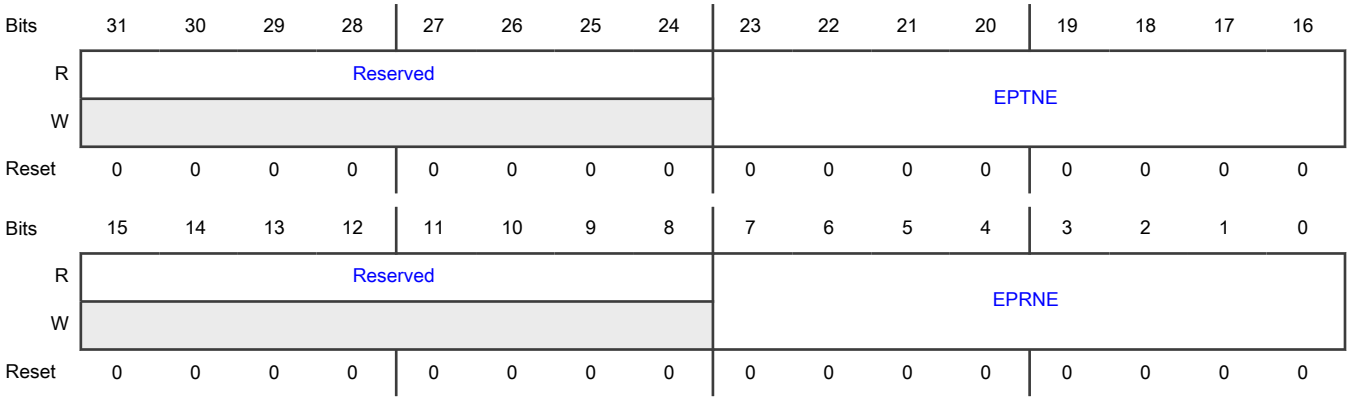
51.8.1.30 Endpoint NAK Enable (ENDPTNAKEN)

Offset

Register	Offset
ENDPTNAKEN	17Ch

Function
Enables RX and TX endpoint NAKs.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 EPTNE	TX Endpoint NAK Enable Enables transmit endpoint NAK. Each bit in this field enables the corresponding TX endpoint NAK bit. If a bit is set and the corresponding TX endpoint NAK bit is set, the NAK interrupt bit is set. Bit [*n*] - Endpoint [*n*], *n* is 0-7
15-8 —	Reserved
7-0 EPRNE	RX Endpoint NAK Enable Enables receive endpoint NAK. Each bit in this field enables the corresponding RX endpoint NAK bit. If a bit is set and the corresponding RX endpoint NAK bit is set, the NAK interrupt bit is set. Bit [*n*] - Endpoint [*n*], *n* is 0-7

51.8.1.31 Configure Flag (CONFIGFLAG)

Offset

Register	Offset
CONFIGFLAG	180h

Function
Controls port-routing control logic.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															CF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-1 —	Reserved
0 CF	Configure Flag Controls the default port-routing control logic. You must set this flag as the last action in the process of configuring the host controller. If this flag is 0, the port routing control logic routes each port to an implementation dependent on classic host controller by default. If this flag is 1, the port routing control logic routes all ports to this host controller by default. 0b - Port routing to classic host controller 1b - Port routing to this host controller

51.8.1.32 Port Status and Control (PORTSC1)

Offset

Register	Offset
PORTSC1	184h

Function

Determines and implements ports that need service.

A host controller can implement one to eight port status and control registers. The number is determined by HCSPARAMS[N_PORTS]. You can read this parameter register to determine how many ports need service.

All controller cores on this product are single-port, so there is only one port status and control register for each controller core.

This register is only reset by POR or controller core reset. The following are the initial conditions of a port:

- No device connected
- Port disabled

If the port supports power control, this state remains until software supplies port power.

A device controller operating in Device mode has only PORTSC1 and does not support power control in that mode. Port control in Device mode is only used for status port reset, suspend, and current connect status. It is also used to initiate Test mode or force signaling, and allows you to place the PHY into Low-Power Suspend mode and disable the PHY clock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PTS_1		STS	PTW	PSPD		PTS_2	PFSC	PHCD	WKOC	WKDC	WKCN	PTC			
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIC		PO	PP	LS		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
W											W1C		W1C		W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Function
31-30 PTS_1	Parallel Transceiver Select 1 **NOTE** All USB port interface modes are listed in this field description, but not all are supported. For details related to each controller core, see Features. The behavior is unknown when an unsupported interface mode is selected. {bit25, bit31, bit30}:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	000b - UTMI/UTMI+ 001b - Reserved 010b - ULPI 011b - Serial/USB 1.1 PHY/IC-USB (FS only) 100b - HSIC
29 STS	Serial Transceiver Select Selects the serial transceiver. The serial interface engine can be used in combination with the UTMI+/ULPI physical interface to provide FS and LS signaling instead of the parallel interface signals. When this field is 1, the serial interface engine is used instead of parallel interface signals. This field has no effect unless PORTSC1[PTS_1] is 1 (to select the UTMI+/ULPI interface). The serial/USB1.1 PHY/IC-USB uses the serial interface engine for FS and LS signaling regardless of the value of this field. 0b - Parallel interface signals 1b - Serial interface engine
28 PTW	Parallel Transceiver Width Selects the parallel transceiver width. This field has no effect if the serial interface engine is used. 0b - 8-bit UTMI interface (60 MHz) 1b - 16-bit UTMI interface (30 MHz)
27-26 PSPD	Port Speed Indicates the speed at which the port is operating. 00b - FS 01b - LS 10b - HS 11b - Undefined
25 PTS_2	Parallel Transceiver Select 2 See description of PORTSC1[PTS_1].
24 PFSC	Port Force Full Speed Connect Specifies the speed at which the port is connected. If this field is 1, the port is forced to connect at FS only. It disables the chirp sequence that allows the port to identify itself as HS.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>0b - Normal operation
1b - Forced to full speed</td></tr>
<tr><td>23
PHCD</td><td>PHY Low-Power Suspend - Clock Disable (PLPSCD)
Disables the PHY clock.
Reading this field indicates the status of the PHY clock.
In Device mode, you can place the PHY into Low-Power Suspend mode when the device is not running (USBCMD[RS] is 0) or the host has signaled suspend (PORTSC1[SUSP] is 1). This field becomes 0 automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must write 0 to this field.
In Host mode, you can place the PHY into Low-Power Suspend mode when the downstream device is placed into Suspend mode or when no downstream device is connected.
Low-Power Suspend mode is completely under your control.
NOTE
You cannot disable the PHY clock if it is used as the system clock.
0b - Enables
1b - Disables</td></tr>
<tr><td>22
WKOC</td><td>Wake on Overcurrent Enable (WKOC)
Enables wake-up on overcurrent.
Writing 1 to this field enables the port to be sensitive to overcurrent conditions as wake-up events.
This field is 0 if PORTSC1[PP] is 0.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>USB0</td><td>PORTSC1</td><td>—</td></tr>
<tr><td>USB1</td><td>—</td><td>PORTSC1</td></tr>
</table>
</td></tr>
<tr><td>21
WKDC</td><td>Wake on Disconnect Enable (WKDSCNNT_E)
Enables wake-up on disconnect.
Writing 1 to this field enables the port to be sensitive to device disconnects as wake-up events.
This field is 0 if PORTSC1[PP] is 0 or is in Device mode.
0b - Disables
1b - Enables</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Function
20 WKCN	Wake on Connect Enable (WKCNNT_E) Enables port sensitivity. Writing 1 to this field enables the port to be sensitive to device connects as wake-up events. This field is 0 if PORTSC1[PP] is 0 or is in Device mode. 0b - Disables 1b - Enables
19-16 PTC	Port Test Control Defines port test control. See Port Test Control mode for the operational model used for these test modes, and chapter 7 of the USB Specification Revision 2.0 for details on each test mode. FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the FORCE_ENABLE_{HS/FS/LS} values to this field forces the port into the connected and enabled states at the selected speed. Writing the TEST_MODE_DISABLE value to this field allows the port state machines to progress normally from that point. **NOTE** LS operations are not supported as a peripheral device. Any value other than 0 indicates that the port is operating in Test mode. 0000b - TEST_MODE_DISABLE 0001b - J_STATE 0010b - K_STATE 0011b - SE0 (host) or NAK (device) 0100b - Packet 0101b - FORCE_ENABLE_HS 0110b - FORCE_ENABLE_FS 0111b - FORCE_ENABLE_LS 1000b-1111b - Reserved
15-14 PIC	Port Indicator Control Specifies the port indicator. Writing to this field has no effect if HCSPARAMS[PI] is 0. See the USB Specification Revision 2.0 for a description on how these fields must be used. This field is 0 if PORTSC1[PO] is 0. 00b - Port indicators are off 01b - Amber

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Green 11b - Undefined
13 PO	Port Owner Indicates port ownership. This field is used to release ownership of the port to a selected host controller (in the event that the attached device is not an HS device). This field unconditionally becomes 0 when the configured field in Configure Flag (CONFIGFLAG) makes a transition from 0 to 1. The field unconditionally becomes 1 whenever the configured field is 0. Write 1 to this field when the attached device is not an HS device. If this field is 1, it means that an internal companion controller owns and controls the port. Port owner handoff is not supported in all controller cores; therefore, this field is always 0.
12 PP	Port Power (PP) Works in accordance with HCSPARAMS[PPC]. In other words, the functioning of this field depends on the value of HCSPARAMS[PPC]. • PPC = 0b, PP = 0b (read-only): A controller in Device mode does not have port power control switches. • PPC = 1b, PP = 1b or 0b (read/write): A controller in Host mode requires port power control switches. This field represents the current setting of the switch (0 means off; 1 means on). When power is not available on a port (that is, PP = 0), the port is nonfunctional and does not report attachments, detachments, and so on. When an overcurrent condition is detected on a powered port and PPC is 1, this field, in each affected port, may be transitioned by the controller driver from 1 to 0 (removing power from the port). In Device mode, port power control is not necessary. Therefore, PPC and PP are 0. 0b - Off 1b - On
11-10 LS	Line Status Indicates the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In Host mode (unlike EHCI), the host controller driver may not necessarily use LineState because the port controller state machine and port routing manage the connection of LS and FS. In Device mode, the device controller driver may not necessarily use LineState. 00b - SE0 01b - K-state 10b - J-state 11b - Undefined

Table continues on the next page...

Table continued from the previous page...

Field	Function
9 HSP	High-Speed Port Indicates whether the host or device connected to the port is in HS mode. This field is redundant for PORTSC1[PSPD] but is included for compatibility. 0b - Not in HS mode 1b - In HS mode
8 PR	Port Reset Specifies whether the port is in reset. In Host mode (read/write), the bus-reset sequence (as defined in the USB Specification Revision 2.0) starts when you write 1 to this field. This field automatically changes to 0 after the reset sequence is complete. This behavior is different from that of EHCI where the host controller driver is required to write 0 to this field after the reset duration is timed in the driver. In Device mode, this is a read-only status field. Device reset from the USB bus is also indicated in USB Status (USBSTS). This field is 0 if PORTSC1[PP] is 0. 0b - Port not in reset 1b - Port in reset
7 SUSP	Suspend Specifies whether the port is in the Suspended state. In Host mode (read/write), this field works with PORTSC1[PEC] to define the port states as follows: If {PE, SUSP} = 0x, the port state is disabled. When in the Suspended state, downstream propagation of data is blocked on this port (except for port reset). Blocking occurs at the end of the current transaction if a transaction was in progress when this field was 1. The port is sensitive to resume detection in the Suspended state. The status of the field does not change until the port is suspended; there may be a delay in suspending a port if a transaction is in progress on USB. The host controller unconditionally writes 0 to this field when you write 0 to PORTSC1[FPR]. The host controller ignores a write of 0 to this field. If host software writes 1 to this field when the port is not enabled (that is, PORTSC1[PE] is 0), the results are undefined. This field is 0 if PORTSC1[PP] is 0 in Host mode. In Device mode, this is a read-only status field. 0b - Port not in Suspended state 1b - Port in Suspended state
6 FPR	Force Port Resume Specifies whether force resume is detected or driven on the port.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	In Host mode, software writes 1 to this field to drive resume signaling. The host controller writes 1 to this field if a J-to-K transition is detected when the port is in the Suspended state. When this field transitions to 1 because a J-to-K transition is detected, USBSTS[PCI] also becomes 1. This field automatically changes to 0 after the resume sequence is complete. This behavior is different from that of EHCI where the host controller driver is required to write 0 to this field after the resume duration is timed in the driver. When the host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (FS "K") is driven on the port until this field remains 1 (that is, until the port has switched to the HS idle state). Writing 0 to this field has no effect because the port controller times the resume operation, clears the field, and the port control state switches to HS or FS idle. This field is non-EHCI compatible and remains 0 if PORTSC1[PP] is 0 in Host mode. In Device mode, you must write 1 to this field to drive resume signaling after the device is in the Suspended state for 5 ms or more. The device controller writes 1 to this field if a J-to-K transition is detected when the port is in the Suspended state. The field becomes 0 when the device returns to normal operation. Additionally, when this field becomes 0 because of a K-to-J transition detection, USBSTS[PCI] also becomes 1. 0b - No resume (K-state) detected or driven on port 1b - Resume detected or driven on port
5 OCC	Overcurrent Change Flag Indicates whether an overcurrent change occurred. Hardware sets this field to 1 when there is a change in an overcurrent active condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No change occurred 1b - Change occurred When writing 0b - No effect 1b - Clear the flag
4 OCA	Overcurrent Active Indicates whether the port currently has an overcurrent condition. This field automatically transitions from 1 to 0 when the overcurrent condition is removed. 0b - No overcurrent condition exists 1b - Overcurrent condition exists
3	Port Enable and Disable Change Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
PEC	Indicates whether port enabled and disabled status has changed. In Host mode (for the root hub), this field is set to 1 only when a port is disabled because of disconnect on the port or because of appropriate conditions existing at the EOF2 point. See chapter 11 of the USB specification. This field is zero if PORTSC1[PP] is 0. In Device mode, the device port is always enabled; therefore, this field is always 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No change occurred 1b - Change occurred When writing 0b - No effect 1b - Clear the flag
2 PE	Port Enable and Disable Enables a port. In Host mode, this field is reset to 0b. Only the host controller can enable ports as part of reset and enable. You cannot enable a port by writing 1 to this field, and either a fault condition (such as a disconnect event) or host software can disable a port. The status of this field does not change until the port state actually changes. There may be a delay in disabling or enabling a port because of other host controller and bus events. If this field is 0 (the port is disabled), downstream propagation of data is blocked except for reset. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device mode, this field is always 1 (the device port is always enabled). 0b - Disables 1b - Enables
1 CSC	Connect Status Change Flag Indicates whether a change has occurred in the port's current connect status. In Host mode, the host or device controller sets this field for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the connect status (PORTSC1[CCS]) changes twice before you clear PORTSC1[CSC]. The hub hardware sets an already-set field (that is, CSC remains set). You can clear this field by writing 1 to it. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device Controller mode, this field is undefined.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No change occurred 1b - Change occurred When writing 0b - No effect 1b - Clear the flag
0 CCS	Current Connect Status Indicates the current port connection status. In Host mode, if this field is 1, the device is present on port, and if this field is 0 (default), no device is present. This value reflects the current state of the port, and may not correspond directly to the event that causes this field to be set. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device mode, if this field is 1, it indicates that the device is successfully attached and is operating in either HS or FS, as indicated by PORTSC1[HSP]. If this field is 0, it indicates that the device is not attached successfully or is forcibly disconnected by writing 0 to USBCMD[RS]. 0b - No device present or attached 1b - Device present and attached

51.8.1.33 On-The-Go Status and Control (OTGSC)

Offset

Register	Offset
OTGSC	1A4h

Function

Provides OTG status and control.

This register is available only in the OTG controller core and includes fields for:

- OTG interrupt enables
- OTG interrupt status
- OTG status inputs
- OTG controls

The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms do not cause an update of the status input register, or cause an OTG interrupt.

See USB Device Mode (USBMODE) for more information.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
USB0	OTGSC	—
USB1	—	OTGSC

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	DPIE	EN_1MS	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	Reserved	DPIS	STATUS_...	BSEIS	BSVIS	ASVIS	AVVIS	IDIS
W										W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	u	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	DPS	TOG_1MS	BSE	BSV	ASV	AVV	ID	HABA		IDPU	DP	OT	HAAR	VC	VD
W										HADP						
Reset	0	0	u	u	u	u	u	0	0	0	1	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 DPIE	Data Pulse Interrupt Enable Enables the data pulse interrupt. Writing 1 to the field enables the interrupt.
29 EN_1MS	1 Millisecond Timer Interrupt Enable Enables the 1 ms timer interrupt. Writing 1 to the field enables the interrupt.
28 BSEIE	B Session End Interrupt Enable Enables the B session end interrupt. Writing 1 to the field enables the interrupt.
27 BSVIE	B Session Valid Interrupt Enable Enables the B session valid interrupt. Writing 1 to the field enables the interrupt.
26 ASVIE	A Session Valid Interrupt Enable Enables the A session valid interrupt. Writing 1 to the field enables the interrupt.
25	A VBUS Valid Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
AVVIE	Enables the A VBUS valid interrupt. Writing 1 to the field enables the interrupt.
24 IDIE	USB ID Interrupt Enable Enables the USB ID interrupt. Writing 1 to the field enables the interrupt.
23 —	Reserved
22 DPIS	Data Pulse Interrupt Status Indicates the status of the data pulse interrupt. This field becomes 1 when data bus pulsing occurs on a DP or DM line. Data bus pulsing is detected only when USBMODE[CM] = 11b and PORTSC1[PP] = 0b.
21 STATUS_1MS	1 Millisecond Timer Interrupt Status Indicates the status of the timer interrupt. This field becomes 1 once every millisecond.
20 BSEIS	B Session End Interrupt Status Indicates the status of the B session end interrupt. This field becomes 1 when VBUS has fallen below the B session end threshold.
19 BSVIS	B Session Valid Interrupt Status Indicates the status of the B session valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the B session valid threshold.
18 ASVIS	A Session Valid Interrupt Status Indicates the status of the A session valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the A session valid threshold.
17 AVVIS	A VBUS Valid Interrupt Status Indicates the status of the A VBUS valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the VBUS valid threshold on A device.
16 IDIS	USB ID Interrupt Status Indicates the status of the USB ID interrupt. This field becomes 1 when a change on the ID input is detected.
15 —	Reserved
14	Data Bus Pulsing Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
DPS	Indicates the status of data bus pulsing. If this field is 1, it indicates that data bus pulsing is detected on the port.
13 TOG_1MS	1 Millisecond Timer Toggle Toggles once per millisecond.
12 BSE	B Session End Indicates that VBUS is below the B session end threshold.
11 BSV	B Session Valid Indicates that VBUS is above the B session valid threshold.
10 ASV	A Session Valid Indicates that VBUS is above the A session valid threshold.
9 AVV	A VBUS Valid Indicates that VBUS is above the A VBUS valid threshold.
8 ID	USB ID Indicates the USB ID. 0b - A device 1b - B device
7 HABA	Hardware Assist B-Disconnect to A-connect Enables hardware assist for B-disconnect to A-connect. During HNP, the B-disconnect occurs from the OTG-A Suspended state and within 3 ms, the A-device must enable the pullup on the DP leg in the A-peripheral state. When this field is 1, the host controller port is in Suspend mode, and the device disconnects. Then this hardware assist begins. 0b - Disables 1b - Enables
6 HADP	Hardware Assist Data Pulse Enables starting a data pulse. Writing 1 to this field starts a data pulse of approximately 7 ms in duration and then automatically ceases the data pulsing. During the data pulse, OTGSC[DP] becomes 1 and then becomes 0. This automation relieves software from accurately controlling the data-pulse duration. During the data pulse, HCD can poll to see that the HADP and DP fields have become 0 to recognize the completion of the data pulse, or launch the data pulse and wait to see if a VBUS valid interrupt occurs when the A-side supplies bus power. This field ensures data pulsing meets the OTG requirement of > 5 ms and < 10 ms. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
5 IDPU	ID Pullup Provides control over the ID pull-up resistor. If this field is 0, the functionality is off and the ID input is not sampled. If this field is 1 (default), the functionality is on and the ID input is sampled.
4 DP	Data Pulsing Controls pullup on the DP line. Writing 1 to this field causes the pullup on the DP line to be asserted for data pulsing during SRP.
3 OT	OTG Termination Controls pulldown on the DM line. This field must be 1 when the OTG device is in Device mode to control the pulldown on the DM line.
2 HAAR	Hardware Assist Auto Reset Enables automatic reset of hardware after a connect event on host port. When this field is 1, the host automatically starts a reset after a connect event. This is a shortcut to the normal process through which you are notified of the connect event to start the reset. Even after receiving a notification of the connect event, you must not write to the reset field when HAAR is 1. You are again notified after the reset is complete via PORTSC1[PTS_1] that causes a port change interrupt. 0b - Disables 1b - Enables
1 VC	VBUS Charge Controls VBUS pulsing during SRP. Writing 1 to this field causes the VBUS line to be charged.
0 VD	VBUS Discharge Discharges VBUS through a resistor. Writing 1 to this field causes VBUS to discharge through a resistor.

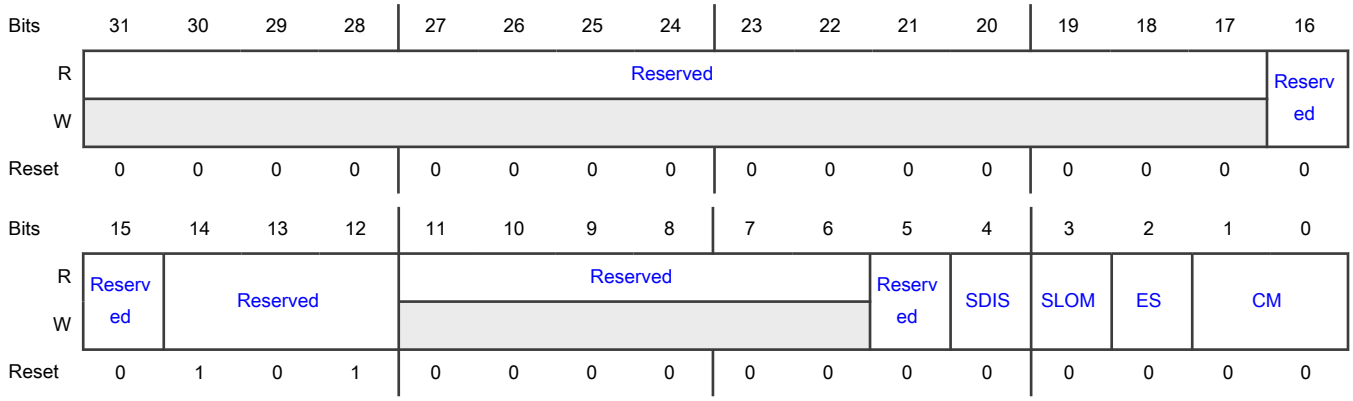
51.8.1.34 USB Device Mode (USBMODE)

Offset

Register	Offset
USBMODE	1A8h

Function
Selects the USB device mode.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15 —	Reserved
14-12 —	Reserved
11-6 —	Reserved
5 —	Reserved
4 SDIS	Stream Disable Mode Disables double priming for low-bandwidth systems. In Device mode, writing 1 to this field disables double priming on both RX and TX buffers for low-bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet, the standard double buffering scheme is disabled to prevent overruns and underruns in bandwidth limited systems. **NOTE** In HS mode, all packets received are responded to with an NYET handshake when stream disable is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	In Host mode, writing 1 to this field ensures that overruns and underruns of the latency FIFO are eliminated for low-bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also ensures that the TX latency is filled to capacity before the packet is launched onto the USB. **NOTE** The time duration to prefill the FIFO becomes significant when stream disable is active. See TX FIFO Fill Tuning (TXFILLTUNING) to characterize the adjustments needed for the scheduler when using this feature. The use of this feature substantially limits the overall USB performance that can be achieved. 0b - Disables 1b - Enables
3 SLOM	Setup Lockout Mode Specifies whether the setup lockouts are on. In Device mode, this field controls the behavior of the setup lock mechanism. See Control endpoint operation model for more information. The DCD requires you to use the setup data buffer tripwire in USB Command (USBCMD). 0b - On (default) 1b - Off
2 ES	Endian Select Selects the endianness. You can use this field to change the byte alignment of the transfer buffers to match the host microprocessor. The fields in the microprocessor interface and the data structures are unaffected by the value of this field because they are based on the 32-bit word. 0b - Little endian (default) 1b - Big endian
1-0 CM	Controller Mode Specifies the controller mode. The controller mode is defaulted to the proper mode for host-only and device-only implementations. For those designs that contain both host and device capabilities, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host and device controllers, you can write to this register only once after reset. If it is necessary to switch modes, you must reset the controller by writing to USBCMD[RST] before reprogramming this register. 00b - Idle (default for host and device combination) 01b - Reserved 10b - Device controller (default for device-only controller) 11b - Host controller (default for host-only controller)

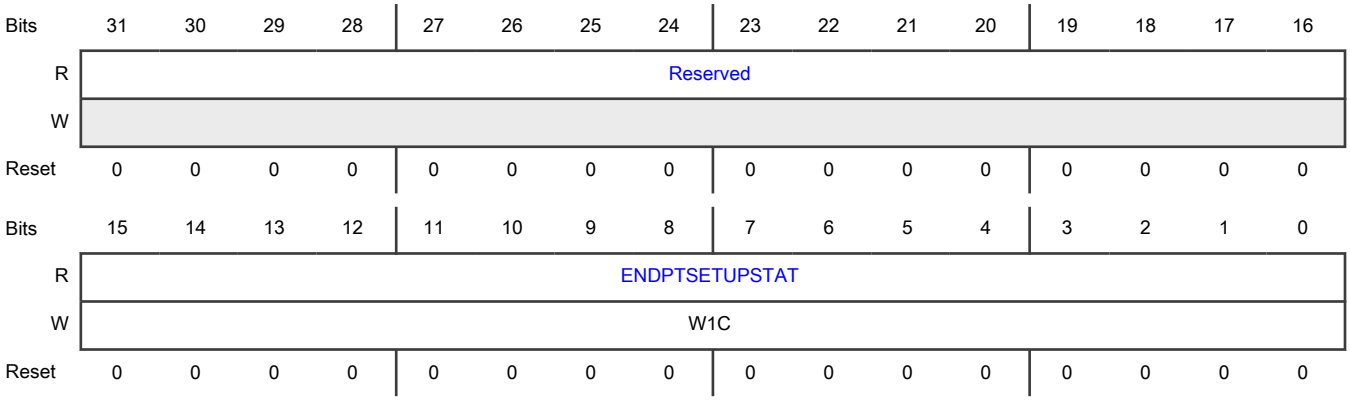
51.8.1.35 Endpoint Setup Status (ENDPTSETUPSTAT)

Offset

Register	Offset
ENDPTSETUPSTAT	1ACh

Function
Indicates the endpoint setup status.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 ENDPTSETUPSTAT	Endpoint Setup Status Flag Indicates the endpoint setup status and whether the setup transaction is received. For every setup transaction that is received, a corresponding bit in this field becomes 1. You must clear or acknowledge the setup transfer by writing 1 to the respective bit after it has read the setup data from the queue head. The response to a setup packet includes many operations in an order. The total response time is crucial to limit bus timeouts if the setup lockout mechanism is engaged. See Managing endpoints in the device operational model. This register is used only in Device mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000_0000_0000b - Not received

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000_0000_0000_0001b - Received When writing 0000_0000_0000_0000b - No effect 0000_0000_0000_0001b - Clear the flag

51.8.1.36 Endpoint Prime (ENDPTPRIME)

Offset

Register	Offset
ENDPTPRIME	1B0h

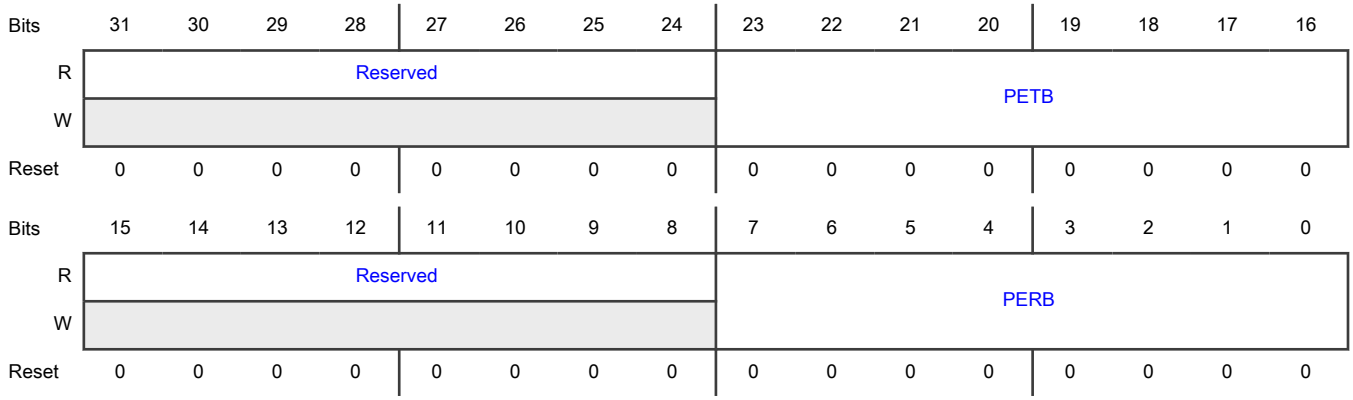
Function

Primes the endpoint.

When you set the prime bit for a given endpoint, the device controller loads the transfer descriptor, pointed to by the queue head, so that the endpoint is ready to transmit or receive when the host sends a request ("In" or "Out" token). The endpoint NAKs all requests from the host until the endpoint is primed. The controller automatically reprimes the endpoint with a new transfer descriptor when one is found via the next_dtd pointer of the current transfer descriptor. Therefore, you must set the prime bit only when a descriptor is added to the queue head.

This register is only used in Device mode.

Diagram



Fields

Field	Function
31-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
23-16 PETB	Prime Endpoint Transmit Buffer Parses a new transfer descriptor from the queue head and prepares for a transmit buffer. For each endpoint, a corresponding bit is used to request that a buffer is prepared for a transmit operation to respond to a USB "In" or "Interrupt" transaction. You must write 1 to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this field to begin parsing for a new transfer descriptor from the queue head and preparing a transmit buffer. Hardware writes 0 to this field when an associated endpoint is successfully primed. PETB[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily writes 1 to this field during hardware repriming operations when a dTD is retired and the dQH is updated.
15-8 —	Reserved
7-0 PERB	Prime Endpoint Receive Buffer Parses a new transfer descriptor from the queue head and prepares for a receive buffer. For each endpoint, a corresponding bit is used to request that a buffer is prepared for a receive operation when a USB host initiates a USB "Out" transaction. You must write 1 to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this field to begin parsing for a new transfer descriptor from the queue head and preparing a receive buffer. Hardware writes 0 to this field when an associated endpoint is successfully primed. PERB[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily writes 1 to this field during hardware repriming operations when a dTD is retired and the dQH is updated.

51.8.1.37 Endpoint Flush (ENDPTFLUSH)

Offset

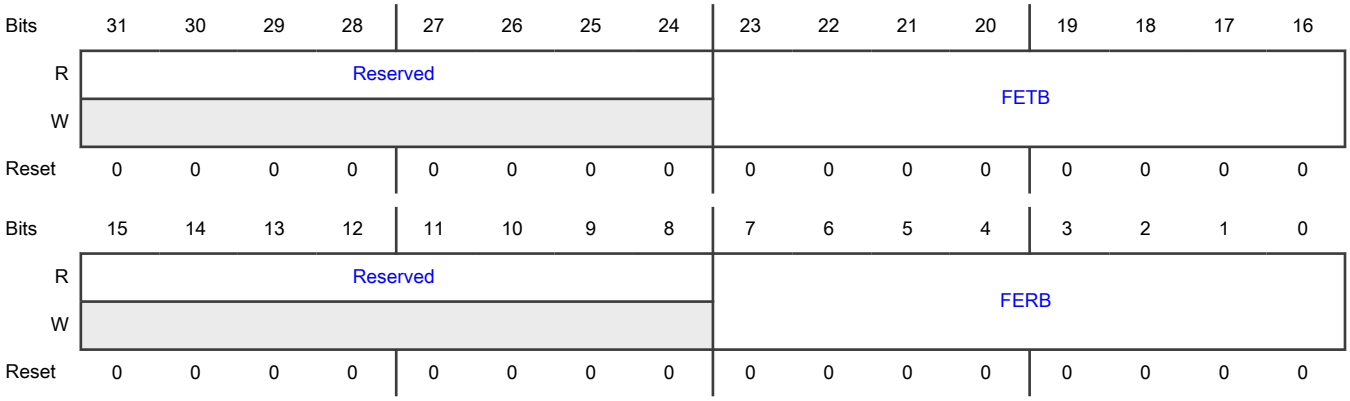
Register	Offset
ENDPTFLUSH	1B4h

Function

Flushes the endpoint.

This register is only used in Device mode.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 FETB	Flush Endpoint Transmit Buffer Clears any primed transmit buffers. When you write 1 to this field, it causes the associated endpoint(s) to clear any primed transfer buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[*n*]: Endpoint [*n*], *n* is in 0–7
15-8 —	Reserved
7-0 FERB	Flush Endpoint Receive Buffer Clears any primed receive buffers. When you write 1 to this field, it causes the associated endpoints to clear any primed receive buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[*n*]: Endpoint [*n*], *n* is in 0–7

51.8.1.38 Endpoint Status (ENDPTSTAT)

Offset

Register	Offset
ENDPTSTAT	1B8h

Function

Indicates endpoint transmit and receive buffer ready status.

This register is only used in Device mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								ETBR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERBR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-16 ETBR	Endpoint Transmit Buffer Ready Indicates the endpoint transmit buffer ready status. One bit for each endpoint indicates the status of the respective endpoint buffer. Hardware writes 1 to this field as a response to receiving a command from a corresponding field in Endpoint Prime (ENDPTPRIME). There is always a delay between writing 1 to the corresponding field in Endpoint Prime (ENDPTPRIME) and the endpoint indicating the ready status. This delay time varies based on the current USB traffic and the number of fields with value 1 in Endpoint Prime (ENDPTPRIME). This field becomes 0 via the USB reset, USB DMA system, or through Endpoint Flush (ENDPTFLUSH) ETBR[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily clears this field during hardware endpoint repriming operations when a dTD is retired and the dQH is updated.
15-8 —	Reserved
7-0 ERBR	Endpoint Receive Buffer Ready Indicates the endpoint receive buffer ready status. One bit for each endpoint indicates the status of the respective endpoint buffer. Hardware writes 1 to this field as a response to receiving a command from a corresponding field in Endpoint Prime (ENDPTPRIME). There is always a delay between writing 1 to the corresponding field in Endpoint Prime (ENDPTPRIME) and the endpoint indicating the ready status. This delay time varies based on the current USB traffic and the number of fields with value 1 in Endpoint Prime (ENDPTPRIME). This field becomes 0 via the USB reset, USB DMA system, or through Endpoint Flush (ENDPTFLUSH).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	ERBR[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily clears this field during hardware endpoint repriming operations when a dTD is retired and the dQH is updated.

51.8.1.39 Endpoint Complete (ENDPTCOMPLETE)

Offset

Register	Offset
ENDPTCOMPLETE	1BCh

Function

Indicates the endpoint receive and transmit complete event status.

This register is only used in Device mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								ETCE							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERCE							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-16 ETCE	Endpoint Transmit Complete Event Indicates whether a transmit event (Out or Setup) occurred. You must read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the transfer descriptor, then this field is set along with USBSTS[UI].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	ETCE[*n*]: Endpoint [*n*], *n* is in 0–7 0000_0000b - Transmit incomplete 0000_0001b - Transmit complete
15-8 —	Reserved
7-0 ERCE	Endpoint Receive Complete Event Indicates whether a received event (Out or Setup) occurred. You must read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the transfer descriptor, then this field is set along with USBSTS[UI]. ERCE[*n*]: Endpoint [*n*], *n* is in 0–7 0000_0000b - Transmit incomplete 0000_0001b - Transmit complete

51.8.1.40 Endpoint Control 0 (ENDPTCTRL0)

Offset

Register	Offset
ENDPTCTRL0	1C0h

Function

Forces the endpoint to return a STALL handshake to the host. Every device uses this register as a control endpoint.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE	Reserved			TXT		Reserved	TXS
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE	Reserved			RXT		Reserved	RXS
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22-20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. If this field is 0, endpoint 0 is fixed as a control endpoint.
17 —	Reserved
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If you write 1 to this field, it forces the endpoint to return a STALL handshake to the host. The field continues returning a STALL response until you write 0 to it or if the field automatically becomes 0 after it receives a new Setup request. After receiving a Setup request, hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) is cleared. **NOTE** There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and the hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1 or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. Endpoint 0 is always enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
6-4 —	Reserved
3-2 RXT	RX Endpoint Type Indicates the receive endpoint type. If this field is 0, endpoint 0 is fixed as a control endpoint.
1 —	Reserved
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If you write 1 to this field, it forces the endpoint to return a STALL handshake to the host. The field continues returning a STALL response until you write 0 to it or if the field automatically becomes 0 after it receives a new Setup request. After receiving a Setup request, hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) is cleared. **NOTE** There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and the hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.41 Endpoint Control 1 (ENDPTCTRL1)

Offset

Register	Offset
ENDPTCTRL1	1C4h

Function

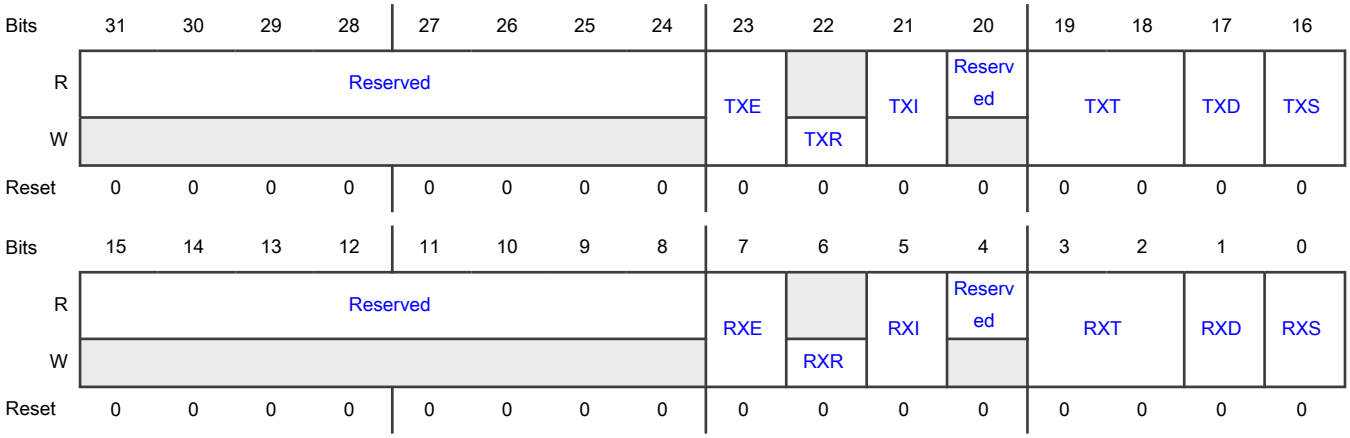
Enables endpoint control 1.

This register is used for endpoint 1 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Allow 1b - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a "Setup" request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum of 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow 1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.42 Endpoint Control 2 (ENDPTCTRL2)

Offset

Register	Offset
ENDPTCTRL2	1C8h

Function

Enables endpoint control 2.

This register is used for endpoint 2 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE		TXI	Reserved	TXT		TXD	TXS
W										TXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE		RXI	Reserved	RXT		RXD	RXS
W										RXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to it causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables receive endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.43 Endpoint Control 3 (ENDPTCTRL3)

Offset

Register	Offset
ENDPTCTRL3	1CCh

Function

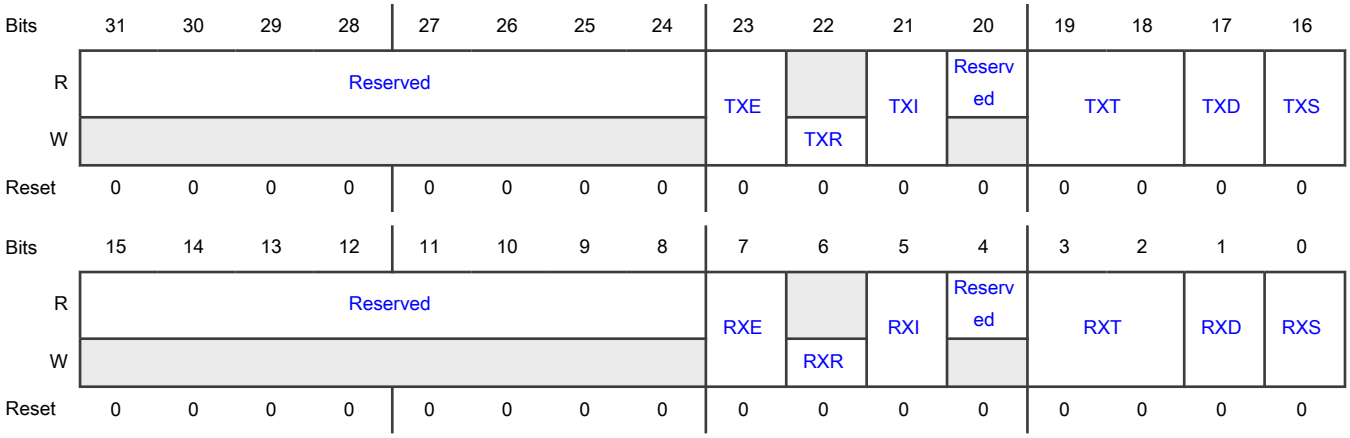
Enables endpoint control 3.

This register is used for endpoint 3 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21	TX Data Toggle Inhibit

Table continues on the next page...

Table continued from the previous page...

Field	Function
TXI	Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow 1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.44 Endpoint Control 4 (ENDPTCTRL4)

Offset

Register	Offset
ENDPTCTRL4	1D0h

Function

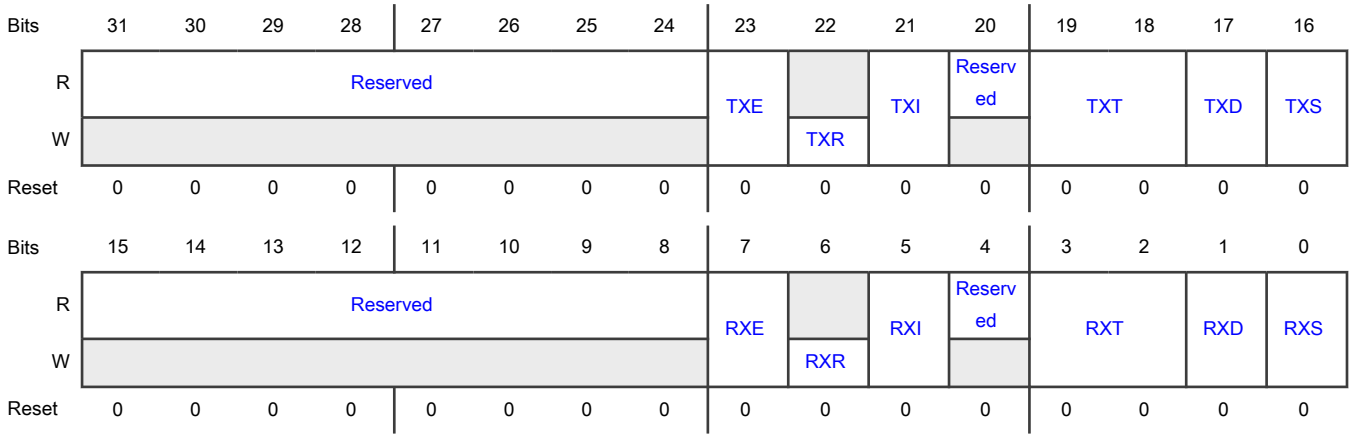
Enables endpoint control 4.

This register is used for endpoint 4 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to this field. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. An must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6	RX Data Toggle Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
RXR	Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to it causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Disables 1b - Enables
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Control endpoint only: There is a slight delay (of maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.45 Endpoint Control 5 (ENDPTCTRL5)

Offset

Register	Offset
ENDPTCTRL5	1D4h

Function

Enables endpoint control 5.

This register is used for endpoint 5 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE		TXI	Reserved	TXT		TXD	TXS
W										TXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE		RXI	Reserved	RXT		RXD	RXS
W										RXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to this field. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.46 Endpoint Control 6 (ENDPTCTRL6)

Offset

Register	Offset
ENDPTCTRL6	1D8h

Function

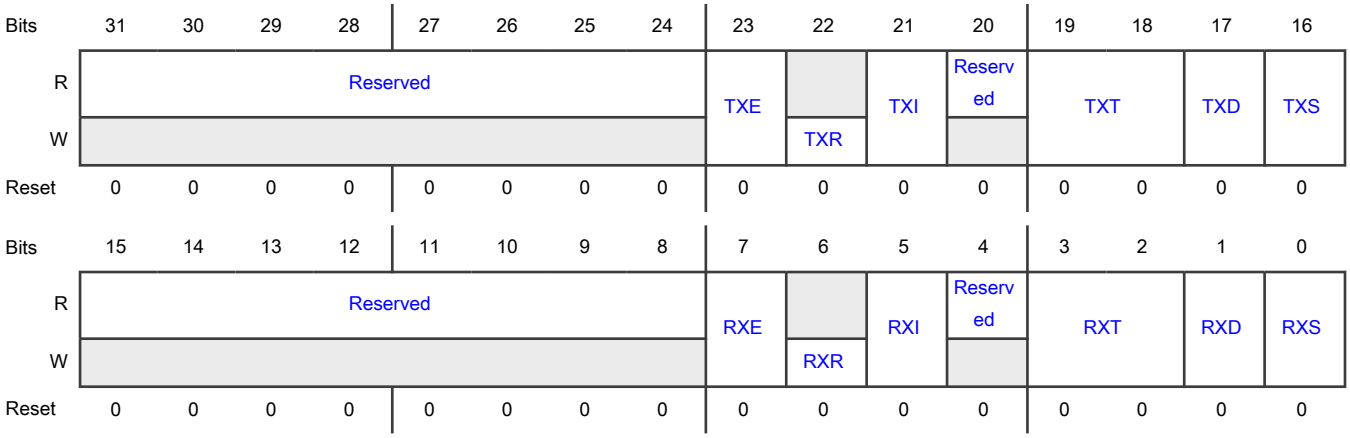
Enables endpoint control 6.

This register is used for endpoint 6 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21	TX Data Toggle Inhibit

Table continues on the next page...

Table continued from the previous page...

Field	Function
TXI	Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow 1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0	RX Endpoint Stall

Table continues on the next page...

Table continued from the previous page...

Field	Function
RXS	Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

51.8.1.47 Endpoint Control 7 (ENDPTCTRL7)

Offset

Register	Offset
ENDPTCTRL7	1DCh

Function

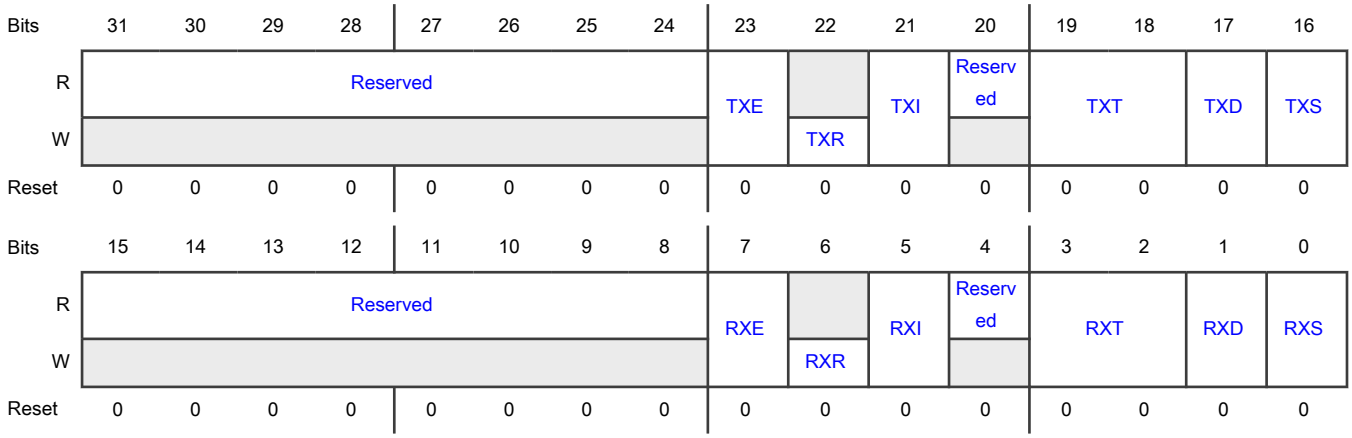
Enables endpoint control 7.

This register is used for endpoint 7 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is only used for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0b - Allow 1b - Inhibit
20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0b - Disables 1b - Enables
6	RX Data Toggle Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
RXR	Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0b - Does not reset 1b - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0b - Allow 1b - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00b - Control 01b - Isochronous 10b - Bulk 11b - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0b - Endpoint OK 1b - Endpoint stalled

Chapter 52
Universal Serial Bus 2.0 PHY (USBPHY)

52.1 Chip-specific USBPHY information

Table 503. Reference links to related information

Topic	Related module	Reference
Full description	USBPHY	USBPHY
USB	USB	USB
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

52.1.1 Module instances

This device contains one instance of the USBPHY module.

Table 504. USBPHY instance

	USBPHY
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
Related USB	USB0

52.1.2 USBPHY0 clock configuration

You can use CLKCTL4.PSCCTL1_MEDIA[USB0] to enable or disable clocks to the USBPHY0. You should also configure the SYSCON4.USBPHY0_CLK_ACTIVE[IPG_CLK_ACTIVE] as below to make the USBPHY0 work normally.

When the USBPHY0 clock is enabled, the SYSCON4.USBPHY0_CLK_ACTIVE[IPG_CLK_ACTIVE] must be configured as 1b (IPG_CLK is active).

When the USBPHY0 clock is gated, the SYSCON4.USBPHY0_CLK_ACTIVE[IPG_CLK_ACTIVE] must be configured as 0b (IPG_CLK is inactive).

52.2 Overview

USBPHY interfaces with the USB host and device system at a low-speed (LS) rate of 1.5 Mbit/s, full-speed (FS) rate of 12 Mbit/s, or USB 2.0 high-speed (HS) rate of 480 Mbit/s. The following sections describe the external and internal interfaces, major blocks, and programmable registers that comprise the integrated USBPHY.

52.2.1 Block diagram

The integrated PHY provides a standard USB transceiver macrocell interface (UTMI). The DP and DN pins (as shown in the following figure) connect directly to a USB connector.

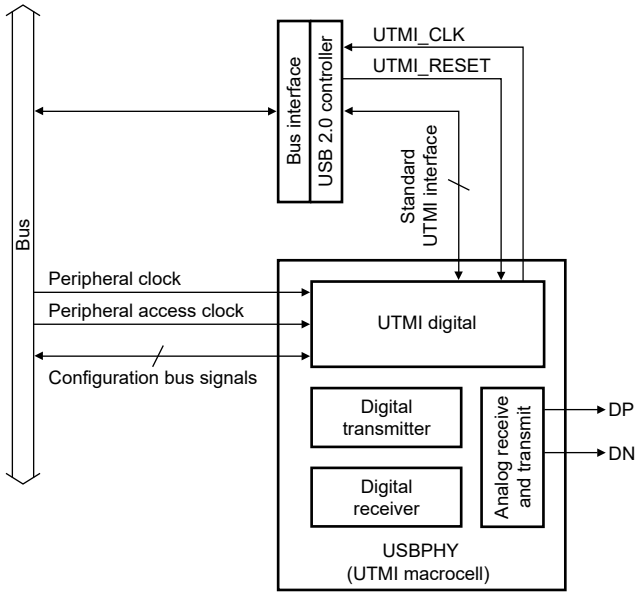


Figure 178. Block diagram

52.2.2 Features

- Supports HS and FS in Device mode
- Supports HS, FS, and LS in Host mode
- Complies with the UTMI+ standard, having a 16-bit data interface to the USB controller
- Provides an integrated 480 MHz PLL
- Provides an integrated USBDCD (see the "USB Device Charger Detection Module (USBDCD)" chapter) with features of *USB Battery Charging Specification, Revision 1.2*

52.3 Functional description

This section covers the USBPHY core components.

- The digital portions of USBPHY include the UTMI, digital transmitter, digital receiver, and programmable registers.
- The analog transceiver comprises an analog receiver and analog transmitter, as shown in Figure 179.

52.3.1 UTMI

The UTMI block handles the LineState bits, reset buffering, suspend distribution, transceiver speed selection, transceiver termination selection, and clock generation logic for the clocks derived from the PHY's USB PLL.

The PLL in USBPHY supplies a 480 MHz clock to the UTMI digital block that divides the clock to generate the 30 MHz clock (UTMI_CLK) used in the interface.

52.3.2 Digital transmitter

The digital transmitter receives a 16-bit transmit data from the USB controller and handles the TX_VALID, TX_VALIDH, and TX_READY handshake. Additionally, it contains the transmit serializer that converts 16-bit parallel words, at a frequency of 30

MHz, to a single bitstream at 480 Mbit/s for HS, 12 Mbit/s for FS, or 1.5 Mbit/s for LS. The digital transmitter does this while implementing the bit-stuffing algorithm and NRZI encoder that you use to remove the DC component from the serial bitstream. The output of this encoder is sent to the LS, FS, or HS drivers in the analog transceiver section's transmitter block.

52.3.3 Digital receiver

The digital receiver receives the raw serial bitstream from the LS differential transceiver, FS differential transceiver, and 480 MHz sampled data from the HS differential transceiver. As the phase of the USB host transmitter shifts relative to the local PLL, the receiver section's HS DLL tracks these changes to give an appropriate sample of the incoming 480 Mbit/s bitstream. Because this sample point shifts relative to the PLL phase that the digital logic uses, a rate-matching elastic buffer is provided to cross this clock domain boundary.

After the bitstream is in the local clock domain, an NRZI decoder and a bit unstuffer restore the original payload data bitstream and pass it to a deserializer and holding register. The RX state machine handles the RX_READY and RX_VALIDH signals, and a handshake with the USB controller. The handshake is not interlocked, with no RX_READY signal coming from the controller, which must take each 16-bit value that the PHY presents. The RX state machine provides an RX_ACTIVE signal to the controller that indicates whether it is inside a valid packet (synchronization detected, for example).

52.3.4 Analog receiver

The following figure illustrates the analog receiver that comprises differential receivers—two single-ended ones, a 9X, and a 480 MHz HS data sampling module.

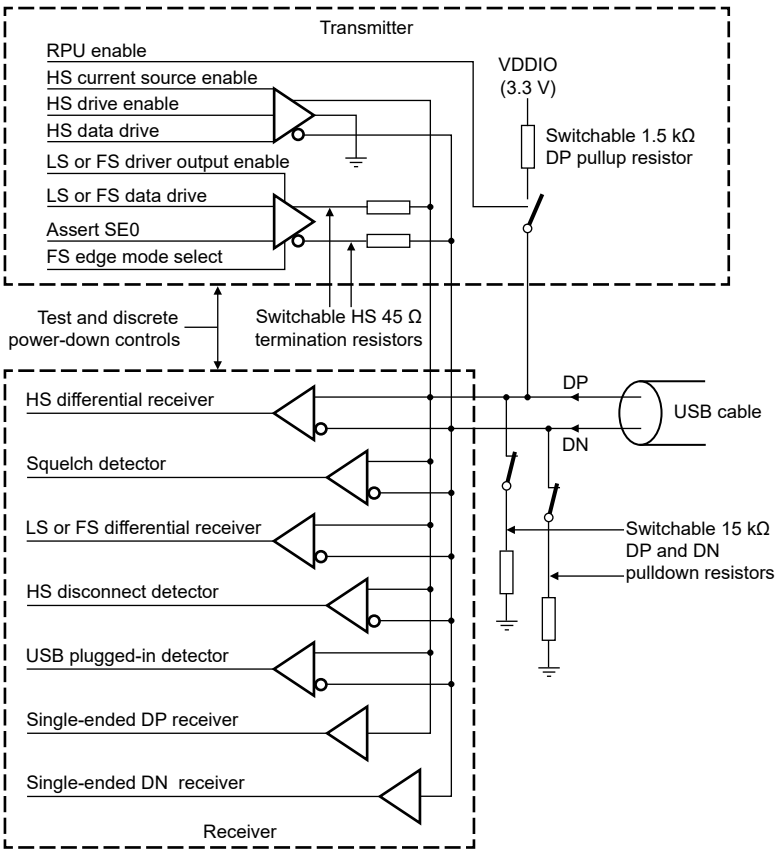


Figure 179. Analog transceiver block diagram

52.3.4.1 HS differential receiver

The HS differential receiver is both a differential analog receiver and threshold comparator. Its output is high (1b) if the differential signal is greater than a 0 V threshold. Otherwise, its output is low (0b). The purpose of the HS differential receiver is to discriminate the ± 400 mV differential voltage resulting from the HS drivers' current flow into the dual 45 Ω terminations found on each pin of the differential pair. The envelope or squelch detector (see Squelch detector) ensures that the differential signal has sufficient magnitude to be valid. The HS differential receiver tolerates up to 500 mV of common mode offset.

52.3.4.2 Squelch detector

The squelch detector is a differential analog receiver and threshold comparator. Its output is high (1b) if the differential magnitude is less than a nominal 100 mV threshold. Otherwise, its output is low (0b). The purpose of the squelch detector is to invalidate the HS differential receiver if the incoming signal is too low for appropriate reception.

52.3.4.3 LS or FS differential receiver

The LS or FS differential receiver is both a differential analog receiver and threshold comparator. The crossover voltage lies between 1.3 V and 2.0 V. Its output is 1 if the DP line is above the crossover point and the DN line is below the crossover point. The digital receiver section decodes the receiver data into the J or K state, according to the speed.

52.3.4.4 HS disconnect detector

The HS disconnect detector is a differential analog receiver and threshold comparator. Its output is high when the differential magnitude is greater than a nominal 575 mV threshold. Otherwise, its output is low.

52.3.4.5 USB plugged-in detector

When operating in Device mode, USBPHY provides the following methods for the local USB device to detect the attachment of a cable to a remote USB host or hub. You must deploy only one of these methods at a time when waiting to identify the cable attachment event, and disable all methods during USB data communication:

- USB plugged-in detector: This method looks for both DP and DN to be high. A large pair of on-chip pullup resistors (200 kΩ) hold both the DP and DN pins high when the USB cable is not attached. The USB plugged-in detector signals a 0 in this case. When operating in Device mode, the upstream port in the host or hub interface contains a 15 kΩ pulldown resistor that could easily override the 200 kΩ pullup resistor. When plugged in, at least one signal in the pair is low, which forces the plugged-in detector's output to become high.

  CTRL[ENDEVPLUGINDETECT] controls the USB plugged-in detector, and you can observe the results through STATUS[DEVPLUGIN_STATUS].

- Data contact detect: This method functions according to *USB Battery Charging Specification, Revision 1.2*. See the "USB Device Charger Detection (USBDCD)" chapter that describes the usage of this method.

52.3.4.6 Single-ended DP receiver

The single-ended DP receiver output is high whenever the DP input is above its nominal 1.8 V threshold.

52.3.4.7 Single-ended DN receiver

The single-ended DN receiver output is high whenever the DN input is above its nominal 1.8 V threshold.

52.3.5 Analog transmitter

The analog transmitter comprises two differential drivers: one for HS signaling and another for FS signaling. It also contains the switchable 1.5 kΩ pullup resistor (see Figure 180).

52.3.5.1 Switchable HS 45 Ω termination resistors

The HS current mode differential signaling requires a 90 Ω differential termination at each end of the USB cable. This results from switching in 45 Ω terminating resistors from each signal line to ground at each end of the cable.

Because each signal is terminated in parallel with 45 Ω at each end, each driver sees a 22.5 Ω load. This load impedance is too low for FS signaling levels, generating the need for switchable HS terminating resistors. Switchable trimming resistors are provided to tune the actual termination resistance for each device, as shown in Figure 180.

TX[TXCAL45DP], for example, allows one of the trimming resistor values to be placed in parallel with the 45 Ω terminator on the DP signal.

52.3.5.2 LS or FS differential driver

The LS or FS differential drivers are essentially a pair of low-impedance pullup and pulldown devices that are switched in a differential mode for most LS or FS signaling. One output is driven high while the other output is driven low to signal the "J" state or the "K" state. Both outputs are driven low for the LS or FS "SE0" state.

52.3.5.3 HS differential driver

The HS differential driver receives a 17.78 mA current from the constant current source (IREF) and essentially steers it down either the DP signal or DN signal, or alternatively to the ground. This current produces an approximately 400 mV drop across the 22.5 Ω termination that the driver sees when it is steered onto one of the signal lines. The approximately 17.78 mA current source is referenced back to the integrated voltage-band-gap (Vbg) circuit.

52.3.5.4 Switchable 1.5 kΩ DP pullup resistor

This chip contains a switchable 1.5 kΩ pullup resistor on the DP signal. You switch the register on to indicate to the host or hub controller that an FS-capable device is on the USB cable, powered on, and ready. You switch this resistor off at POR so that the host does not recognize a USB device until the processor software enables the announcement of an FS device.

52.3.5.5 Switchable 15 kΩ DP and DN pulldown resistors

This chip contains switchable 15 kΩ pulldown resistors on both DP and DN signals. You enable them in Host mode to indicate to the device controller that a host is present.

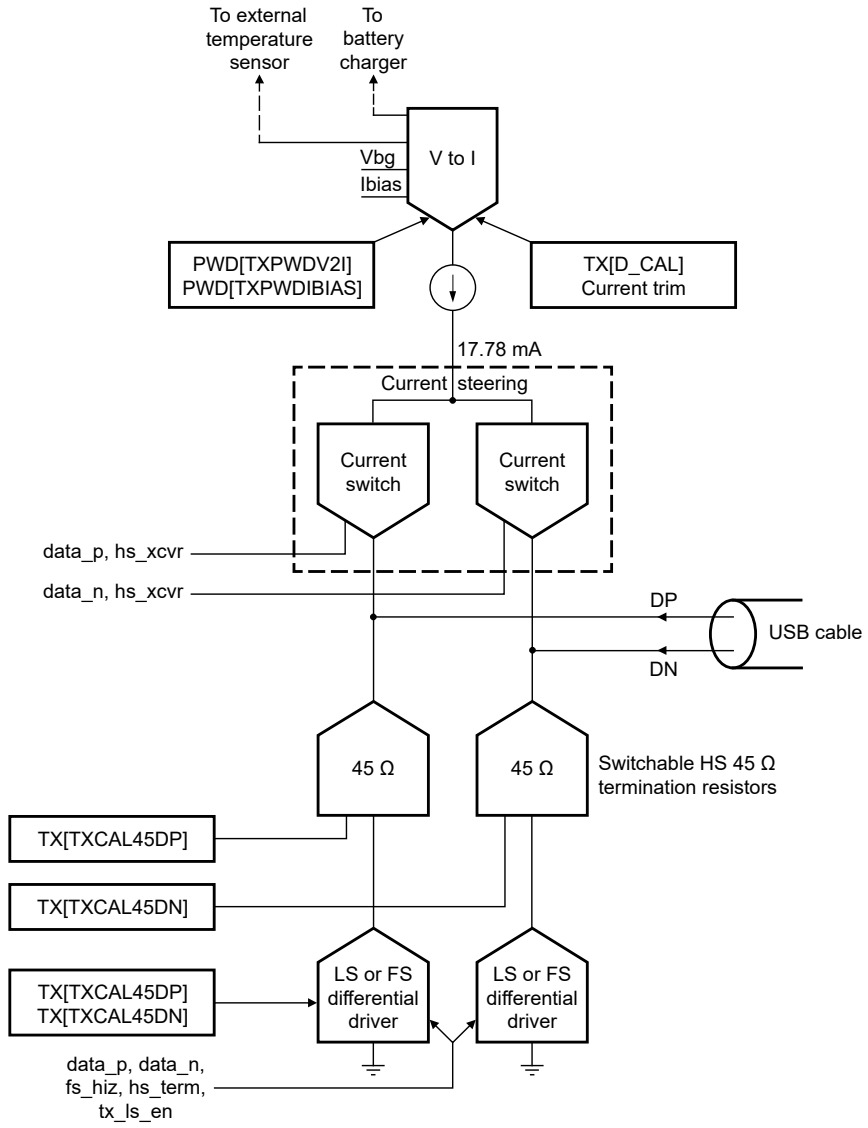


Figure 180. USBPHY transmitter block diagram

52.3.6 Low-Power mode

USBPHY includes a low-power mode (Suspend mode) to save power consumption.

When the USB bus is idle, you can place the transceiver in Low-Power mode (suspend it), after which the clocks to USBPHY can be stopped. The 32 kHz low-power clock must remain active because you need it for wake-up detection.

The USB HS controller monitors USBPHY's entry and exit into Low-Power mode:

- If USB.PORTSC1[SUSP] is 1, the UTMI interface is driven to the FS or LS state.
- If USB.PORTSC1[PHCD] is 1, the UTMI signal "SuspendM" goes low.
- If CTRL[ENAUTOCLR_PHY_PWD] and CTRL[ENAUTOCLR_CLKGATE] are 1, the UTMI clock and PHY transmitter are automatically enabled when the "SuspendM" signal goes high.

- The SW stack must write 1b to PLL_MISC2_CONTROL0 during the USB initialization routine. If PLL_MISC2_CONTROL0 is set, the PLL is auto powered down when in Suspend mode. The PLL is auto enabled after the "SuspendM" signal goes high, and after PLL is locked, USBPHY uses the PLL clock again.

USBPHY enters Low-Power mode under the control of the driver software. See the "Universal Serial Bus Controller (USB)" chapter for details.

52.3.7 Clocking

The following table describes the clock sources for USBPHY. See the chip's clocking chapter for clock setting, configuration, and gating information.

Table 505. USBPHY clocks

Clock name	Description
Peripheral clock	Free running internal register bus (IPS) clock for register configuration
Peripheral access clock	Gated IPS bus clock

52.3.8 Reset

Table 506. USBPHY resets

Reset	How it works
Chip	USBPHY's logic and registers are reset to their default states.
UTMI	UTMI_RESET signal from the USB controller resets the UTMI state machines, FIFO, and so on.
Software	A soft reset (SFTRST) of USBPHY's USBPHY Powerdown (PWD), USBPHY Transmitter Control (TX), USBPHY Receiver Control (RX), USBPHY General Control (CTRL), DEBUG0_STATUS[UTMI_RXERROR_FAIL_COUNT], USBPHY Debug (DEBUG), and UTMI Debug Status 1 (DEBUG1) registers and fields takes place when you write 1 to CTRL[SFTRST]. This also resets state machines in the PHY. Write 0 to CTRL[SFTRST] to release the PHY from reset. **NOTE** A soft reset can take multiple clock periods to complete; therefore, do not write 1 to CTRL[CLKGATE] when writing 1 to CTRL[SFTRST]. The reset process gates the clocks automatically.

52.3.9 Interrupts

Table 507. USBPHY interrupts

Flag or event	Interrupt enable	Description
CTRL[DEVPLUGIN_IRQ]	CTRL[ENIRQDEVPLUGIN]	Indicates nonstandard resistive plugged-in detection

Table continues on the next page...

Table 507. USBPHY interrupts (continued)

Flag or event	Interrupt enable	Description
CTRL[RESUME_IRQ]	CTRL[ENIRQRESUMEDETECT]	Indicates that the host is sending a "resume" signal after a "suspend" signal
CTRL[WAKEUP_IRQ]	CTRL[ENIRQWAKEUP]	Indicates that the USB controller has stopped driving the "SuspendM" signal in its active-low state to the PHY
CTRL[HOSTDISCONDETECT_IRQ]	CTRL[ENIRQHOSTDISCON]	Indicates that the host has detected disconnection of the device in HS mode
USBDCD interrupt	USBDCD interrupt enable	See the "Interrupts" section in the "USB Device Charger Detection Module (USBDCD)" chapter for details

52.4 External signals

Table 508. USBPHY external signals

Signal	Description	Direction
DP	Data positive (D+) pin A didirectional pin that carries USB data to and from USBPHY	Input and output
DN	Data negative (D-) pin A bidirectional pin that carries USB data to and from USBPHY	Input and output
VBUS	USB power supply pin A charge pump external to USBPHY provides power to this pin.	Power

52.5 Initialization

Perform the following procedure to initialize the internal 480 MHz USB PLL clock:

1. Enable the reference clock for USB PLL.
2. Enable the USBPHY PLL regulator.
3. Release the PHY from reset.
4. Power up the PLL.
5. Configure PLL_SIC[PLL_DIV_SEL] appropriately, according to the selected external reference. The system oscillator ranges from 16 MHz to 32 MHz.

52.6 Application information

52.6.1 Recommended register configuration for USB certification

USBPHY has programmability to adjust several transceiver parameters. You can:

- Adjust the TX HS driver termination impedance, FS driver output impedance, and HS driver output currents by using USBPHY Transmitter Control (TX).
- Adjust the HS RX thresholds for the envelope detector or squelch comparator and the host disconnect comparator by using USBPHY Receiver Control (RX).

The default values of the transceiver parametric trim fields are centered with no changes needed for USB certification testing when the chip is used with boards optimized for signal integrity on the HS USB port.

It may be useful to make changes to the parametric trim fields in cases where external components are inserted between the USB DP and DN pins, or compromises are made on USB DP and DN signal routing.

52.6.2 Register macro usage

A read-modify-write (RMW) operation involves updating a field without disrupting the contents of the remaining fields in the register. In this operation, the CPU reads the register, modifies the target field, and then writes the results back to the register. Because of the initial register read, this is an expensive operation in terms of CPU cycles.

To address this issue, you can implement some USBPHY registers as a group, including registers that can be used to either set, clear, or toggle (SCT) individual fields of the primary register. When writing to an SCT register, all fields that are 1 perform the associated operation on the primary register, while all fields that are 0 are not affected. The SCT registers always read back 0, and must have the write-only access type. You cannot implement SCT registers if the primary register has a read-only access type.

With this architecture, you can update one or more fields using only register writes. First, all bits of the target fields are cleared by a write to the associated clear register, and then the desired value of the target fields is written to the set register. This sequence of two writes is referred to as a clear-set (CS) operation.

A CS operation does have a potential drawback—whenever a field is modified, USBPHY sees a value of 0 before you write the final value. For most fields, passing through the 0 state is not a problem. Nonetheless, this behavior is something you must consider when using a CS operation.

Also, you do not require a CS operation for fields that are 1-bit wide. While the CS operation works in this case, it is more appropriate to set or clear the target bit (that is, one write instead of two). A simple set or clear operation is also atomic, while a CS operation is not.

NOTE

All macros for SCT are not atomic. For registers that do not provide hardware support for this functionality, these macros are implemented as a sequence of RMW operations. You must remember this when an atomic operation is required because an unexpected behavior may take place if an interrupt occurs during a critical update sequence.

52.7 USBPHY register descriptions

USBPHY does not generate transfer errors when accessing unimplemented registers in its register space.

Also, the SCT feature is implemented for some registers (see Register macro usage for information about this feature).

52.7.1 USBPHY memory map

USBPHY base address: 4041_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	USBPHY Powerdown (PWD)	32	RW	001E_1C00h
4h	USBPHY Powerdown (PWD_SET)	32	RW	001E_1C00h
8h	USBPHY Powerdown (PWD_CLR)	32	RW	001E_1C00h
Ch	USBPHY Powerdown (PWD_TOG)	32	RW	001E_1C00h
10h	USBPHY Transmitter Control (TX)	32	RW	1008_0807h
14h	USBPHY Transmitter Control (TX_SET)	32	RW	1008_0807h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
18h	USBPHY Transmitter Control (TX_CLR)	32	RW	1008_0807h
1Ch	USBPHY Transmitter Control (TX_TOG)	32	RW	1008_0807h
20h	USBPHY Receiver Control (RX)	32	RW	0000_0000h
24h	USBPHY Receiver Control (RX_SET)	32	RW	0000_0000h
28h	USBPHY Receiver Control (RX_CLR)	32	RW	0000_0000h
2Ch	USBPHY Receiver Control (RX_TOG)	32	RW	0000_0000h
30h	USBPHY General Control (CTRL)	32	RW	See section
34h	USBPHY General Control (CTRL_SET)	32	RW	See section
38h	USBPHY General Control (CTRL_CLR)	32	RW	See section
3Ch	USBPHY General Control (CTRL_TOG)	32	RW	See section
40h	USBPHY Status (STATUS)	32	R	0000_0000h
50h	USBPHY Debug (DEBUG)	32	RW	7F18_0000h
54h	USBPHY Debug (DEBUG_SET)	32	RW	7F18_0000h
58h	USBPHY Debug (DEBUG_CLR)	32	RW	7F18_0000h
5Ch	USBPHY Debug (DEBUG_TOG)	32	RW	7F18_0000h
60h	UTMI Debug Status 0 (DEBUG0_STATUS)	32	R	0000_0000h
70h	UTMI Debug Status 1 (DEBUG1)	32	RW	0000_1000h
74h	UTMI Debug Status 1 (DEBUG1_SET)	32	RW	0000_1000h
78h	UTMI Debug Status 1 (DEBUG1_CLR)	32	RW	0000_1000h
7Ch	UTMI Debug Status 1 (DEBUG1_TOG)	32	RW	0000_1000h
80h	USBPHY Version (VERSION)	32	R	0500_0000h
A0h	USBPHY PLL Control and Status (PLL_SIC)	32	RW	00D1_2000h
A4h	USBPHY PLL Control and Status (PLL_SIC_SET)	32	RW	00D1_2000h
A8h	USBPHY PLL Control and Status (PLL_SIC_CLR)	32	RW	00D1_2000h
ACh	USBPHY PLL Control and Status (PLL_SIC_TOG)	32	RW	00D1_2000h
C0h	USBPHY VBUS Detect Control (USB1_VBUS_DETECT)	32	RW	0070_0004h
C4h	USBPHY VBUS Detect Control (USB1_VBUS_DETECT_SET)	32	RW	0070_0004h
C8h	USBPHY VBUS Detect Control (USB1_VBUS_DETECT_CLR)	32	RW	0070_0004h
CCh	USBPHY VBUS Detect Control (USB1_VBUS_DETECT_TOG)	32	RW	0070_0004h
D0h	USBPHY VBUS Detector Status (USB1_VBUS_DET_STAT)	32	R	0000_0001h
E0h	USBPHY Charger Detect Control (USB1_CHRG_DETECT)	32	RW	8018_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
E4h	USBPHY Charger Detect Control (USB1_CHRG_DETECT_SET)	32	RW	8018_0000h
E8h	USBPHY Charger Detect Control (USB1_CHRG_DETECT_CLR)	32	RW	8018_0000h
ECh	USBPHY Charger Detect Control (USB1_CHRG_DETECT_TOG)	32	RW	8018_0000h
F0h	USBPHY Charger Detect Status (USB1_CHRG_DET_STAT)	32	R	0000_0000h
100h	USBPHY Analog Control (ANACTRL)	32	RW	8A00_0402h
104h	USBPHY Analog Control (ANACTRL_SET)	32	RW	8A00_0402h
108h	USBPHY Analog Control (ANACTRL_CLR)	32	RW	8A00_0402h
10Ch	USBPHY Analog Control (ANACTRL_TOG)	32	RW	8A00_0402h
110h	USBPHY Loopback Control and Status (USB1_LOOPBACK)	32	RW	0055_0000h
114h	USBPHY Loopback Control and Status (USB1_LOOPBACK_SET)	32	RW	0055_0000h
118h	USBPHY Loopback Control and Status (USB1_LOOPBACK_CLR)	32	RW	0055_0000h
11Ch	USBPHY Loopback Control and Status (USB1_LOOPBACK_TOG)	32	RW	0055_0000h
120h	USBPHY Loopback Packet Number Selection (USB1_LOOPBACK_HSFSCNT)	32	RW	0004_0010h
124h	USBPHY Loopback Packet Number Selection (USB1_LOOPBACK_HSFSCNT_SET)	32	RW	0004_0010h
128h	USBPHY Loopback Packet Number Selection (USB1_LOOPBACK_HSFSCNT_CLR)	32	RW	0004_0010h
12Ch	USBPHY Loopback Packet Number Selection (USB1_LOOPBACK_HSFSCNT_TOG)	32	RW	0004_0010h
130h	USBPHY Trim Override Enable (TRIM_OVERRIDE_EN)	32	RW	0000_007Fh
134h	USBPHY Trim Override Enable (TRIM_OVERRIDE_EN_SET)	32	RW	0000_007Fh
138h	USBPHY Trim Override Enable (TRIM_OVERRIDE_EN_CLR)	32	RW	0000_007Fh
13Ch	USBPHY Trim Override Enable (TRIM_OVERRIDE_EN_TOG)	32	RW	0000_007Fh

52.7.2 USBPHY Powerdown (PWD)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
PWD	0h	USBPHY Powerdown

Table continues on the next page...

Table continued from the previous page...

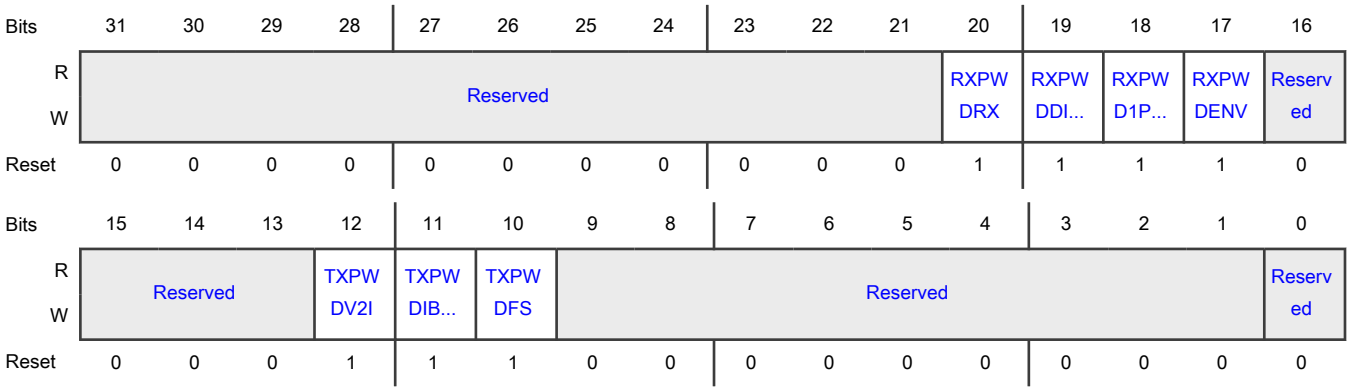
Register	Offset	Description
PWD_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in PWD is 1
PWD_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in PWD is 0
PWD_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in PWD

Function

Provides an overall control of the PHY power state.

You must enable the PHY clocks before programming this register.

Diagram



Fields

Field	Function
31-21 —	Reserved
20 RXPWDRX	Receiver Powerdown Disables the HS RX differential receiver, HS envelope detector, and HS host disconnect comparator along with their RX current mirror bias circuits. The overall control of the RX current bias block is shared with the battery charge circuit, but this field powers down the other aforementioned RX circuits (the entire USBPHY receiver block) unconditionally, except for the FS differential receiver. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Enable for normal operation

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Disable or power down RX circuits
19 RXPWDDIFF	HS Receiver Powerdown Disables the USB HS RX differential receiver that is used for data reception in HS operation. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Enable for normal operation 1b - Disable or power down
18 RXPWD1PT1	FS Receiver Powerdown Disables the USB FS RX differential receiver that is used for data reception in both FS and LS operations. This field does not affect the bias and operation of the FS differential receiver. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Enable for normal operation 1b - Disable or power down
17 RXPWDENV	Receiver Envelope Powerdown Disables the USB HS RX envelope detector that generates the squelch signal to qualify the HS RX data. The host disconnect comparator is used in Host mode to check for a disconnection from the remote device. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Enable for normal operation 1b - Disable or power down
16-13 —	Reserved
12 TXPWDV2I	USBPHY TX V-I Converter and Current Mirror Powerdown Disables the USBPHY TX V-I converter that provides currents used in the HS TX drivers. The converter depends on both USBPHY's bias module and the bias current block. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Enable for normal operation 1b - Disable or power down
11 TXPWDIBIAS	Transmitter Bias Powerdown Disables the USBPHY current bias block for the transmitter that provides mirrored currents to track USBPHY's bias module reference current generator for several functions in the TX, RX, and detector sections. Control of the current bias block is shared with the battery charge circuit. Writing 1 to this field does not power down the circuit unless the value of the corresponding field in the battery charge control is also 1 for power down. The field must be 1 only when the USB is in the Suspend bus state. This effectively powers down the entire USB transmit path. The field must become 1 only when the USB is in Suspend mode. This effectively powers down the entire USB transmit path. This field becomes 0 automatically in case of a wake-up event when USB is suspended while CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. **NOTE** These circuits are shared with the battery charge circuit. Writing 1 to this field does not power down these circuits, unless the corresponding field in the battery charger is also configured as 1 for a power down. 0b - Enable for normal operation 1b - Disable or power down
10 TXPWDFS	FS Transmitter Powerdown Disables the USB FS TX drivers that must have their bias enabled during USB data communication, including resume and bus reset signaling. They are used in HS, FS, and LS because pulldown of the FS TX drivers provides local HS termination. If this field is 1, it turns off the current starvation sources and puts the drivers into high-impedance output. This field becomes 0 in case of a wake-up event if USB is suspended when CTRL[ENAUTOCLR_PHY_PWD] is 1. Additionally, this field becomes 1 automatically at the falling edge of the signal from the USB controller that drives CTRL[UTMI_SUSPENDM] to enter the Suspend state. 0b - Provide bias to enable for normal operation 1b - Disable or power down
9-1 —	Reserved
0	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	

52.7.3 USBPHY Transmitter Control (TX)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TX	10h	USBPHY Transmitter Control
TX_SET	14h	Writing 1 to a bit in this register ensures that the corresponding bit in TX is 1
TX_CLR	18h	Writing 1 to a bit in this register ensures that the corresponding bit in TX is 0
TX_TOG	1Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TX

Function

Allows adjustment of transmit parametric values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				Reserved				Reserved		Reserved	Reserved	TXCAL45DP			
Reset	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved	Reserved	TXCAL45DN				Reserved				D_CAL			
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1

Fields

Field	Function
31-29 —	Reserved
28-24	The value of this reserved bit must remain 10000b.

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
23-22 —	Reserved
21 —	The value of this reserved bit must remain 0b.
20 —	Reserved
19-16 TXCAL45DP	Transmit Calculation 45 Ω DP Specifies the trim value of the 45 Ω DP resistance. Decode to trim the nominal 45 Ω series termination resistance to the DP output pin. Resistance is centered at value 0111b, with the maximum resistance being 0000b. 0000b - +19.95% 0001b - +17.35% 0010b - +14.85% 0011b - +12.46% 0100b - +9.07% 0101b - +5.87% 0110b - +2.85% 0111b - 0% 1000b - –2.70% 1001b - –5.25% 1010b - –7.67% 1011b - –9.98% 1100b - –12.17% 1101b - –14.25% 1110b - –18.14% 1111b - –21.68%
15-14 —	Reserved
13 —	The value of this reserved bit must remain 0b.

Table continues on the next page...

Table continued from the previous page...

Field	Function
12 —	Reserved
11-8 TXCAL45DN	Transmit Calculation 45 Ω DN Specifies the trim value of the 45 Ω DN resistance. Decode to trim the nominal 45 Ω series termination resistance to the DN output pin. The resistance is centered at value 0110b, with the maximum resistance being 0000b. 0000b - +19.95% 0001b - +17.35% 0010b - +14.85% 0011b - +12.46% 0100b - +9.07% 0101b - +5.87% 0110b - +2.85% 0111b - 0% 1000b - –2.70% 1001b - –5.25% 1010b - –7.67% 1011b - –9.98% 1100b - –12.17% 1101b - –14.25% 1110b - –18.14% 1111b - –21.68%
7-4 —	The value of this reserved bit must remain 0b.
3-0 D_CAL	HS Transmit Output Current Trim Specifies the HS transmit output current trim value. Decode to trim the nominal 17.78 mA current source for the HS transmit drivers on the DP and DN pins. This current is directly proportional to the amplitude of the HS TX eye diagram. 0000b - +20.30% 0001b - +17.60% 0010b - +14.80% 0011b - +12.60%

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0100b - +8.79% 0101b - +6.04% 0110b - +2.75% 0111b - 0% 1000b - –2.75% 1001b - –5.49% 1010b - –7.69% 1011b - –10.40% 1100b - –12.60% 1101b - –14.30% 1110b - –18.10% 1111b - –22.00%

52.7.4 USBPHY Receiver Control (RX)

Offset

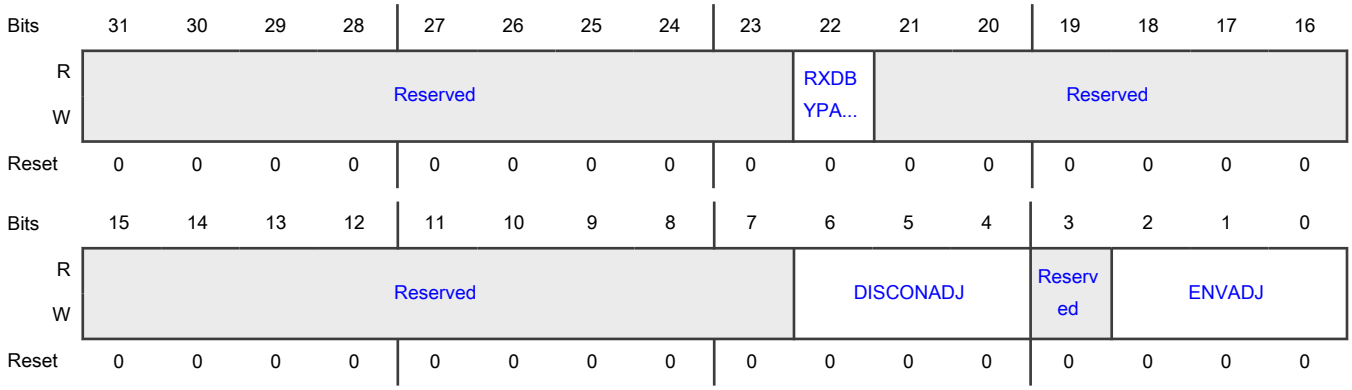
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
RX	20h	USBPHY Receiver Control
RX_SET	24h	Writing 1 to a bit in this register ensures that the corresponding bit in RX is 1
RX_CLR	28h	Writing 1 to a bit in this register ensures that the corresponding bit in RX is 0
RX_TOG	2Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in RX

Function

Allows adjustment of receive parametric values.

Diagram



Fields

Field	Function
31-23 —	Reserved
22 RXDBYPASS	Differential Receiver Bypass Bypasses the FS differential receiver. If this field is 0, normal operation continues. If this field is 1, the output of the DP single-ended receiver is used instead of the FS differential receiver. 0b - Operate normally 1b - Bypass
21-7 —	Reserved
6-4 DISCONADJ	Disconnect Adjustment Adjusts the trip point (trip-level voltage) for the disconnect detector. 000b - 0.56875 V 001b - 0.55000 V 010b - 0.58125 V 011b - 0.60000 V 1xxb - Reserved
3 —	Reserved
2-0 ENVADJ	Envelope Adjustment

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Adjusts the trip point for the envelope detector. The field values are trip-level voltages in nominal DC settings to indicate the effect of changing these fields. AC values measured during compliance testing are a bit higher. 000b - 0.1000 V 001b - 0.1125 V 010b - 0.1250 V 011b - 0.0875 V 1xxb - Reserved

52.7.5 USBPHY General Control (CTRL)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	30h	USBPHY General Control
CTRL_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Function

Handles OTG and host controls.

This register also includes interrupt enables, connectivity detect enables, and results.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFTRST	CLKGATE	UTMI_SU...	HOST_FO...	Reserved	Reserved		FSDLL_R...	Reserved		Reserved	ENAUTOC...	ENAUTOC...	AUTORES...	WAKEUP_...	ENIRQWA...
W																
Reset	1	1	0	0	u	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENUTMIL...	ENUTMIL...	Reserved	DEVPLUG...	ENIRQDE...	RESUME_...	ENIRQRE...	RESUMEI...	Reserved		DEVPLUG...	ENDEVPL...	HOSTDIS...	ENIRQHO...	ENHOSTD...	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Fields

Field	Function
31 SFTRST	Software Reset Soft-resets USBPHY Powerdown (PWD), USBPHY Transmitter Control (TX), USBPHY Receiver Control (RX), and USBPHY General Control (CTRL). 0b - Release from reset 1b - Reset
30 CLKGATE	Clock Gating Gates UTMI clocks. Write 1 to this field to save power when the USB is not actively used. Retain the configuration state when the clock is gated. **NOTE** This field can automatically become 0 in case of a wake-up event when USB is suspended, if ENAUTOCLR_CLKGATE is 1. 0b - Run clocks 1b - Gate clocks
29 UTMI_SUSPENDM	UTMI Suspend Mode Indicates a power-down state. UTMI_SUSPENDM is negative logic, as the UTMI specification requires it to be. If this field is 0, all powerdown fields in USBPHY Powerdown (PWD) are enabled. If this field is 0, the USB controller enters Suspend mode. 0b - Suspended 1b - Not suspended
28 HOST_FORCE_LS_SE0	FS EOP LS Timing Works with DEBUG[HOST_RESUME_DEBUG] and forces the next FS packet that is transmitted to have an EOP with LS timing. This field is used in Host mode for the resume sequence. After the packet is transferred, the field becomes 0. You can use this function to force the LS SE0 or use UTMI_SUSPENDM to trigger this event when existing the Suspend state. 0b - Do not force the next FS packet 1b - Force the next FS packet
27 —	Reserved
26-25 —	Reserved
24	FSDLL Reset Enable Enables resetting of the FSDLL lock detection logic at the end of each TX packet.

Table continues on the next page...

Table continued from the previous page...

Field	Function
FSDLL_RST_EN	0b - Disables 1b - Enables
23-22 —	Reserved
21 —	This bit must remain at value 0b.
20 ENAUTOCLR_PHY_PWD	PHY PWD Auto Clear Enable Enables automatic writing of 0 to the fields in USBPHY Powerdown (PWD) in case of a wake-up event while USB is suspended. You must write 1 to this field if auto wake-up is required without software interaction. 0b - Disables 1b - Enables
19 ENAUTOCLR_CLKGATE	Clock Gating Auto Clear Enable Enables automatic writing of 0 to CLKGATE in case of a wake-up event while USB is suspended. You must write 1 to this field if auto wake-up is required without software interaction. 0b - Disables 1b - Enables
18 AUTORESUME_EN	Auto Resume Enable Enables the auto resume feature. If this field is 1, USBPHY uses a 32 kHz clock to send resume signaling to respond to the USB device remote wakeup (for Host mode only). The field functionality is useful when PLL is off and the reference clock is powered down. 0b - Disables 1b - Enables
17 WAKEUP_IRQ	Wake-Up Interrupt Specifies whether a wake-up event exists. Reset this field by writing 1 to the clear address space, and not by a general write. 0b - No wake-up event exists 1b - Wake-up event exists
16 ENIRQWAKEUP	Wake-Up Interrupt Enable Enables interrupts for wake-up events. 0b - Disables 1b - Enables
15	UTMI Level 3 Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
ENUTMILEVEL3	Enables UTMI+ level 3 operation for USBPHY. You must write 1 to this field if an embedded host use case needs to support an external FS hub with an LS device connected. 0b - Disables 1b - Enables
14 ENUTMILEVEL2	UTMI Level 2 Enable Enables UTMI+ level 2 operation for USBPHY. You must write 1 to this field if an embedded host use case needs to support an LS device. 0b - Disables 1b - Enables
13 —	The value of this reserved bit must remain 0b.
12 DEVPLUGIN_IRQ	Device Plug-In Interrupt Indicates that the device is connected. Reset this field by writing 1 to the SCT clear address space, and not by a general write. 0b - Not connected 1b - Connected
11 ENIRQDEVPLUGIN	Device Plug-In Interrupt Enable Enables an interrupt for the detection of connectivity to the USB line. 0b - Disables 1b - Enables
10 RESUME_IRQ	Interrupt Resume Specifies whether the host is sending a wake-up signal after suspend signaling. This field also becomes 1 after a reset during Suspend mode. Use this field to wake up the USB device from Suspend mode for either the host resume or host reset case. You must reset this field by writing 1 to the clear address space, and not by a general write. 0b - No resume interrupt 1b - Resume interrupt
9 ENIRQRESUMEDETECT	Resume Detection Interrupt Enable Enables an interrupt for detection of a non-J state on the USB line. You must write 1 to this field only after the USB device has entered Suspend mode. 0b - Disables 1b - Enables
8	RESUME_IRQ Sticky

Table continues on the next page...

Table continued from the previous page...

Field	Function
RESUMEIRQSTICKY	Specifies until when RESUME_IRQ remains 1 during the resume or reset state signaling period. 0b - Remains 1 during the wake-up period 1b - Remains 1 until you write 0 to it
7-6 —	Reserved
5 DEVPLUGIN_POLARITY	Device Plug-In Polarity Specifies whether to trip the interrupt when the USB device is plugged in or when it is unplugged. If this field is 0, it trips the interrupt when the USB cable from the host to the device is plugged in. If this field is 1, it trips the interrupt if the USB cable from the host to the device is unplugged. 0b - Plugged in 1b - Unplugged
4 ENDEVPLUGINDETECT	Nonstandard Resistive Plugged-In Detection Enable Controls connection of nominal 200 kΩ pullup resistors with both the DP and DN pins as a method of detecting, in Device mode, when a USB cable is attached. This field must remain 0 for normal USB data communication, or when using DCD for battery charger detection, according to *USB Battery Charger Specification Revision 1.2* or any other detection mechanism for USB cable plugin. STATUS[DEVPLUGIN_STATUS] reports the results of this detection method. 0b - Disables 1b - Enables
3 HOSTDISCONDETECT_IRQ	Host Disconnect Detection Interrupt Specifies whether the device is disconnected in HS Host mode. Reset this field by writing 1 to the SCT clear address space, and not by a general write. 0b - Connected 1b - Disconnected
2 ENIRQHOSTDISCON	Host Disconnect Interrupt Enable Enables an interrupt for detection of device disconnection when in HS Host mode. You must write 1 to this field after writing 1 to ENDEVPLUGINDETECT. 0b - Disables 1b - Enables
1 ENHOSTDISCONDETECT	Host Disconnect Detection Enable Enables the HS disconnect detector for Host mode. This signal allows the override of detection enabling, which the UTMI controller normally performs. The UTMI controller enables this circuit whenever the host sends a start-of-frame packet.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Write 1 to this field when an HS device is connected. This is suggested during a bus reset. Ensure that this field is not 1 at the end of resume signaling; otherwise, a wrong disconnect status may be detected. 0b - Disables 1b - Enables
0 —	Reserved

52.7.6 USBPHY Status (STATUS)

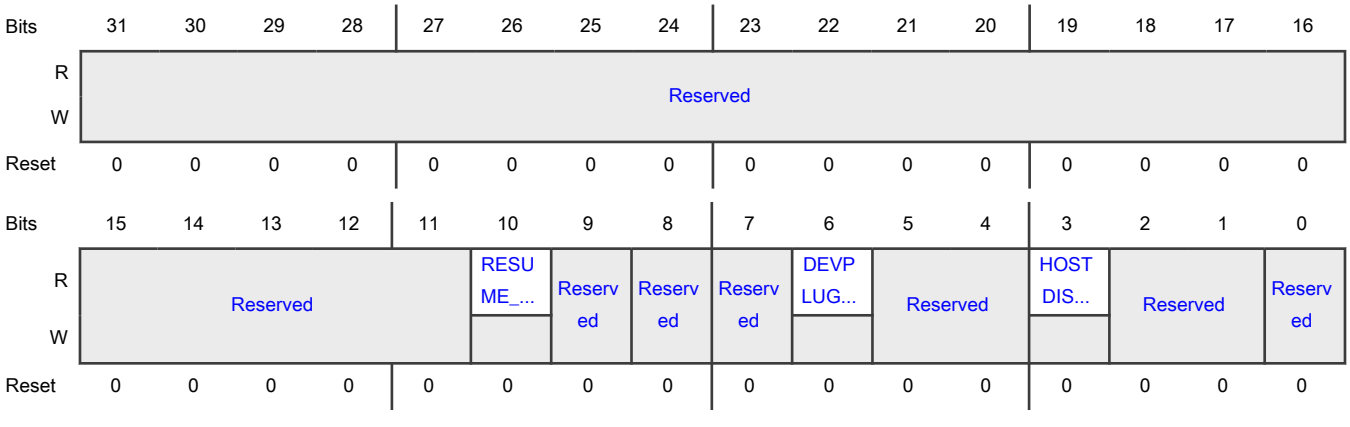
Offset

Register	Offset
STATUS	40h

Function

Holds results of IRQ and other detects.

Diagram



Fields

Field	Function
31-11 —	Reserved
10	Resume Status

Table continues on the next page...

Table continued from the previous page...

Field	Function
RESUME_STATUS	Indicates whether the line state is in J state. This field provides a valid indication of resume signaling only when operating in FS or HS mode, after being in the Suspend bus state. This status field may toggle during USB data packet communication. Also, it does not indicate resume signaling when operating as a host that is directly attached to an LS device. 0b - Is in J state 1b - Is not in J state
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 DEVPLUGIN_STATUS	Status Indicator for Nonstandard Resistive Plugged-In Detection Indicates whether the device is connected on the DP and DN lines using the nonstandard resistive plugged-in detection method that CTRL[ENDEVPLUGINDETECT] controls. When a USB cable attached to a remote host is attached to the local device, the 15 kΩ host pulldowns override the high value resistors used in this detection method. 0b - No attachment detected 1b - Cable attachment detected
5-4 —	Reserved
3 HOSTDISCONDETECT_STATUS	Host Disconnect Detection Status Detects at the local host (downstream) port that the USB cable is disconnected when in HS mode. 0b - Do not detect 1b - Detect
2-1 —	Reserved
0 —	Reserved

52.7.7 USBPHY Debug (DEBUG)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DEBUG	50h	USBPHY Debug
DEBUG_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in DEBUG is 1
DEBUG_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in DEBUG is 0
DEBUG_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DEBUG

Function

Debugs USBPHY.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	CLKGATE	HOST_RE...	SQUELCHRESETLENGTH				ENSQUEL...	Reserved			SQUELCHRESETCOUNT				
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			ENTX2RX...	TX2RXCOUNT				Reserved		ENHSTPULLDOWN		HSTPULLDOWN		Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30 CLKGATE	Clock Gating Gates the test clocks. To save power while the USB is not being actively used, retain the value of 1 in this field. The configuration state is maintained when the clock is gated. 0b - Run clocks 1b - Gate clocks
29 HOST_RESUME_DEBUG	Host Resume Selects the method used to generate the LS SE0 at the end of host resume signaling. The default method triggers from a change to the UTMI_SUSPENDM signal from the USB controller to the PHY.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The other method depends on programming CTRL[HOST_FORCE_LS_SE0] at an appropriate time. If this field is 0, it generates LS SE0 at the end of host resume based on CTRL[HOST_FORCE_LS_SE0]. If this field is 1, it generates LS SE0 at the end of host resume based on the state of CTRL[UTMI_SUSPENDM]. 0b - Based on CTRL[HOST_FORCE_LS_SE0] 1b - Based on CTRL[UTMI_SUSPENDM]
28-25 SQUELCHRESETLENGTH	Squelch Reset Length Indicates the number of 480 MHz clock intervals used for the delay time. The changes you make to this field are used only for a specific HS RX debug mode that ENSQUELCHRESET enables. This field adjusts the delay time between the detection of valid HS RX data and release of squelch for that debug mode. The field value indicates the number of 480 MHz clock intervals used for the delay time.
24 ENSQUELCHRESET	Enable Squelch Reset Enables squelch reset. Let the value of this field be 1 to allow normal operation of the squelch state machine for qualification of HS receive data. Writing 0 to this field freezes the functional mode squelch state machine in its reset state, and must be done only for a specific HS RX debug mode. 0b - Disables 1b - Enables
23-21 —	Reserved
20-16 SQUELCHRESETCOUNT	Squelch Reset Count Adjusts the delay time, in 480 MHz clock intervals, between the detection of squelch to the reset of HS RX state machines. Any changes that you make to this field must only be used for a specific HS RX debug mode.
15-13 —	Reserved
12 ENTX2RXCOUNT	Enable Countdown from TX to RX Packets for Debug Enables a countdown to transition between the TX and RX packets for debug. 0b - Disables 1b - Enables
11-8 TX2RXCOUNT	Set Countdown Delay Value from TX to RX Packets for Debug Specifies the delay between the end of transmit to the beginning of receive packets for debug. This is a Johnson counter value and therefore counts to 8.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-6 —	Reserved
5-4 ENHSTPULLDOWN	Enable Host Pulldown Overdrive Mode Selects whether to override the control of DP and DN pins. This field selects Host Pulldown Overdrive mode. Write 1b to bit 5 to override the control of the DP 15 kΩ pulldown, which the USB controller normally performs. Write 1b to bit 4 to override the control of the DN 15 kΩ pulldown, which the USB controller normally performs. Write 00b to both the bits to disable Host Pulldown Overdrive mode and return the control of the pulldown resistors to the USB controller. When in Host Pulldown Overdrive mode, HSTPULLDOWN further controls the connection of the individual pulldown resistors. Both pulldown resistors are also unconditionally enabled when ANACTRL[DEV_PULLDOWN] becomes 1 without considering the value of this field. 00b - Disable Host Pulldown Overdrive mode 01b - Override the control of DN 15 kΩ pulldown 10b - Override the control of DP 15 kΩ pulldown 11b - Override the control of DP and DN 15 kΩ pulldown
3-2 HSTPULLDOWN	Select DP and DN Pulldown Resistors in Host Pulldown Overdrive Mode Selects whether to connect pulldown resistors on the DP and DN pins if Host Pulldown Overdrive mode is enabled through ENHSTPULLDOWN. Write 1b to bit 3 of this field to connect the 15 kΩ pulldown on the DP line, and write 1b to bit 2 to connect the 15 kΩ pulldown on the DN line. Write 00b to both the bits to disconnect the resistors in Host Pulldown Override mode. 00b - Disconnect the resistors 01b - Connect 15 kΩ pulldown on DN 10b - Connect 15 kΩ pulldown on DP 11b - Connect 15 kΩ pulldown on DP and DN
1 —	Reserved
0 —	Reserved

52.7.8 UTMI Debug Status 0 (DEBUG0_STATUS)

Offset

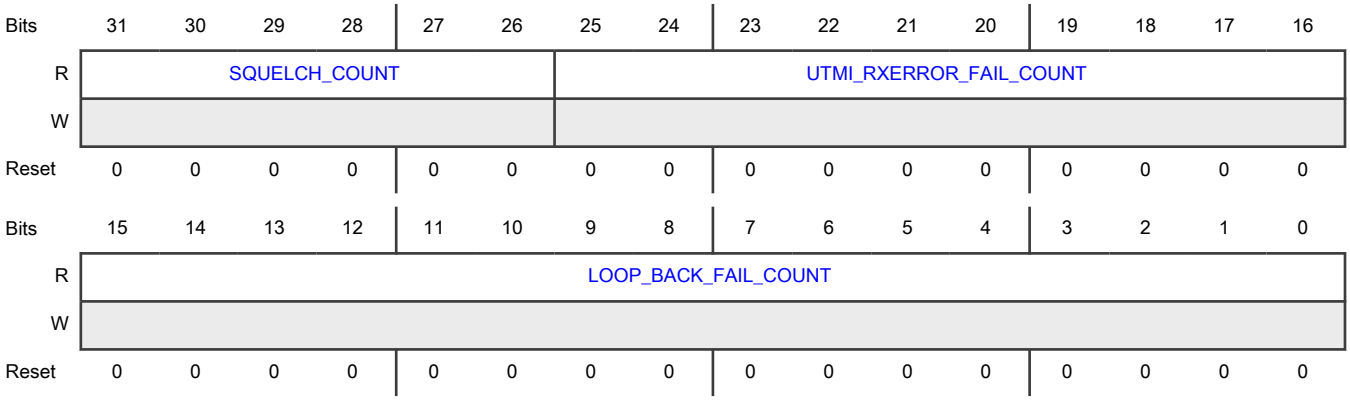
Register	Offset
DEBUG0_STATUS	60h

Function

Holds multiple views for counters and status of state machines.

This field works with DEBUG1[DBG_ADDRESS] to select the function to view. The default settings used to count errors are described in this register.

Diagram



Fields

Field	Function
31-26 SQUELCH_COUNT	Squelch Count Indicates the running count of the squelch reset instead of the normal end for HS RX.
25-16 UTMI_RXERROR_FAIL_COUNT	UTMI Receive Error Fail Count Indicates the running count of UTMI_RXERROR.
15-0 LOOP_BACK_FAIL_COUNT	Loopback Fail Count Indicates the running count of the failed pseudo-random generator loopback. Each time it enters Test mode, the counter goes to 900D and counts up for every detected packet failure in digital and analog loopback tests.

52.7.9 UTMI Debug Status 1 (DEBUG1)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DEBUG1	70h	UTMI Debug Status 1

Table continues on the next page...

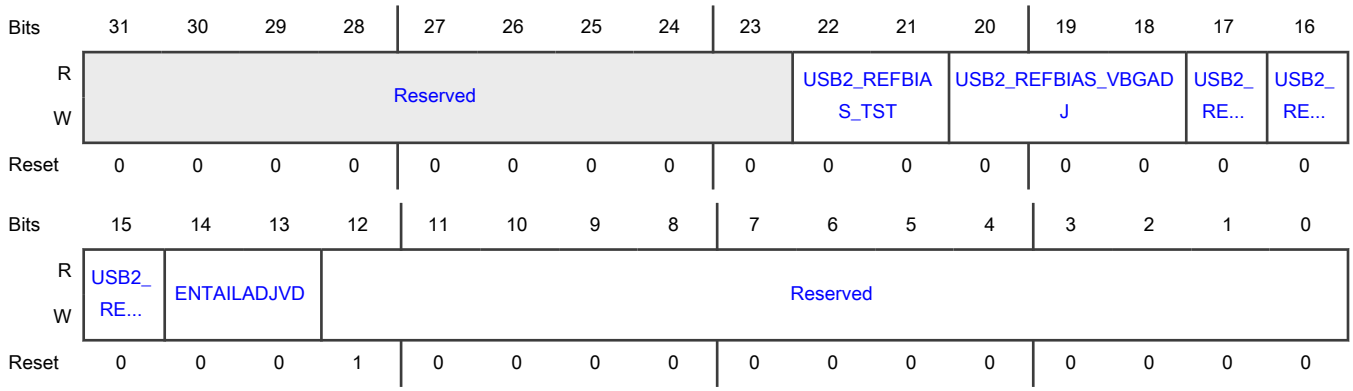
Table continued from the previous page...

Register	Offset	Description
DEBUG1_SET	74h	Writing 1 to a bit in this register ensures that the corresponding bit in DEBUG1 is 1
DEBUG1_CLR	78h	Writing 1 to a bit in this register ensures that the corresponding bit in DEBUG1 is 0
DEBUG1_TOG	7Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DEBUG1

Function

Selects the muxing of USBPHY Debug (DEBUG) to be indicated in UTMI Debug Status 0 (DEBUG0_STATUS).

Diagram



Fields

Field	Function
31-23 —	Reserved
22-21 USB2_REFBIAS_TST	Bias Current Control Adjustment Specifies the main current trim for the USBPHY bias module current reference generator. Use this field to trim the PHY's current reference when TRIM_OVERRIDE_EN[TRIM_REFBIAS_TST_OVERRIDE] is 1. If TRIM_OVERRIDE_EN[TRIM_REFBIAS_TST_OVERRIDE] becomes 0, the current trim values in TRIM_OVERRIDE_EN[TRIM_USB2_REFBIAS_TST] are used instead. 00b - ≈ 10 µA reference current; nominal 01b - ≈ 0.9x compared to nominal 10b - ≈ 0.8x compared to nominal 11b - ≈ 1.1x compared to nominal

Table continues on the next page...

Table continued from the previous page...

Field	Function
20-18 USB2_REFBIAS_VBGADJ	Bandgap Voltage Adjustment Specifies the voltage adjustment trim on USBPHY bias module bandgap generator. Use this field to trim the PHY's bandgap voltage when TRIM_OVERRIDE_EN[TRIM_REFBIAS_VBGADJ_OVERRIDE] is 1. If TRIM_OVERRIDE_EN[TRIM_REFBIAS_VBGADJ_OVERRIDE] becomes 0, the voltage trim values in TRIM_OVERRIDE_EN[TRIM_USB2_REFBIAS_VBGADJ] are used instead. 000b - Nominal bandgap voltage; flattest temperature coefficient 001b - ≈ +10 mV compared to nominal 010b - ≈ +20 mV compared to nominal 011b - ≈ +30 mV compared to nominal; most-positive temperature coefficient 100b - ≈ -10 mV compared to nominal 101b - ≈ -20 mV compared to nominal 110b - ≈ -30 mV compared to nominal 111b - ≈ -40 mV compared to nominal; most-negative temperature coefficient
17 USB2_REFBIAS_LOWPWR	Reference Bias Low Power Configuration Configures the bias currents in the bandgap reference amplifier in the USBPHY bias module. If this field is 0, the bandgap amplifier for a nominal bias current is configured. If this field is 1, the bandgap amplifier for 50% of the nominal bias current is configured. The field must remain 0 for normal operation. Writing 1 is only intended for lab characterization purposes. 0b - Nominal bias current 1b - 50% of nominal bias current
16 USB2_REFBIAS_PWDVBGUP	Bandgap Voltage Status Comparator Powerdown Disables the VBGUP comparator to save its bias current after the bandgap voltage is valid and stable. The VBGUP comparator monitors the USBPHY bias module bandgap voltage to check that the voltage is in the valid range. If this field is 1, MISC[VBGUP] is also forced to become 1. You cannot detect a bandgap voltage problem if this field is 1, and the PHY does not automatically restore the VBGUP comparator if such a problem occurs. Therefore, it is not recommended to make changes to this field. 0b - Enables 1b - Disables
15 USB2_REFBIAS_SELFBIASOFF	Self-Bias Off for Reference Bias Amplifiers and Comparators Selects the local bias sources to use for the amplifiers and comparators in the USBPHY bias module. If this field is 0, self-bias is used for bias reference amplifier and comparator circuits. If this field is 1, the current reference bias is used for amplifier and comparator circuits.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Self-bias is required at reference generator startup. After the bandgap voltage and reference current generator are stable, bias from the reference generator's own regulated currents can be used instead. You must let the value of this field be 0 during a ramp of the 1.8 V power domain and during any time that the USBPHY bias reference circuit transitions from the disabled to enabled state. Switching to the current reference bias after the bandgap voltage is stable may somehow improve bandgap voltage stability, but could cause problems if the 1.8 V power is interrupted. You must monitor MISC[VBGUP]; intervention is mandatory if the value of this field changes. You must also note the interaction of USB2_REFBIAS_PWDVBGUP and MISC[VBGUP]. 0b - Self-bias 1b - Current reference bias
14-13 ENTAILADJVD	HS RX Squelch Rise Time Delay Trim Adjusts the squelch rise time delay for the squelch signal, from the HS RX envelope detector, when TRIM_OVERRIDE_EN[TRIM_ENV_TAIL_ADJ_VD_OVERRIDE] is 1. If TRIM_OVERRIDE_EN[TRIM_ENV_TAIL_ADJ_VD_OVERRIDE] becomes 0, the squelch rise time trim values in TRIM_OVERRIDE_EN[TRIM_USB_REG_ENV_TAIL_ADJ_VD] are used instead. 00b - Squelch rising edge delay is nominal 01b - +20% delay compared to nominal 10b - -20% delay compared to nominal 11b - -40% delay compared to nominal
12-0 —	The value of this reserved bit must remain 0b.

52.7.10 USBPHY Version (VERSION)

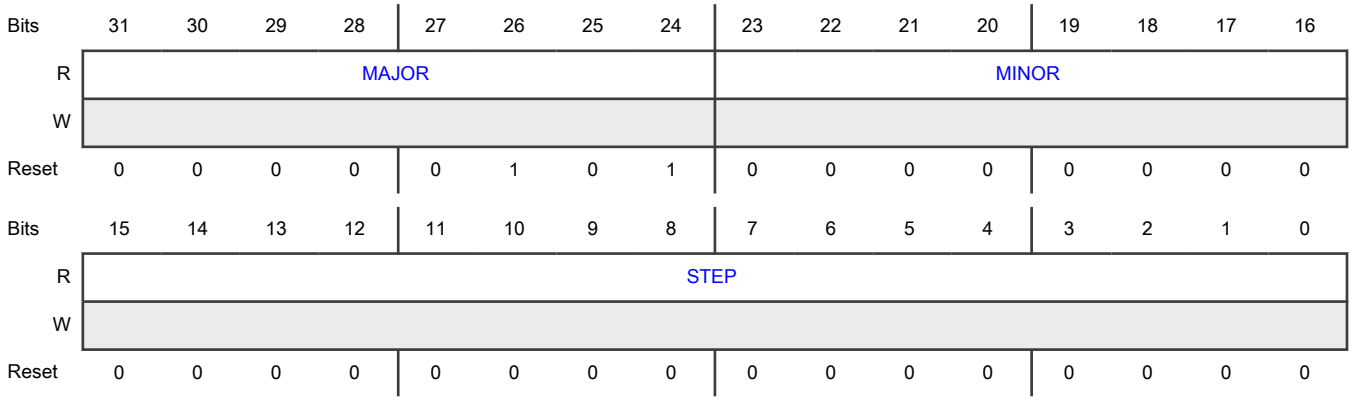
Offset

Register	Offset
VERSION	80h

Function

Indicates the version of USBPHY.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Indicates the "Major" field of the USBPHY version.
23-16 MINOR	Minor Indicates the "Minor" field of the USBPHY version.
15-0 STEP	Step Indicates the stepping of the USBPHY version.

52.7.11 USBPHY PLL Control and Status (PLL_SIC)

Offset

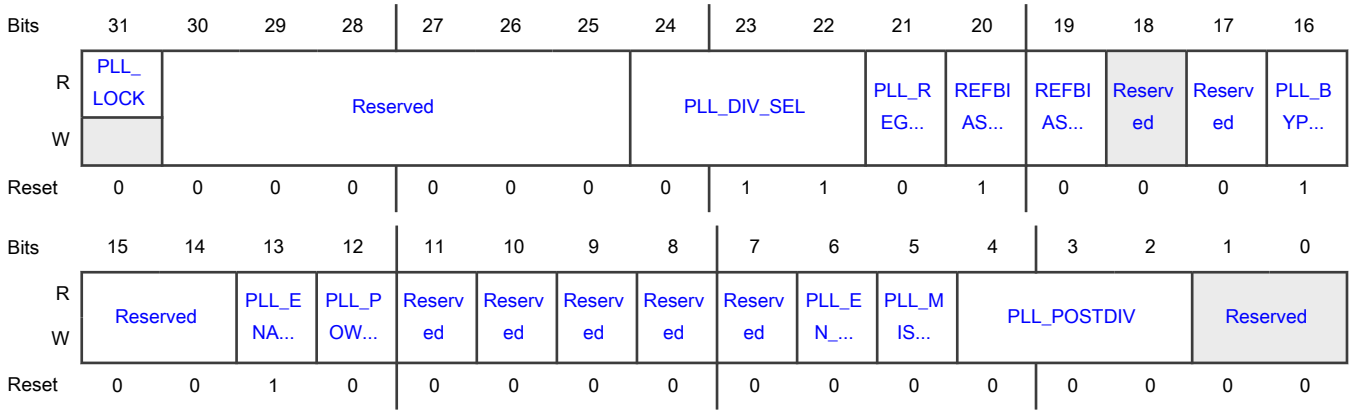
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
PLL_SIC	A0h	USBPHY PLL Control and Status
PLL_SIC_SET	A4h	Writing 1 to a bit in this register ensures that the corresponding bit in PLL_SIC is 1
PLL_SIC_CLR	A8h	Writing 1 to a bit in this register ensures that the corresponding bit in PLL_SIC is 0
PLL_SIC_TOG	ACh	Writing 1 to a bit in this register inverts the value of the corresponding bit in PLL_SIC

Function

Configures the 480 MHz USBPHY PLL.

Diagram



Fields

Field	Function
31 PLL_LOCK	PLL Lock Indicates the current status of the USB PLL lock. 0b - Not locked 1b - Locked
30-25 —	Reserved
24-22 PLL_DIV_SEL	PLL Divider Selection Controls the USB PLL feedback loop divider. The valid range for divider values is 13–40. (Fout = Fin × div_select). The USB PLL produces a 480 MHz output clock. This field allows the use of different frequency signals for the PLL reference clock input connected to the OSCCLK signal from the system oscillator. If TRIM_OVERRIDE_EN[TRIM_DIV_SEL_OVERRIDE] is 1, the USB PLL uses this register value. 000b - Divide by 13 001b - Divide by 15 010b - Divide by 16 011b - Divide by 20 100b - Divide by 22 101b - Divide by 25 110b - Divide by 30 111b - Divide by 40
21 PLL_REG_ENABLE	PLL Regulator Enable Enables the USB PLL regulator. You must set this field to 15 µs before selecting PLL_POWER to avoid glitches on the PLL output clock. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
20 REFBIAS_PWD	Reference Bias Powerdown Enables reference bias powerdown. This field is used only when REFBIAS_PWD_SEL is 1. 0b - Disables 1b - Enables
19 REFBIAS_PWD_SEL	Reference Bias Powerdown Selection Selects PLL_POWER or REFBIAS_PWD to control the reference bias. 0b - PLL_POWER 1b - REFBIAS_PWD
18 —	Reserved
17 —	The value of this reserved bit must remain 0b.
16 PLL_BYPASS	PLL Bypass Bypasses the USB PLL. 0b - Do not bypass 1b - Bypass
15-14 —	Reserved
13 PLL_ENABLE	PLL Enable Enables the clock output from the USB PLL (the PLL powerup control also controls the PLL output enable signal). USBPHY disables the PLL output before PLL powerdown, and enables the PLL output after PLL powerup. You must only set it when initializing the PLL. 0b - Disables 1b - Enables
12 PLL_POWER	PLL Power Enables the USB PLL (USBPHY also controls PLL powerup and powers it down when the USB is suspended and not in use). 0b - Disables 1b - Enables
11	The value of this reserved bit must remain 0b for normal operation.

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
10 —	The value of this reserved bit must remain 0b.
9 —	The value of this reserved bit must remain 0b.
8 —	The value of this reserved bit must remain 0b.
7 —	The value of this reserved bit must remain 0b.
6 PLL_EN_USB_CLKS	PLL USB Clocks Enable Enables the USB clock output from USBPHY PLL. 0b - Disables 1b - Enables
5 PLL_MISC2_CONTROL0	PLL_MISC2_CONTROL0 This field together with PLL_POWER determines when USBPHY's 480 MHz PLL is powered up. The SW stack must write 1b to this bit during the USB initialization routine. 0b - Power up PLL without regard to state of "SuspendM" signal 1b - Power down PLL when in Suspend bus state
4-2 PLL_POSTDIV	PLL Post-Divider Output Value Configuration Specifies the integer divide values for PLL post divider. 000b - Disable the output of PLL post divider 001b - Divide value is 1 010b - Divide value is 2 011b - Divide value is 3 100b - Divide value is 4 101b - Divide value is 5 110b - Divide value is 6 111b - Reserved
1-0 —	Reserved

52.7.12 USBPHY VBUS Detect Control (USB1_VBUS_DETECT)

Offset

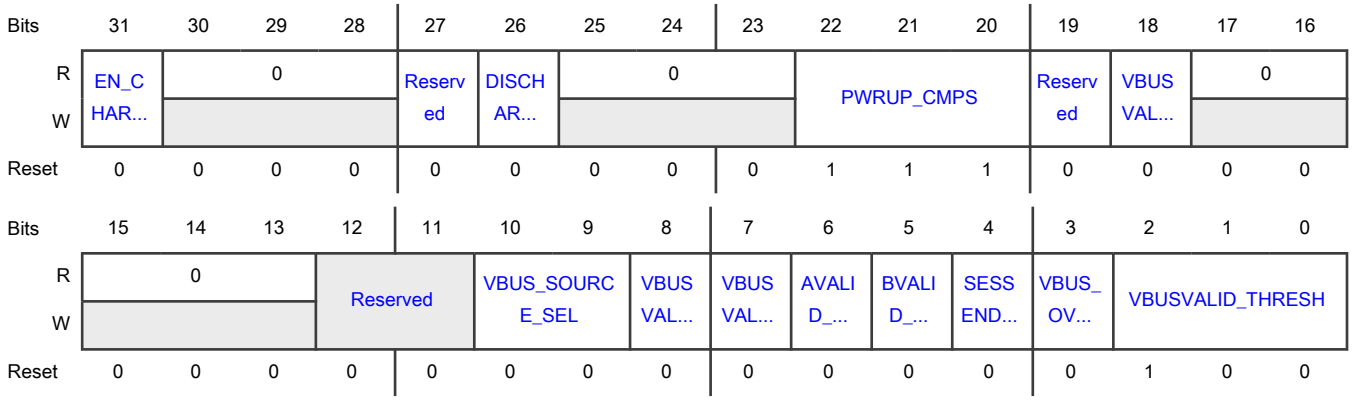
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
USB1_VBUS_DETECT	C0h	USBPHY VBUS Detect Control
USB1_VBUS_DETECT_SET	C4h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_VBUS_DETECT is 1
USB1_VBUS_DETECT_CLR	C8h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_VBUS_DETECT is 0
USB1_VBUS_DETECT_TOG	CCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in USB1_VBUS_DETECT

Function

Defines controls for USB VBUS detect and some additional out-of-band signaling functions.

Diagram



Fields

Field	Function
31 EN_CHARGER_RESISTOR	Charger Resistor Enable Enables a nominal 125 kΩ pullup on DP and a nominal 375 kΩ pulldown on DN. This allows resistive battery charger detection according to *USB Battery Charging Specification Revision 1.0*, which is now obsolete. With this method, you perform the detection of the DN and DP pin states by using USB1_CHRG_DET_STAT[DN_STATE] and USB1_CHRG_DET_STAT[DP_STATE], respectively. This field must remain 0 for normal USB data communication, or when using the DCD for battery charger detection, according to *USB Battery Charger Specification Revision 1.2* or any other detection mechanism for USB cable plugin. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
30-28 —	Reserved
27 —	Reserved
26 DISCHARGE_VBUS	VBUS Discharge Resistor Controller Enables (controls) a nominal 22 kΩ resistor between the USB1_VBUS pin and the ground that can be used to accelerate the fall of the VBUS signal at the end of a session. 0b - Disables 1b - Enables
25-23 —	Reserved
22-20 PWRUP_CMPS	VBUS_VALID Comparator Enable Enables (powers up) the comparator used for the VBUS_VALID detector. You can reset this field to 0 to save power if the internal VBUS_VALID comparator is not used. The default value is 111b. 000b - Disable the VBUS_VALID comparator 001b - Enable the SESS_VALID comparator 010b - Enable 3V detection
19 —	Reserved
18 VBUSVALID_TO_SESSVALID	VBUS_VALID Comparator Selection Controls the comparator that is used to report the VBUS_VALID results of the VBUS_VALID comparator and session valid comparators, in USB1_VBUS_DETECT[ID_OVERRIDE_EN]. If this field is 0, the VBUS_VALID comparator for the VBUS_VALID results is used. If this field is 1, the session end comparator for the VBUS_VALID results is used. The session end threshold is > 0.8 V and < 4.0 V. The VBUS_VALID comparator is the most accurate and has a programmable threshold that VBUSVALID_THRESH defines. The session valid comparator may be useful in systems using nonstandard VBUS voltages. The mux selection in this field happens before any VBUS_VALID selection controlled by VBUSVALID_SEL and VBUS_SOURCE_SEL. 0b - VBUS_VALID comparator for the VBUS_VALID results

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Session end comparator for the VBUS_VALID results
17-13 —	Reserved
12-11 —	Reserved
10-9 VBUS_SOURCE_SEL	VBUS_VALID Source Selection Selects the source of the VBUS_VALID signal reported to the USB controller. If this field is 0b, by default, the VBUS_VALID comparator results for the signal reported to the USB controller are used. If this field is 1b or 10b, the session valid comparator results for the signal reported to the USB controller are used. The VBUS_VALID source selections in this field take effect only if both VBUSVALID_SEL and VBUS_OVERRIDE_EN are 0. This field does not impact the value of USB1_VBUS_DET_STAT[VBUS_VALID]. 00b - VBUS_VALID comparator results 01b - Session valid comparator results 10b - Session valid comparator result 11b - Reserved; do not use
8 VBUSVALID_SEL	VBUS_VALID Source Selection Selects the source of the VBUS_VALID signal reported to the USB controller. If this field is 0, by default, the VBUS_VALID comparator results for the signal reported to the USB controller are used. If this field is 1, the VBUS_VALID_3V detector results for the signal reported to the USB controller are used. The VBUS_VALID source selection in this field only takes effect if VBUS_VALID is 0 . This field does not impact the value of USB1_VBUS_DET_STAT[VBUS_VALID]. 0b - VBUS_VALID comparator results 1b - VBUS_VALID_3V detector results
7 VBUSVALID_OVERRIDE	Override Value for the VBUS_VALID Signal Provides the value of USB1_VBUS_DET_STAT[VBUS_VALID] reported to the USB controller if the value of VBUS_OVERRIDE_EN is 1. This field does not impact the value of USB1_VBUS_DET_STAT[VBUS_VALID]. 0b - Overridden to 0 1b - Overridden to 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
6 AVALID_OVERRIDE	Override Value for A-Device Session Valid Provides the value for USB1_VBUS_DET_STAT[AVALID] if the value of VBUS_OVERRIDE_EN is 1. 0b - Overridden to 0 1b - Overridden to 1
5 BVALID_OVERRIDE	Override Value for B-Device Session Valid Provides the value for USB1_VBUS_DET_STAT[BVALID] if the value of VBUS_OVERRIDE_EN is 1. 0b - Overridden to 0 1b - Overridden to 1
4 SESSEND_OVERRIDE	Override Value for SESSEND Provides the value for USB1_VBUS_DET_STAT[SESSEND] if the value of VBUS_OVERRIDE_EN is 1. 0b - Overridden to 0 1b - Overridden to 1
3 VBUS_OVERRIDE_EN	VBUS Detect Signal Override Enable Allows you to override the results from VBUS_VALID and session valid comparators using the values in USB1_VBUS_DETECT[7:4]. If this field is 0, by default, the results of the internal VBUS_VALID and session valid comparators for VBUS_VALID, AVALID, BVALID, and SESSEND are used. If this field is 1, the override values for VBUS_VALID, AVALID, BVALID, and SESSEND are used. The VBUS_VALID, AVALID, BVALID, and SESSEND signals sent to the USB controller are affected by these bit selections. The values reported for USB1_VBUS_DET_STAT[AVALID], USB1_VBUS_DET_STAT[BVALID], and USB1_VBUS_DET_STAT[SESSEND] are also affected but the value of USB1_VBUS_DET_STAT[VBUS_VALID] is not affected. This override method may be useful if you do not perform VBUS detection using the internal VBUS_VALID or session end comparators. 0b - Results of VBUS_VALID and session valid comparators for VBUS_VALID, AVALID, BVALID, and SESSEND 1b - Override values for VBUS_VALID, AVALID, BVALID, and SESSEND
2-0 VBUSVALID_THRESH	VBUSVALID Threshold Sets the threshold for the VBUSVALID comparator, which is the most accurate method to determine the presence of 5 V, and includes hysteresis to minimize the need for software debounce of the detection. This comparator has ~50 mV of hysteresis to prevent chattering at the comparator trip point. 000b - 4.0 V 001b - 4.1 V

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010b - 4.2 V 011b - 4.3 V 100b - 4.4 V 101b - 4.5 V 110b - 4.6 V 111b - 4.7 V

52.7.13 USBPHY VBUS Detector Status (USB1_VBUS_DET_STAT)

Offset

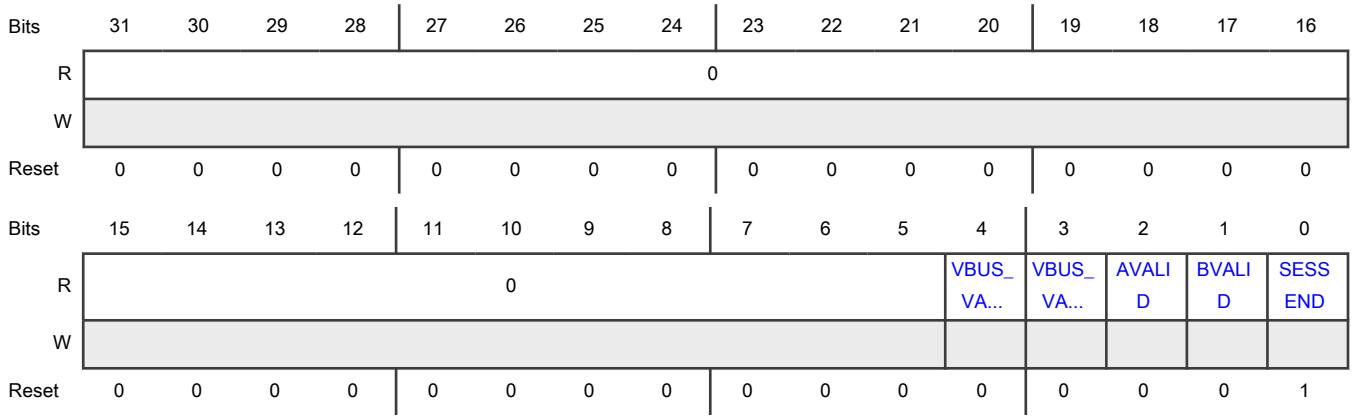
Register	Offset
USB1_VBUS_DET_STAT	D0h

Function

Allows status observation for USB VBUS detect functions.

The APB clock synchronizes the values reported in this register.

Diagram



Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 VBUS_VALID_3V	VBUS_VALID_3V Detector Status Indicates the status of the VBUS_VALID_3V detector. If this field is 0, the VBUS voltage is below the VBUS_VALID_3V threshold. If this field is 1, the VBUS voltage is above the VBUS_VALID_3V threshold. The VBUS_VALID_3V detector has a lower threshold for the voltage on the USB1_VBUS pin than either the session valid or VBUS_VALID comparators. This signal may be useful for applications using a nonstandard VBUS voltage. 0b - Below threshold 1b - Above threshold
3 VBUS_VALID	VBUS Voltage Status Indicates the result of VBUS_VALID detection for the USB1_VBUS pin. If this field is 0, the VBUS voltage is below the comparator threshold. If this field is 1, the VBUS voltage is above the comparator threshold. USB1_VBUS_DETECT[VBUSVALID_TO_SESSVALID] selects the VBUS_VALID comparator used. You cannot overwrite the VBUS_VALID source, which is not affected by the values of USB1_VBUS_DETECT[VBUSVALID_OVERRIDE] and USB1_VBUS_DETECT[VBUS_OVERRIDE_EN]. 0b - Below threshold 1b - Above threshold
2 AVALID	A-Device Session Valid Status Indicates A-device session valid status that the session valid comparator determines. If this field is 0, the VBUS voltage is below the session valid threshold. If this field is 1, the VBUS voltage is above the session valid threshold. The voltage on the USB1_VBUS pin determines the default value of this field. You can overwrite this field by using USB1_VBUS_DETECT[AVALID_OVERRIDE] and USB1_VBUS_DETECT[VBUS_OVERRIDE_EN]. 0b - Below threshold 1b - Above threshold
1 BVALID	B-Device Session Valid Status Indicates B-device session valid status that the session valid comparator determines. If this field is 0, the VBUS voltage is below the session valid threshold. If this field is 1, the VBUS voltage is above the session valid threshold. The voltage on the USB1_VBUS pin determines the default value of this field. You can overwrite this field by using USB1_VBUS_DETECT[BVALID_OVERRIDE] and USB1_VBUS_DETECT[VBUS_OVERRIDE_EN].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Below threshold 1b - Above threshold
0 SESSEND	Session End Indicator Indicates session end status, the value inverted from the session valid comparator. If this field is 0, the VBUS voltage is above the session valid threshold. If this field is 1, the VBUS voltage is below the session valid threshold. The voltage on the USB1_VBUS pin determines the default value of this field. You can overwrite this field by using USB1_VBUS_DETECT[SESSEND_OVERRIDE] and USB1_VBUS_DETECT[VBUS_OVERRIDE_EN]. 0b - Above threshold 1b - Below threshold

52.7.14 USBPHY Charger Detect Control (USB1_CHRG_DETECT)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
USB1_CHRG_DETECT	E0h	USBPHY Charger Detect Control
USB1_CHRG_DETECT_SET	E4h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_CHRG_DETECT is 1
USB1_CHRG_DETECT_CLR	E8h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_CHRG_DETECT is 0
USB1_CHRG_DETECT_TOG	ECh	Writing 1 to a bit in this register inverts the value of the corresponding bit in USB1_CHRG_DETECT

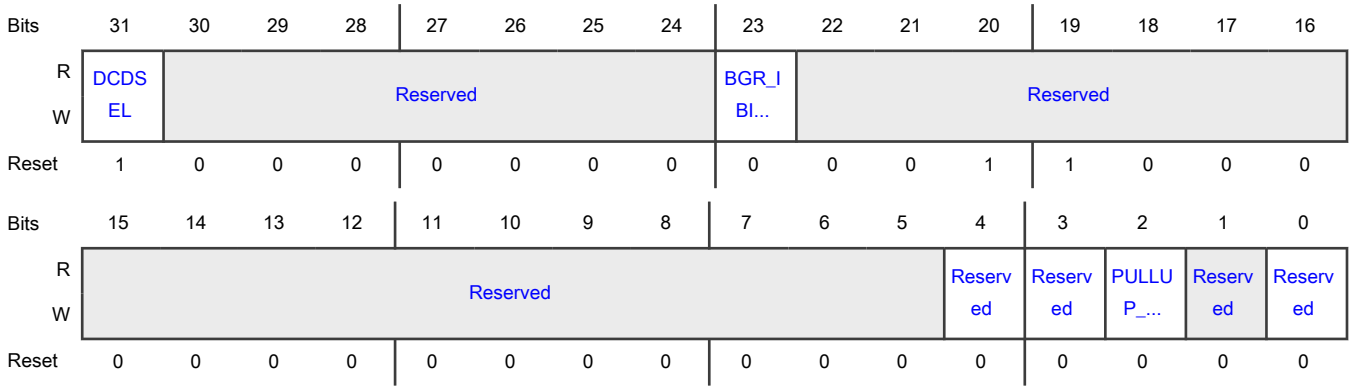
Function

Defines controls for USB battery charging detection functions.

USBDCD controls the USB battery charging detection functions (see the "USB Device Charger Detection Module (USBDCD)" chapter for more information).

You must not change the values of this field except for characterization tasks.

Diagram



Fields

Field	Function
31 DCDSEL	DCD Selection Selects the control source for battery charging detection. If this field is 0, USBPHY Charger Detect Control (USB1_CHRG_DETECT) controls the BC 1.2 functionality. If this field is 1, fields and state machines in USBHSDCD control the BC 1.2 functionality (see the "USB Device Charger Detection Module (USBDCD)" chapter for more information). You can control several of the USBPHY analog circuits associated with the USB battery charging v1.2 functions by either using control fields in USBPHY Charger Detect Control (USB1_CHRG_DETECT) or by using the USBHSDCD module's registers. Use of USBHSDCD is recommended for software simplification, for device mode charger detection because USBHSDCD can auto-sequence through the detection states. 0b - Fields in the USB1_CHRG_DETECT register 1b - Fields and state machines in the USBHSDCD module
30-24 —	Reserved
23 BGR_IBIAS	BGR Bias Selects the bias method deployed for the voltage and current sources used for the analog circuits associated with USB battery charging detection in the USBHSDCD instantiation of USBDCD (see the "USB Device Charger Detection Module (USBDCD)" chapter for more information). If this field is 0, by default, the local bias powered from VBUS for 10 µA reference is used. If this field is 1, the bandgap bias powered from VREGIN0 or VREGIN1 for 10 µA reference is used. The V_{DP_SRC} and V_{DN_SRC} bias is generated from the USB1_VBUS voltage, independent of the value of this field. 0b - Local bias 1b - Bandgap bias
22-5	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
4 —	Reserved The value of this reserved bit must remain 0b.
3 —	The value of this reserved bit must remain 0b.
2 PULLUP_DP	DP Pullup Resistor Enable Override Control Enables FS Device mode operation that is normally controlled using the UTMI bus signals that the USB controller sends to USBPHY. If this field is 1, the pullup resistor is enabled independently of the UTMI signals to the PHY. The USB software stack uses this capability during charger type detection, according to the obsolete *USB Battery Charging Specification Revision 1.1*. For USB battery charging detection and normal USB data communication (associated with the USB battery charging v1.2), retain 0 as the value of this field. 0b - Disables 1b - Enables
1 —	Reserved
0 —	This value of this bit must remain 0b to avoid interference with USB battery charging detection.

52.7.15 USBPHY Charger Detect Status (USB1_CHRG_DET_STAT)

Offset

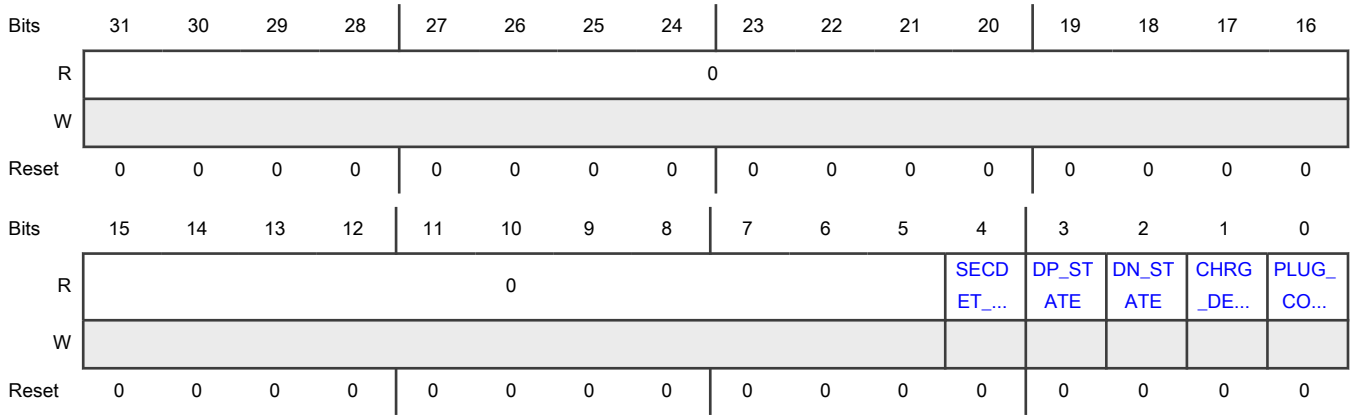
Register	Offset
USB1_CHRG_DET_STAT	F0h

Function

Indicates the USBPHY charger detect status.

Use the USBHSDCD registers for USB battery charging detection functions and standards-based charger detection purposes.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 SECDET_DCP	Battery Charging Secondary Detection Phase Output Indicates the type of charging port detected—charging downstream port (CDP) or downstream charging port (DCP)—during the USB battery charging secondary detection phase, using the USBHSDCD module. 0b - CDP detected 1b - DCP detected
3 DP_STATE	DP State Indicates the single-ended receiver output for the DP pin, from charger detection circuits. 0b - < 0.8 V 1b - > 2.0 V
2 DN_STATE	DN State Indicates the single-ended receiver output for the DN pin, from charger detection circuits. 0b - < 0.8 V 1b - > 2.0 V
1 CHRG_DETECTED	Battery Charging Primary Detection Phase Output Indicates whether a standard downstream port (SDP) or charging port is detected during the USB battery charging primary detection phase. 0b - SDP detected 1b - Charging port detected
0	Battery Charging Data Contact Detection Phase Output

Table continues on the next page...

Table continued from the previous page...

Field	Function
PLUG_CONTACT	Indicates whether an attached USB cable is detected between the remote host and the local device during the data contact detection phase, according to *USB Battery Charging Specification Revision 1.2* (using the USBHSDCD module). 0b - Not detected 1b - Detected

52.7.16 USBPHY Analog Control (ANACTRL)

Offset

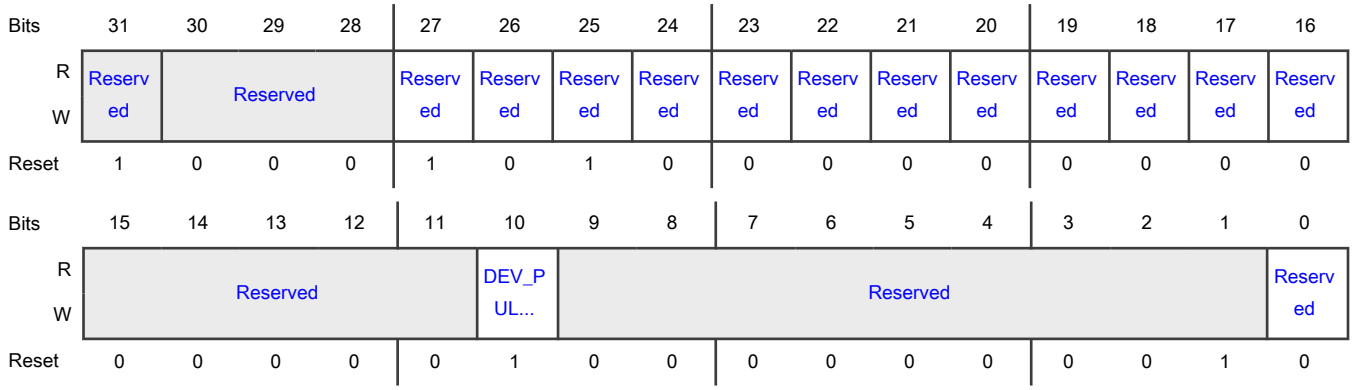
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
ANACTRL	100h	USBPHY Analog Control
ANACTRL_SET	104h	Writing 1 to a bit in this register ensures that the corresponding bit in ANACTRL is 1
ANACTRL_CLR	108h	Writing 1 to a bit in this register ensures that the corresponding bit in ANACTRL is 0
ANACTRL_TOG	10Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in ANACTRL

Function

Includes fields for adding pre-emphasis to the USB HS TX output drivers.

Diagram



Fields

Field	Function
31 —	Reserved
30-28 —	Reserved
27 —	This field must remain at value 1b.
26 —	This field must remain at value 0b.
25 —	This field must remain at value 1b.
24 —	This field must remain at value 0b.
23 —	This field must remain at value 0b.
22 —	This field must remain at value 0b.
21 —	This field must remain at value 0b.
20 —	This field must remain at value 0b.
19 —	This field must remain at value 0b.
18 —	This field must remain at value 0b.
17 —	This field must remain at value 0b.
16 —	This field must remain at value 0b.
15-11	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
10 DEV_PULLDOWN	Device Pulldown Enables the 15 kΩ pulldown resistors on both DP and DN pins in Device mode. You can use this feature in Device mode, while the USB cable is disconnected, to retain the data pins at known values, avoiding unnecessary interrupts from single-ended receivers. This field must be reset to 0 during normal USB data communication in Device mode, or during battery charger detection, using the USBHSDCD module. 0b - Disables 1b - Enables
9-1 —	Reserved
0 —	The value of this reserved bit must remain 0b for normal operation.

52.7.17 USBPHY Loopback Control and Status (USB1_LOOPBACK)

Offset

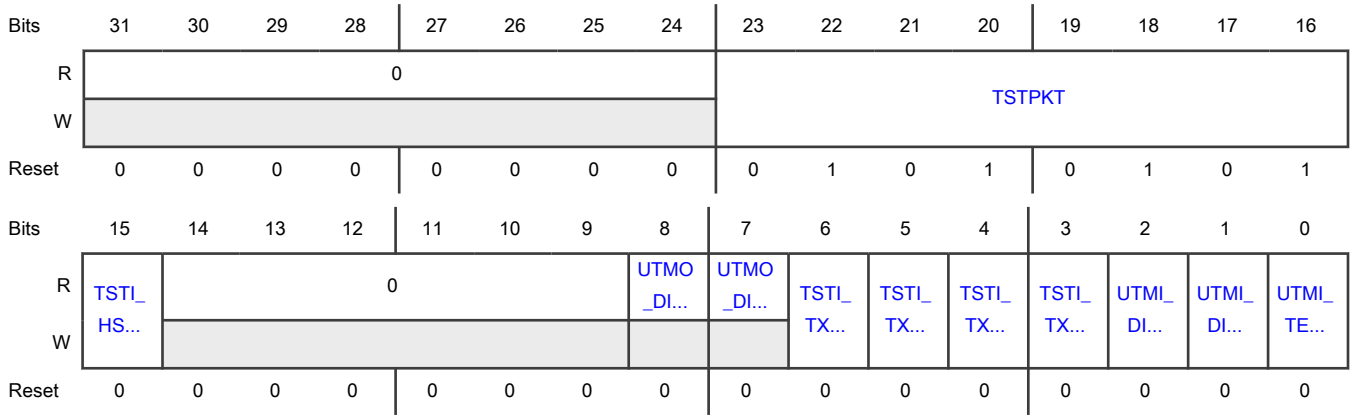
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
USB1_LOOPBACK	110h	USBPHY Loopback Control and Status
USB1_LOOPBACK_SET	114h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_LOOPBACK is 1
USB1_LOOPBACK_CLR	118h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_LOOPBACK is 0
USB1_LOOPBACK_TOG	11Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in USB1_LOOPBACK

Function

Controls loopback testing for USBPHY. Loopback mode is for test purposes only; you cannot use it for normal USB data communication.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 TSTPKT	Testing Packet Selects the packet data byte used for USB loopback testing in Pulse mode. Pulse mode is selected if you write 01b to UTMI_DIG_TST0 and UTMI_DIG_TST1. The default value produces a data inversion at each unit interval.
15 TSTI_HSFS_MODE_EN	Loopback Test HS-FS Mode Enable Enables the loopback test to dynamically change the packet speed. It sends a number of HS packets determined by USB1_LOOPBACK_HSFSCNT[TSTI_HS_NUMBER], followed by a number of FS packets determined by USB1_LOOPBACK_HSFSCNT[TSTI_FS_NUMBER]. 0b - Disables 1b - Enables
14-9 —	Reserved
8 UTMO_DIG_TST1	UTMO Digital Test 1 Indicates USB loopback test results. Test results are only valid when UTMI_TESTSTART is 1. 0b - Not passing 1b - Passing
7 UTMO_DIG_TST0	UTMO Digital Test 0 Indicates USB loopback test results. Test results are only valid when UTMI_TESTSTART is 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Passing 1b - Not passing
6 TSTI_TX_HIZ	Loopback Test Transmit Hi-Z Enables TX Hi-Z for USB loopback test. 0b - Disables 1b - Enables
5 TSTI_TX_EN	Loopback Test Transmit Enable Enables TX for USB loopback test. 0b - Disables 1b - Enables
4 TSTI_TX_LS_MODE	Loopback Test LS Mode Selects the TX mode. 0b - HS or FS (defined by TSTI1_TX_HS) 1b - LS
3 TSTI_TX_HS_MODE	Loopback Test HS Mode Selects HS or FS mode for USB loopback testing. 0b - FS 1b - HS
2 UTMI_DIG_TST1	UTMI Digital Test 1 Selects the mode for USB loopback test 1. If this field is 0, Pulse mode for the loopback test is selected. This field remains 0 when UTMI_DIG_TST0 is 1. If this field is 1, Pseudorandom mode for the loopback test is selected. This field becomes 1 when UTMI_DIG_TST0 is 0. 0b - Pulse mode 1b - Pseudorandom mode
1 UTMI_DIG_TST0	UTMI Digital Test 0 Selects the mode for USB loopback test 0. If this field is 0, Pseudorandom mode for the loopback test is selected. This field remains 0 when UTMI_DIG_TST1 is 1. If this field is 1, Pulse mode for the loopback test is selected. This field becomes 1 when UTMI_DIG_TST1 is 0. 0b - Pseudorandom mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Pulse mode
0 UTMI_TESTSTART	UTMI Test Start Enables the USB loopback test. 0b - Disables 1b - Enables

52.7.18 USBPHY Loopback Packet Number Selection (USB1_LOOPBACK_HSFSCNT)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

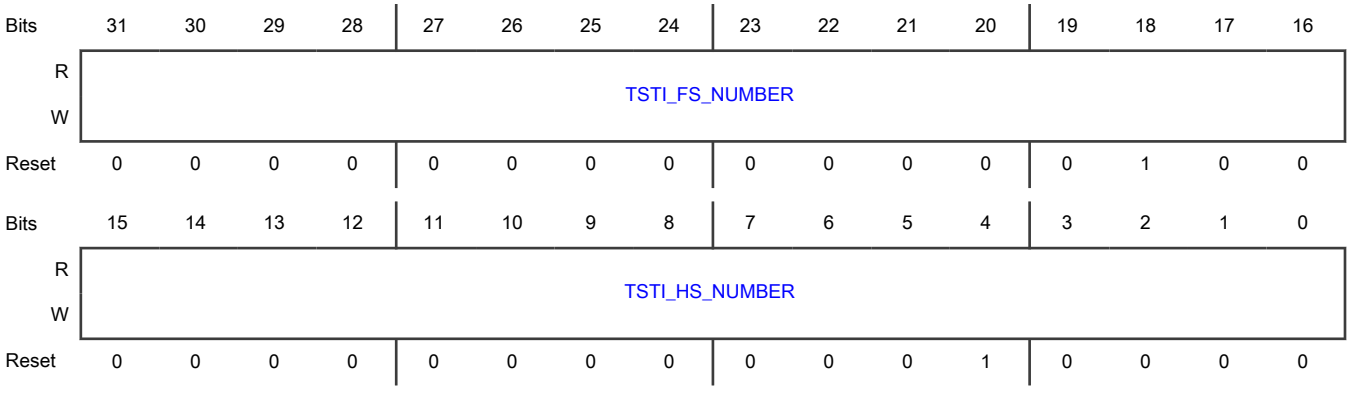
Register	Offset	Description
USB1_LOOPBACK_HSFSCNT	120h	USBPHY Loopback Packet Number Selection
USB1_LOOPBACK_HSFSCNT_SET	124h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_LOOPBACK_HSFSCNT is 1
USB1_LOOPBACK_HSFSCNT_CLR	128h	Writing 1 to a bit in this register ensures that the corresponding bit in USB1_LOOPBACK_HSFSCNT is 0
USB1_LOOPBACK_HSFSCNT_TOG	12Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in USB1_LOOPBACK_HSFSCNT

Function

Selects the packet counts used for the specific loopback test mode that dynamically alternates the transmit speed.

See USBPHY Loopback Control and Status (USB1_LOOPBACK) for other configuration of this mode.

Diagram



Fields

Field	Function
31-16 TSTI_FS_NUMBER	Loopback Test FS Packet Number Specifies the FS packet number, used when USB1_LOOPBACK[TSTI_HSFS_MODE_EN] is 1.
15-0 TSTI_HS_NUMBER	Loopback Test HS Packet Number Specifies the HS packet number, used when USB1_LOOPBACK[TSTI_HSFS_MODE_EN] is 1.

52.7.19 USBPHY Trim Override Enable (TRIM_OVERRIDE_EN)

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TRIM_OVERRIDE_EN	130h	USBPHY Trim Override Enable
TRIM_OVERRIDE_EN_SET	134h	Writing 1 to a bit in this register ensures that the corresponding bit in TRIM_OVERRIDE_EN is 1
TRIM_OVERRIDE_EN_CLR	138h	Writing 1 to a bit in this register ensures that the corresponding bit in TRIM_OVERRIDE_EN is 0
TRIM_OVERRIDE_EN_TOG	13Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TRIM_OVERRIDE_EN

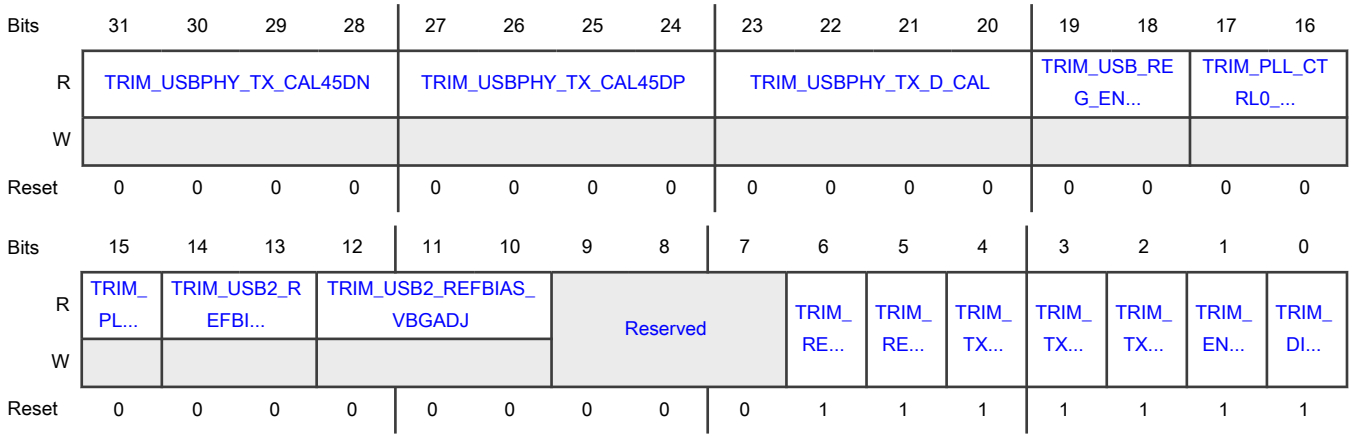
Function

Allows observation of the default IFR settings of the PHY parameters for the following fields:

- TRIM_USBPHY_TX_CAL45DN
- TRIM_USBPHY_TX_CAL45DP
- TRIM_USBPHY_TX_D_CAL
- TRIM_USB_REG_ENV_TAIL_ADJ_VD
- TRIM_PLL_CTRL0_DIV_SEL

Additional fields also determine whether you can overwrite those values by using the settings defined in USBPHY Transmitter Control (TX), UTMI Debug Status 1 (DEBUG1), and USBPHY PLL Control and Status (PLL_SIC).

Diagram



Fields

Field	Function
31-28 TRIM_USBPHY_TX_CAL45DN	DN Series Termination Resistance Trim Bits from Outside USBPHY Trims the termination resistance if TRIM_TX_CAL45DN_OVERRIDE is 0. If TRIM_TX_CAL45DN_OVERRIDE is 1, the resistance trim values in TXCAL45DN are used instead. Decode to trim the nominal 45 Ω series termination resistance to the DN output pin, with values loaded using signals outside USBPHY. The methods used to load the read-only values into this field during reset and boot are chip-specific. The maximum resistance is at value 0000b and minimum resistance is at value 1111b (resistance is centered at value 0111b). For USBPHY, each incrementally increasing trim value decreases the target resistance by approximately 3.5%, compared to the next lower value. Trimming this resistance impacts both the overshoot and undershoot of the FS TX output and the amplitude of the HS TX output. The methods used to load the read-only values into this field during reset and boot are chip-specific.
27-24 TRIM_USBPHY_TX_CAL45DP	DP Series Termination Resistance Trim Bits from Outside USBPHY Trims the termination resistance when TRIM_TX_CAL45DP_OVERRIDE is 0. If TRIM_TX_CAL45DP_OVERRIDE is 1, the resistance trim values in TX[TXCAL45DP] are used instead. Decode to trim the nominal 45 Ω series termination resistance to the DP output pin, with values loaded using signals outside USBPHY. The methods used to load the read-only values into this field during reset and boot are chip-specific. The maximum resistance is 0000b and minimum resistance is 1111b (the resistance is centered at value 0111b). For USBPHY, each incrementally increasing trim value decreases the target resistance by approximately 3.5%, compared to the next lower value. Trimming this resistance impacts both the overshoot and undershoot of the FS TX output and amplitude of the HS TX output. The methods used to load the read-only values into this field during reset and boot are chip-specific.
23-20	HS TX Output Current Trim Bits from Outside USBPHY

Table continues on the next page...

Table continued from the previous page...

Field	Function
TRIM_USBPHY_TX_D_CAL	Trims the output current if TRIM_TX_D_CAL_OVERRIDE is 0. In this case, the current trim values in TX[D_CAL] are used instead. Decode to trim the nominal 17.78 mA current source for the HS TX drivers on DP and DN, with values loaded using signals from outside the PHY. This current is directly proportional to the amplitude of the HS TX eye diagram. The methods used to load the read-only values into this field during reset and boot are chip-specific. 0000b - Maximum current; approximately 19% above nominal 0111b - Nominal 1111b - Minimum current; approximately 19% below nominal
19-18 TRIM_USB_REG_ENV_TAIL_ADJ_VD	HS RX Squelch Rise Time Delay Trim Bits from Outside USBPHY Allows you to adjust the squelch rise time if TRIM_ENV_TAIL_ADJ_VD_OVERRIDE is 0. In this case, the squelch rise time trim values in DEBUG1[ENTAILADJVD] are used instead. This field defines the delay adjustment for the rise time of the squelch signal from the HS RX envelope detector, with values loaded using signals from outside the PHY. The methods used to load the read-only values into this field during reset and boot are chip-specific. See DEBUG1[ENTAILADJVD] to understand the available trims.
17-15 TRIM_PLL_CTRL0_DIV_SEL	PLL Divider Value Configuration Bits from Outside USBPHY Controls the PLL feedback loop divider if TRIM_DIV_SEL_OVERRIDE is 0. USBPHY's PLL produces a 480 MHz output clock. This field allows the use of different frequency signals for the PLL reference clock input connected to the OSCCLK signal from the system oscillator, with values loaded from outside USBPHY. If TRIM_DIV_SEL_OVERRIDE is 0, the PLL divider values in PLL_SIC[PLL_DIV_SEL] are used instead. See the field value descriptions for PLL_SIC[PLL_DIV_SEL] to understand the available input frequency settings. The methods used to load the read-only values into this field during reset and boot are chip-specific.
14-13 TRIM_USB2_REFBIAS_TST	Bias Current Control Adjustment Bits from Outside USBPHY Trims the PHY's current reference when TRIM_REFBIAS_TST_OVERRIDE is 0. If TRIM_REFBIAS_TST_OVERRIDE is 1, the current trim values in DEBUG1[USB2_REFBIAS_TST] are used instead. This field defines the main current trim for the USBPHY bias module current reference generator, with values loaded using signals from outside the PHY.
12-10 TRIM_USB2_REFBIAS_VBGADJ	Bandgap Voltage Adjustment Bits from Outside USBPHY Trims the PHY's bandgap voltage if TRIM_REFBIAS_VBGADJ_OVERRIDE is 0. In this case, the voltage trim values in DEBUG1[USB2_REFBIAS_VBGADJ] are used instead. This field defines voltage adjustment trim on the USBPHY bias module bandgap generator, with values loaded using signals from outside the PHY.

Table continues on the next page...

Table continued from the previous page...

Field	Function
9-7 —	The value of this reserved bit must remain 0b.
6 TRIM_REFBIAS_TST_OVERRIDE	Override Enable for Bias Current Control Enables override for bias current control. If this field is 1, the value of DEBUG1[USB2_REFBIAS_TST] is used. 0b - Disables 1b - Enables
5 TRIM_REFBIAS_VBGADJ_OVERRIDE	Override Enable for Bandgap Voltage Adjustment Enables override for bandgap voltage adjustment. If this field is 1, the value of DEBUG1[USB2_REFBIAS_VBGADJ] is used. 0b - Disables 1b - Enables
4 TRIM_TX_CAL45DN_OVERRIDE	Override Enable for DN Series Termination Trim Enables override for the DN series termination trim. If this field is 1, the value of TX[TXCAL45DN] is used. 0b - Disables 1b - Enables
3 TRIM_TX_CAL45DP_OVERRIDE	Override Enable for DP Series Termination Trim Enables override for the DP series termination trim. If this field is 1, the value of TX[TXCAL45DP] is used. 0b - Disables 1b - Enables
2 TRIM_TX_D_CAL_OVERRIDE	Override Enable for the HS TX Output Current Trim Enables override for HS TX output current trim. If this field is 1, the value of TX[D_CAL] is used. 0b - Disables 1b - Enables
1 TRIM_ENV_TAIL_ADJ_VD_OVERRIDE	Override Enable for HS RX Squelch Rise Time Delay Trim Enables override for the HS RX squelch rise time delay trim. If this field is 1, the value of DEBUG1[ENTAILADJVD] is used. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
0 TRIM_DIV_SEL_OVERRIDE	Override Enable for PLL Divider Value Enables override for the PLL divider value. If this field is 1, the value of PLL_SIC[PLL_DIV_SEL] is used. 0b - Disables 1b - Enables

Chapter 53
USB Device Charger Detection Module (USBDCD)

53.1 Chip-specific USBDCD information

Table 509. Reference links to related information

Topic	Related module	Reference
Full description	USBDCD	USBDCD
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

53.1.1 Module instances

This device contains one instances of the USBDCD module.

Table 510. USBDCD instance

	USBDCD
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
Controls device charger detect on	USB0

53.1.2 Reset control

Reset control is the same as USBPHY0, see RSTCTL_PRSTCTL1 and PRSTCTL1_SET.

53.1.3 Clock control

The media_main_clock is the clock input for USBDCD, and CLKCTL4_PSCCTL1_MEDIA[USB0] is used to enable or disable this clock.

See .

53.2 Overview

USBDCD works with the USB transceiver to detect whether the USB device is attached to a Charging Port, either a Dedicated Charging Port (DCP) or a Charging Downstream Port (CDP). System software coordinates the detection activities of the module and controls an off-chip integrated circuit that performs the battery charging.

The use of the term "USB transceiver" in this module documentation applies to the USB physical layer instance on the chip, whether it is an Full Speed (FS)/Low Speed (LS) only transceiver or an High Speed (HS)/FS/LS capable PHY.

53.2.1 Block diagram

The following figure is a high-level block diagram of the module.

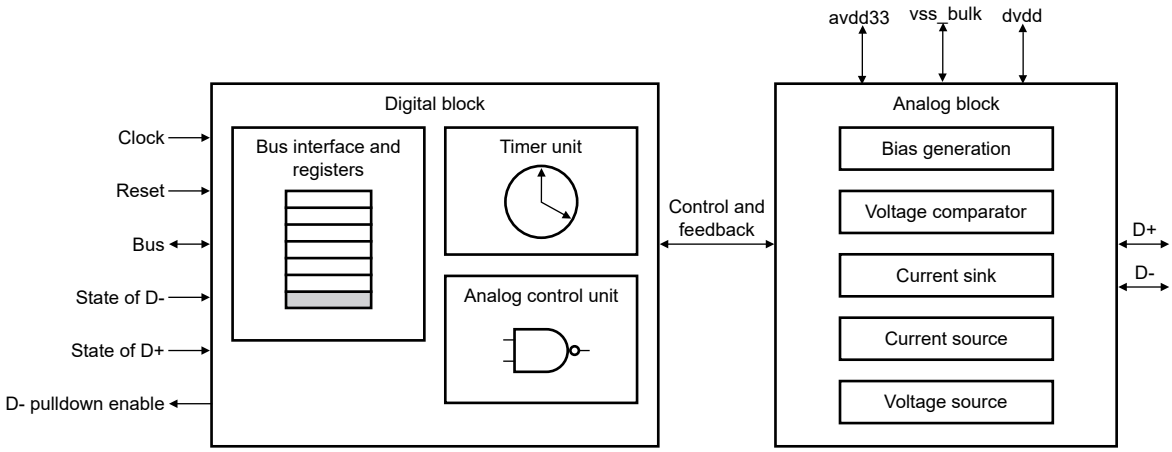


Figure 181. Block diagram

USBDCD consists of two main blocks:

- A Digital block provides the programming interface (memory-mapped registers) and includes the Timer unit and the Analog control unit.
- An Analog block provides the circuitry for the physical detection of the charger that includes the Bias generation, Voltage source, Current source, Current sink, and Voltage comparator circuitry. This block also contains necessary muxes between the source/sink circuits and the D- and D+ pins.

53.2.2 Features

- Compliant with the latest industry-standard specifications: *USB Battery Charging Specification, Revisions 1.1 and 1.2*
- Supports programmable timing parameters:
  - By default set to values required by the industry standards to allow easy configuration. You only need to set the clock frequency before enabling the module.
  - Updates of the standards are programmable.

53.3 Functional description

Several hardware components are involved in the sequence of detecting the presence of the charging port and type of the charging port. System software coordinates this activity. This collection of interacting hardware and software is called USB battery charging subsystem. The following figure shows USBDCD as a component of the subsystem.

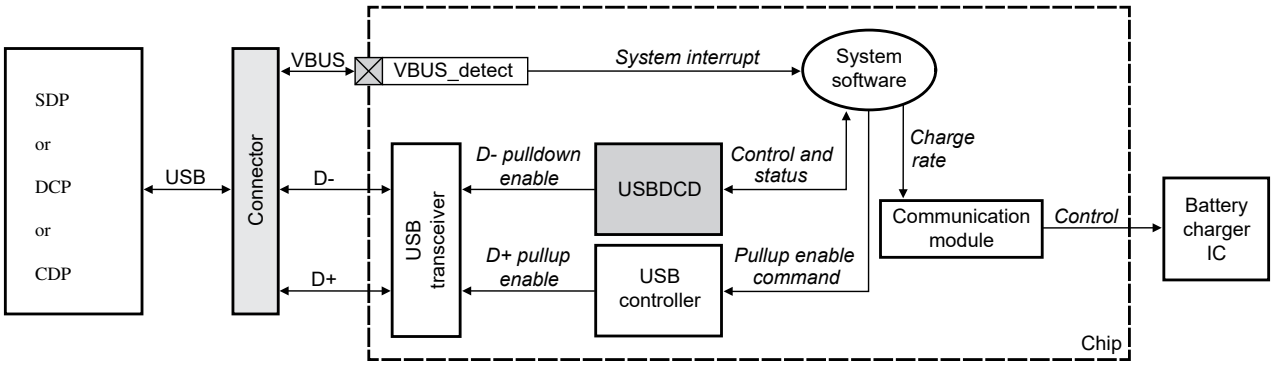


Figure 182. USB battery charging subsystem

The following table describes the components.

Table 511. USB battery charger subsystem components

Component	Description
Battery Charger IC	The external battery charger IC regulates the charge rate to the rechargeable battery. System software is responsible for communicating the appropriate charge rates.

Charger	Maximum current drawn
Standard Downstream Port (SDP)	up to 500 mA (I_{CFG_MAX})[1]
CDP	up to 1500 mA (I_{DEV_CHG})
DCP	up to 1500 mA (I_{DEV_CHG})

1. If a USB SDP host has suspended the USB device, the current drawn by the device has to follow the rules of the Dead Battery Provision in the *USB Battery Charging Specification, Rev. 1.2*.

Comm Module	A communications module on the device can be used to control the charge rate of the battery charger IC.
System software	Coordinates the detection activities of the subsystem.
USB Controller	The D+ pullup enable control signal plays a role during the charger type detection phase. System software must issue a command to the USB controller to assert this signal. After this pullup is enabled, the device is considered to be connected to the USB bus. The host then attempts to enumerate it. **NOTE** In general, the USB controller is used only for USB device applications when using USBDCD. For USB host applications, USBDCD must be disabled.
USB Transceiver	The USB transceiver contains the pullup resistor for the USB D+ signal and the pulldown resistors for the USB D+ and D– signals. The D+ pullup and the D– pulldown are both used during the charger detection sequence in BC1.1, but it is not used during charger detection in

Table continues on the next page...

Table 511. USB battery charger subsystem components (continued)

Component	Description
	BC1.2. The USB transceiver also outputs the digital state of the D+ and D– signals from the USB bus. The pullup and pulldown enable signals are controlled by other modules during the charger detection sequence in BC1.1: The D+ pullup enable is output from the USB controller and is under software control. USBDCD controls the D– pulldown enable.
USBDCD	Detects whether the device has been plugged into either an SDP, a CDP, or a DCP.
VBUS_detect	This interrupt pin connected to the USB VBUS signal detects when the device has been plugged into or unplugged from the USB bus. If the system requires waking up from a low-power mode on being plugged into the USB port, this interrupt should also be a low-power wakeup source. If this pin multiplexes other functions, such as GPIO, the pin can be configured as an interrupt so that the USB plug or unplug event can be detected.

53.3.1 Modes of operation

The operating modes of the USBDCD module are shown in the following table.

Table 512. Module modes and their conditions

Module mode	Description	Conditions when used
Enabled for Charger Detection	The module performs the charger detection sequence.	System software must enable the module only if all of the following conditions are true: • The system is using a rechargeable battery or is otherwise capable of being powered up by an SDP or a Charging Port. • The device is being used in a USB device application. • The device is attached to the USB cable.
Enabled for Signal Overrides	Module bias circuits are enabled for special signaling on DP/DM pins.	See SIGNAL_OVERRIDE[PS].
Disabled	The module is not active and is held in a low-power state.	System software must disable the module when either of the following conditions is true: • The charger detect sequence is complete. • The conditions for being enabled are not met.
Powered Off	The digital supply voltage dvdd is removed.	Low system performance requirements allow putting the device into a very low-power stop mode.

Operating mode transitions are shown in the following table.

Table 513. Entering and exiting module modes

Module mode	Entering	Exiting	Mode after exiting
Enabled	Set CONTROL[START].	Set CONTROL[SR]. [1]	Disabled
Disabled	Take *either* of the following actions: • Set CONTROL[SR].[1] • Reset the module. By default, the module is disabled.	Set CONTROL[START].	Enabled
Powered Off	Perform the following actions: 1. Put the device into very low-power stop mode. 2. Adjust the supply voltages.	Perform the following actions: 1. Restore the supply voltages. 2. Take the device out of very low-power stop mode.	Disabled

1. The effect of setting the CONTROL[SR] field is immediate; that is, the module is disabled even if the sequence has not completed.

53.3.2 The charger detection sequence

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the *USB Battery Charging Specification, Rev. 1.2*.

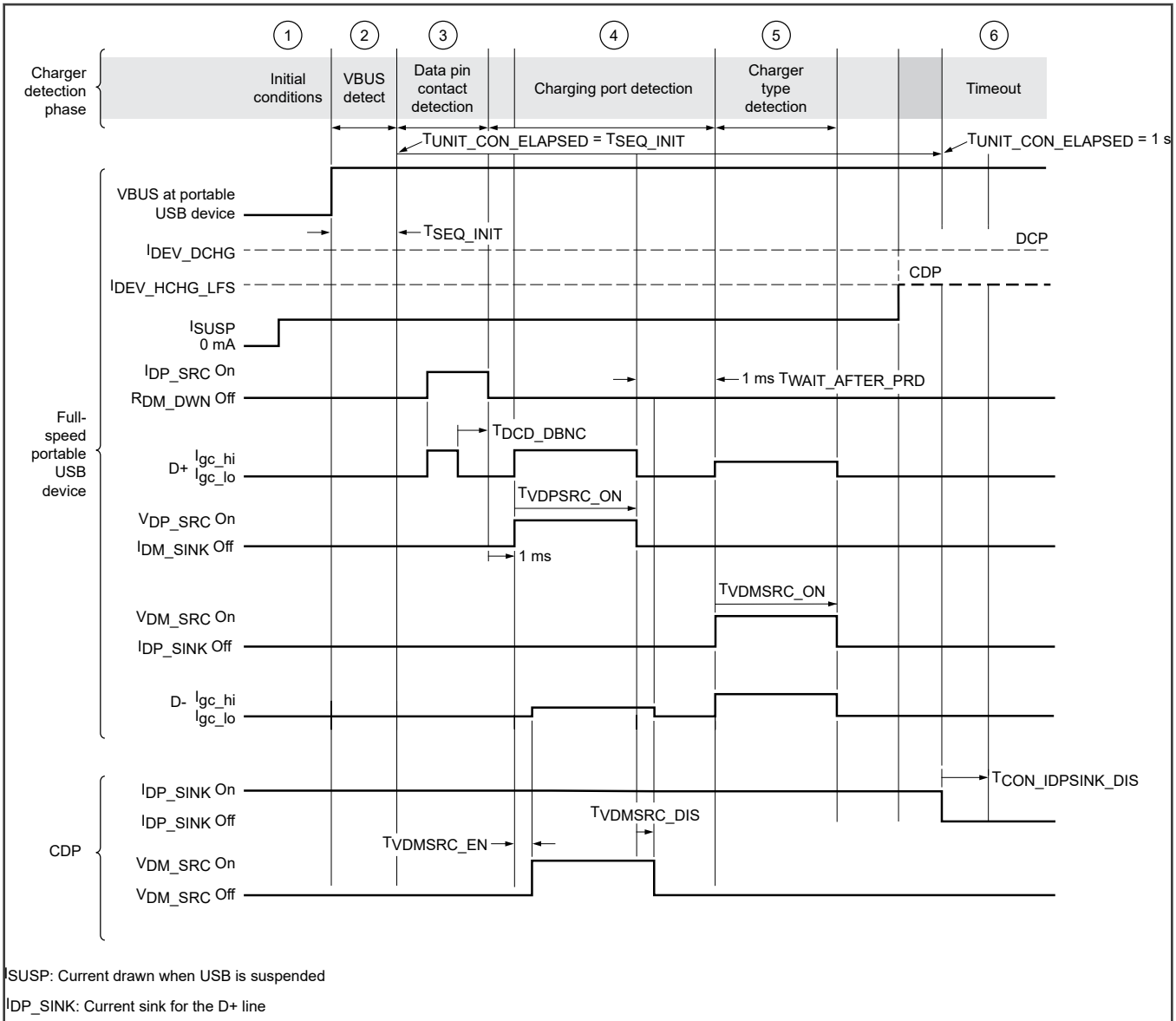


Figure 183. FS charger detection timing for BC1.2

Timing parameter values used in this module for BC1.2 are listed in the following table.

Table 514. Timing parameters for the charger detection sequence for BC1.2

Parameter	USB Battery Charging Spec	Module default	Module programmable range
T_{DCD_DBNC} [1]	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
$T_{WAIT_AFTER_PRD}$	N/A	1 ms	0– 1023 ms
T_{VDMSRC_ON}	40 ms	40 ms	0 –1023 ms

Table continues on the next page...

Table 514. Timing parameters for the charger detection sequence for BC1.2 (continued)

Parameter	USB Battery Charging Spec	Module default	Module programmable range
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON} [1]	1 s	N/A	N/A
T_{VDMSRC_EN} [1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS} [1]	0 – 20 ms	From the USB host	N/A
$T_{CON_IDPSINK_DIS}$ [1]	0 – 10 ms	From the USB host	N/A

1. This parameter is defined by the *USB Battery Charging Specification, Rev. 1.2*.

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the *USB Battery Charging Specification, Rev. 1.1*.

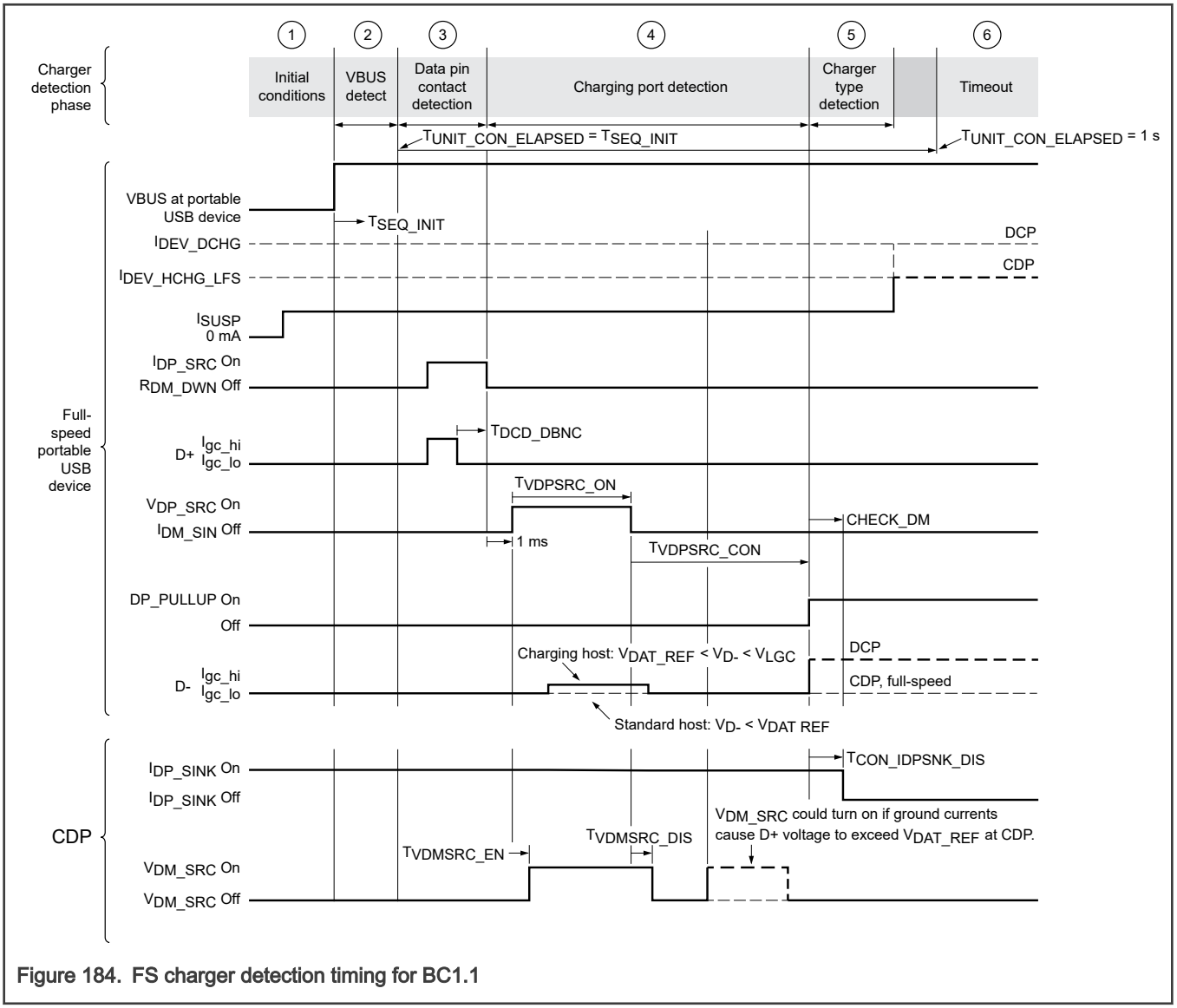


Figure 184. FS charger detection timing for BC1.1

Timing parameter values used in this module for BC1.1 are listed in the following table.

Table 515. Timing parameters for the charger detection sequence for BC1.1

Parameter	USB Battery Charging Spec	Module default	Module programmable range
T_{DCD_DBNC} [1]	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
T_{VDPSRC_CON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
CHECK_DM	N/A	1 ms	0– 15 ms
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON} [1]	1 s	N/A	N/A
T_{VDMSRC_EN} [1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS} [1]	0 –20 ms	From the USB host	N/A
$T_{CON_IDPSINK_DIS}$ [1]	0– 20 ms	From the USB host	N/A

1. This parameter is defined by the *USB Battery Charging Specification, Rev. 1.1*.

The following table provides an overview description of the charger detection sequence shown in the preceding figure.

Table 516. Overview of the charger detection sequence

Phase		Overview description	Full description
1	Initial conditions	Initial system conditions that need to be met before the detection sequence is initiated.	Initial system conditions
2	VBUS detection	System software detects contact of the VBUS signal with the system interrupt pin VBUS_detect.	VBUS contact detection
3	Data pin contact detection	The USBDCD module detects that the USB data pins D+ and D– have made contact with the USB port.	Data pin contact detection
4	Charging port detection	The USBDCD module detects if the port is an SDP or either type of charging port, that is CDP or DCP.	Charging port detection
5	Charger type detection	The USBDCD module detects the type of charging port, if applicable.	Charger type detection
6	Sequence timeout	The USBDCD module did not finish the detection sequence within the timeout interval. The sequence will continue until halted by software.	Charger detection sequence timeout

53.3.2.1 Initial system conditions

The USBDCD module is intended for use with USB device applications using a rechargeable battery. The module does not have support for interfacing with ACA or ACA-Dock equipment as defined in the *USB Battery Charging Specification, Revision 1.2*. It cannot be used with USB applications that are embedded host or OTG.

In addition, before the USBDCD module's charger detection sequence can be initiated, the system must be:

- Powered-up and in run mode. The USBDCD instantiation of this module for the High-Speed port does not directly depend on the VBUS voltage and can operate as long as the avdd33 supply is in a valid range.
- Recently plugged into a USB port.
- Drawing not more than 2.5 mA total system current from the USB bus, except as allowed by the *USB 2.0 Connect Timing Update ECN*.

Examples of allowable precursors to this set of initial conditions include:

- A powered-down device is subsequently powered-up upon being plugged into the USB bus.
- A device in a low-power mode subsequently enters run mode upon being plugged into the USB bus.

53.3.2.2 VBUS contact detection

Once the device is plugged into a USB port, the VBUS_detect system interrupt is triggered. System software must do the following to initialize the module and start the charger detection sequence:

1. Restore power if the module is powered-off.
2. Set CONTROL[SR] to initiate a software reset.
3. Configure the USBDCD module by programming the CLOCK register and the timing parameters as needed.
4. Set CONTROL[IE] to enable interrupts, or clear CONTROL[IE] if the software polling method is used.
5. Program CONTROL[BC12] based on which revision of the USB Battery Charging Specification is needed.
6. Set CONTROL[START] to start the charger detection sequence.

53.3.2.3 Data pin contact detection

The module must ensure that the data pins have made contact because the detection sequence depends upon the state of the USB D+ and D- signals. USB plugs and receptacles are designed such that when the plug is inserted into the receptacle, the power pins make contact before the data pins make contact. See the following figure.

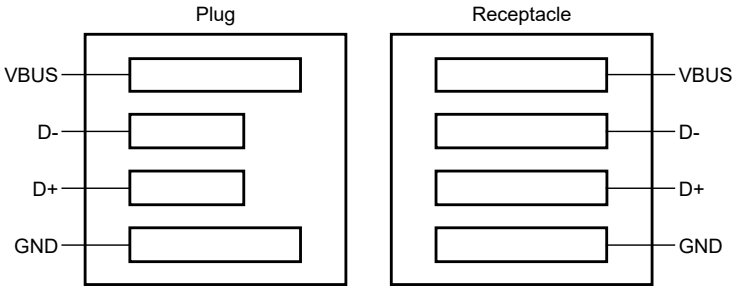


Figure 185. Relative pin positions in USB plugs and receptacles

As a result, when a portable USB device is attached to an upstream port, the portable USB device detects VBUS before the data pins have made contact. The time between power pins and data pins making contact depends on how fast the plug is inserted into the receptacle. Delays of several hundred milliseconds are possible.

53.3.2.3.1 Debouncing the data pin contact

When system software has initiated the charger detection sequence, as described in Initial system conditions, the USBDCD module turns on the I_{DP_SRC} current source and enables the R_{DM_DWN} pulldown resistor. If the data pins have not made contact, the D+ line remains high. After the data pins make contact, the D+ line goes low and debouncing begins.

After the D+ line goes low, the module continuously samples the D+ line over the duration of the T_{DCD_DBNC} debounce time interval. By default, T_{DCD_DBNC} is 10 ms, but it can be programmed in the TIMER0[TDCD_DBNC] field. See the description of TIMER0 for details.

When it has remained low for the entire interval, the debouncing is complete. However, if the D+ line returns high during the debounce interval, the module waits until the D+ line goes low again to restart the debouncing. This cycle repeats until either of the following happens:

- The data pin contact has been successfully debounced (see Success in detecting data pin contact (phase completion)).
- A timeout occurs (see Charger detection sequence timeout).

53.3.2.3.2 Success in detecting data pin contact (phase completion)

After successfully debouncing the D+ state, the module does the following:

- Updates the STATUS register to reflect phase completion (See Table 522 for field values.)
- Directly proceeds to the next step in the sequence: detection of a charging port (See Charging port detection.)

53.3.2.4 Charging port detection

After it detects that the data pins have made contact, the module waits for a fixed delay of 1 ms, and then attempts to detect whether it is plugged into a charging port. The module connects the following analog units to the USB D+ or D– lines during this phase:

- The voltage source V_{DP_SRC} connects to the D+ line
- The current sink I_{DM_SINK} connects to the D– line
- The voltage comparator connects to the USB D– line, comparing it to the voltage V_{DAT_REF}.

After a time of T_{VDPSRC_ON}, the module samples the D– line. The T_{VDPSRC_ON} parameter is programmable and defaults to 40 ms. After sampling the D– line, the module disconnects the voltage source, current sink, and comparator.

The next steps in the sequence depend on the voltage on the D– line as determined by the voltage comparator. See the following table.

Table 517. Sampling D– in the charging port detection phase

If the voltage on D- is...	Then...	See...
Below V_{DAT_REF}	The port is an SDP that does not support using charging currents above I_{CFG_MAX}.	SDP
Above V_{DAT_REF} but below V_{LGC}	The port is a charging port.	Charging port
Above V_{LGC}	This is an error condition.	Error in charging port detection

53.3.2.4.1 SDP

As part of the charger detection handshake with a standard USB host, the module does the following without waiting for the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) to elapse:

- Updates the STATUS register to reflect that an SDP is detected with SEQ_RES = 01b. See Table 522 for field values.
- Sets CONTROL[IF].

- Generates an interrupt if enabled in CONTROL[IE].

At this point, control is passed to system software via the interrupt. The rest of the sequence, which detects the type of charging port, is not applicable, so software must perform the following steps:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 511.

53.3.2.4.2 Charging port

As part of the charger detection handshake with any type of USB host, the module waits until the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) has elapsed before it does the following:

For *USB Battery Charging Specification, Rev. 1.2*:

- Enables V_{DM_SRC}.
- Updates the STATUS register to reflect that a charging port is detected with SEQ_RES = 10b. See Table 522 for field values.
- At this point the detection sequence progresses to Charger type detection without needing software interaction.

For *USB Battery Charging Specification, Rev. 1.1*:

- Updates the STATUS register to reflect that a charging port is detected with SEQ_RES = 10b. See Table 522 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].
- At this point, control is passed to system software via the interrupt. Software must:
  1. Read the STATUS register.
  2. Set CONTROL[IACK] to acknowledge the interrupt.
  3. Issue a command to the USB controller to pullup the USB D+ line.
  4. Wait for the module to complete the final phase of the sequence. See Charger type detection.

53.3.2.4.3 Error in charging port detection

For this error condition, the module does the following:

- Updates the STATUS register to reflect the error with SEQ_RES = 00b. See Table 522 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

Note that in this case the module does not wait for the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) to elapse.

At this point, control has been passed to system software via the interrupt. The rest of the sequence (detecting the type of charging port) is not applicable, so software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.

53.3.2.5 Charger type detection

For *USB Battery Charging Specification, Rev. 1.2*:

After the USBDCD module enables the V_{DM_SRC}, the module starts the T_{VDMSRC_ON} timer counting down the time interval programmed into the TIMER2[T_{VDMSRC_ON}] field.

Once the T_{VDMSRC_ON} timer is elapsed, the module samples the USB D+ line to determine the type of charger. See the following table.

Table 518. Sampling D+ in the charger type detection phase (BC1.2)

If the voltage on D+ is...	Then...	See...
High (D+>V_{DAT_REF})	The port is a DCP. [1]	DCP
Low (D+<V_{DAT_REF})	The port is a CDP. [2]	CDP

1. In a DCP, the D+ and D– lines are shorted together through a small resistor.
2. In a CDP, the D+ and D– lines are not shorted.

For *USB Battery Charging Specification, Rev. 1.1*:

After software enables the D+ pullup resistor, the module is notified automatically (via internal signaling) to start the CHECK_DM timer counting down the time interval programmed in the TIMER2[CHECK_DM] field.

After the CHECK_DM time is elapsed, the module samples the USB D– line to determine the type of charger. See the following table.

Table 519. Sampling D– in the charger type detection phase (BC1.1)

If the voltage on D– is...	Then...	See...
High	The port is a DCP. [1]	DCP
Low	The port is a CDP. [2]	CDP

53.3.2.5.1 DCP

For a DCP, the module does the following:

- Updates the STATUS register to reflect that a dedicated charging port is detected with SEQ_RES = 11b. See Table 522 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE] field.

The *USB Battery Charging Specification, Rev. 1.2* indicates that additionally if the detection sequence determines an attachment to a DCP, then the USB device should signal on the D+ pin by either enabling the D+ pullup resistor or enabling V_{DP_SRC}.

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Disable the USB controller to prevent transitions on the USB D+ or D– lines from causing spurious interrupt or wakeup events to the system.
3. Set CONTROL[IACK] to acknowledge the interrupt.
4. Set CONTROL[SR] to issue a software reset to the module.
5. Disable the module.

6. Enable signaling on the D+ pin. This can be done by enabling the Transceiver D+ pullup resistor through configuration of the register fields in the USB controller instance for this USB port. For USBHSDCD instantiation of USBDCD, this signaling can alternatively be done by enabling V_{DP_SRC} through setting the SIGNAL_OVERRIDE[PS] field to a value of 010b.
7. Communicate the appropriate charge rate to the external battery charger IC; see Table 511.

53.3.2.5.2 CDP

For a CDP, the module does the following:

- Updates the STATUS register to reflect that a CDP is detected with SEQ_RES = 10b. See Table 522 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control is passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 511.

53.3.2.6 Charger detection sequence timeout

The maximum time allowed to connect according to the *USB Battery Charging Specification* is one second. If the Unit Connection Timer reaches the one second limit and the sequence is still running as indicated by the STATUS[ACTIVE] field, the module does the following:

- Updates the STATUS register to reflect that a timeout error has occurred. See Table 522 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].
- The detection sequence continues until explicitly halted by software setting CONTROL[SR].
- The Unit Connection Timer continues counting. See the description of TIMER0.

At this point, control is passed to the system software via the interrupt, which has two options: ignore the interrupt and allow more time for the sequence to complete, or halt the sequence. To halt the sequence, software must:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.

This timeout function is also useful in case software does not realize that the USB device is unplugged from USB port during the charger detection sequence. If the interrupt occurs but the V_{BUS_DETECT} input is low, software can disable and reset the module.

System software might allow the sequence to run past the timeout interrupt under these conditions:

1. The USB Battery Charging Spec is amended to allow more time. In this case, software should poll TIMER0[$T_{UNITCON}$] periodically to track elapsed time after 1s; or
2. For debug purposes.

Note that the $T_{UNITCON}$ register field will stop incrementing when it reaches its maximum value so it will not roll over to zero and start counting up again.

53.3.2.7 Dead Battery Provision signaling

If the system needs to operate under the Dead Battery Provision of the *USB Battery Charging Specification, Revision 1.2*, the USBDCD module must be configured to signal V_{DP_SRC} on the USB_DP pin.

For information on Dead Battery Provision conditions and signaling, see the Dead or weak battery later in this chapter.

53.3.3 Clocking

The following table describes the clock sources for the USBDCD module. Refer to the chip's clocking chapter for clock setting, configuration, and gating information.

Note these clocks share same name with the USB PHY.

Table 520. USBDCD clocks

Clock name	Description
MODULE_CLK	Peripheral clock
MODULE_CLK_S	Peripheral access clock

53.3.4 Resets

There are two ways to reset various register contents in this module: hardware resets and a software reset.

53.3.4.1 Hardware resets

Hardware resets originate at the system or device level and propagate down to the individual module level. They include start up reset, low-voltage reset, and all other hardware reset sources.

Hardware resets cause the register contents to be restored to their default state as listed in the register descriptions.

53.3.4.2 Software reset

A software reset re-initializes the module's status information, but leaves configuration information unchanged. The software reset allows software to prepare the module without needing to reprogram the same configuration each time the USB device is plugged into a USB port.

Setting CONTROL[SR] initiates a software reset. The following table shows all register fields that are reset to their default values by a software reset.

Table 521. Software reset and register fields affected

Register	Fields affected	Fields not affected
CONTROL[1]	IF	IE, START
STATUS	All	None
CLOCK	None	All
TIMER*n*	TUNITCON	All other

1. CONTROL[SR] and CONTROL[IACK] are self-clearing.

A software reset also returns all internal logic, timers, and counters to their reset states. If the module is already active (STATUS[ACTIVE] = 1b), a software reset stops the sequence.

NOTE

Software must always initiate a software reset before starting the sequence to ensure the module is in a known state.

53.3.5 Interrupts

The USBDCD module has an interrupt to alert system software of certain events, which are listed in the following table. All events except the Phase Complete event for the Data Pin Detection phase can trigger an interrupt.

Table 522. Events triggering an interrupt by sequence phase

Sequence phase	Event	Event description	STATUS fields [1]	Phase description
Data Pin Detection	Phase Complete	The module has detected data pin contact. *No interrupt occurs*: CONTROL[IF] = 0b.	ERR = 0b SEQ_STAT = 01b SEQ_RES = 00b TO = 0b	VBUS contact detection
Charging Port Detection	Phase Complete	The module has completed the process of identifying if the USB port is a charging port or not. **NOTE:** An interrupt is triggered at completion of this phase for BC 1.1 operation whether an SDP or a Charging port is detected. For BC 1.2 operation, if an SDP is detected an interrupt is triggered but if a Charging port is detected the sequence goes directly to the Charger Type Detection phase.	ERR = 0b SEQ_STAT = 10b SEQ_RES = 01b or 10b TO = 0b	Charging port detection
	Error	The module cannot identify the type of port because the D– line is above the USB's VLGC threshold.	ERR = 1b SEQ_STAT = 10b SEQ_RES = 00b TO = 0b	Error in charging port detection
Charger Type Detection	Phase Complete	The module has completed the process of identifying the charger type detection. **NOTE:** The ERR flag always reads zero because no known error conditions are checked during this phase.	ERR = 0b SEQ_STAT = 11b SEQ_RES = 11b or 10b TO = 0b	Charger type detection
Sequence Timeout	Error	The timeout interval from the time the USB device attaches to a USB port until it connects has elapsed.	ERR = 1b SEQ_STAT = last value [2] SEQ_RES = last value [2] TO = 1b	Charger detection sequence timeout

1. See the description of the Status register for register information.
2. The SEQ_STAT and SEQ_RES fields retain the values held at the time of the timeout error.

53.3.5.1 Interrupt handling

Software reads which event caused the interrupt from the STATUS register during the interrupt service routine.

An interrupt is generated only if CONTROL[IE] is set. The CONTROL[IF] field is always set under interrupt conditions, even if CONTROL[IE] is cleared. In this case, software can poll CONTROL[IF] to determine if an interrupt condition is pending.

Writes to CONTROL[IF] are ignored. To reset CONTROL[IF], set CONTROL[IACK] to acknowledge the interrupt. Writing to CONTROL[IACK] when CONTROL[IF] is cleared has no effect.

53.4 External signals

This section describes the module signals. The following table shows a summary of module signals that interface with the pins of the device.

Table 523. Signal descriptions

Signal	Description	I/O
USB_DM	USB D– analog data signal. The analog block interfaces directly to the D– signal on the USB bus.	I/O
USB_DP	USB D+ analog data signal. The analog block interfaces directly to the D+ signal on the USB bus.	I/O
avdd33 [1]	3.3 V regulated analog supply	POWER
vss_bulk	Digital/Analog ground	POWER
dvdd	Core voltage digital supply	POWER

1. Voltage must be 3.3 V ± 10% for full functionality of the module. That is, the charger detection function does not work when this voltage is below 3.0 V, and the CONTROL[START] field should not be set.

NOTE

The transceiver module also interfaces to the USB_DM and USB_DP signals. Both modules and the USB host/hub use these signals as bidirectional, tristate signals.

Information about the signal integrity aspects of the lines including shielding, isolated return paths, input or output impedance, packaging, suggested external components, ESD, and other protections can be found in the USB 2.0 Specification and in Application information.

53.5 Initialization

This module is designed for minimal configuration while retaining significant programmability. To initialize USBDCD:

1. Initialize the CLOCK register to the actual system clock frequency, unless the default value already matches the system requirements.
2. Reset ANACTRL[DEV_PULLDOWN] to 0b when this module is in use for the proper operation of the USBHSDCD instantiation of USBDCD.

   For more information, see the "USB PHY Analog Control Register (ANACTRL)" section of the "Universal Serial Bus 2.0 Integrated PHY (USB-PHY)" chapter.
3. Do not modify other registers, as they default to the values that comply with the USB Battery Charging Specification.
4. Configure timing parameters to obtain flexibility as per the requirements of a specific system.

All module configuration must occur *before* initiating the charger detection sequence. Configuration changes made *after* setting CONTROL[START] result in undefined behavior.

53.6 Application information

This section provides application information.

53.6.1 External pullups

Any external pullups applied to the USB D+ or D- data lines must be capable of being disabled to prevent incorrect pullup values or incorrect operation of the USB subsystem.

53.6.2 Dead or weak battery

According to the *USB Battery Charging Specification, Revision 1.2*, a USB device with a dead or weak battery that is attached to an SDP can draw charging current from VBUS without connecting for several minutes, until the battery is charged to the point that the USB device can connect. The conditions for this operation are referred to as the Dead Battery Provision.

The *USB Battery Charging Specification, Revision 1.2* limits this charging condition to 100 mA for a duration of 45 minutes. The later *USB 2.0 Connect Timing Update ECN* modifies this charging condition to 500 mA for a duration of 2 minutes. Customers designing systems to take advantage of the Dead Battery Provision should study both specifications closely.

The USBDCD module is compatible with systems that do not check the strength of the battery. Therefore, this module assumes that the battery is good, so the USB device must immediately connect to the USB bus by pulling the USB_DP pin high after the USBDCD module has determined that the device is attached to an SDP or CDP. (By definition, a DCP does not support connection but still requires a signaling event, see DCP.)

The module is also compatible with systems that have other circuitry to check the strength of the battery. In these systems, if it is known that the battery is weak or dead, software can delay connecting to the USB while charging using the Dead Battery Provision. Once the battery is charged to the good battery threshold, software must then connect to the USB host by pulling the USB_DP pin high.

While using the Dead Battery Provision, the USB_DP pin must signal by enabling V_{DP_SRC}. On this product that signaling can be done most simply by setting the SIGNAL_OVERRIDE[PS] field to value 10b.

53.6.3 Handling unplug events

If the device is unplugged from the USB bus during the charger detection sequence, the contents of the STATUS register must be ignored and the USBDCD module must get a Software Reset, as described in Software reset.

53.7 USBDCD register descriptions

53.7.1 USBDCD memory map

USBDCD base address: 4041_4800h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (CONTROL)	32	RW	0001_0000h
4h	Clock (CLOCK)	32	RW	0000_00C1h
8h	Status (STATUS)	32	R	0000_0000h
Ch	Signal Override (SIGNAL_OVERRIDE)	32	RW	0000_0000h
10h	TIMER0 (TIMER0)	32	RW	0010_0000h
14h	TIMER1 (TIMER1)	32	RW	000A_0028h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
18h	TIMER2_BC11 (TIMER2_BC11)	32	RW	0028_0001h
18h	TIMER2_BC12 (TIMER2_BC12)	32	RW	0001_0028h

53.7.2 Control (CONTROL)

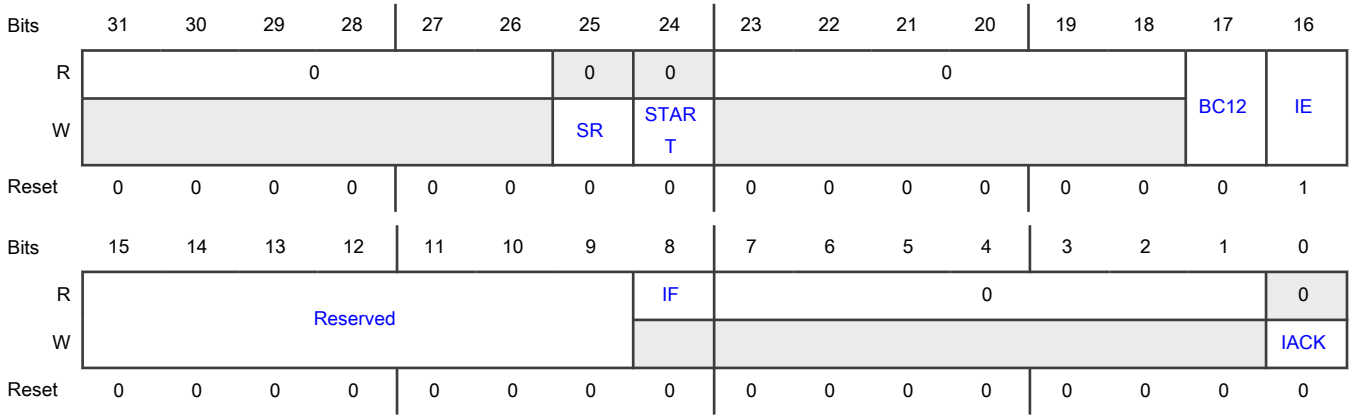
Offset

Register	Offset
CONTROL	0h

Function

Controls various global functions, such as software reset, initiating the charger detection sequence, BC12 compatibility, and interrupt enablement.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 SR	Software Reset Determines whether a software reset is performed. 0b - Do not perform a software reset. 1b - Perform a software reset.

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 START	Start Change Detection Sequence Determines whether the charger detection sequence is initiated. 0b - Do not start the sequence. Writes of this value have no effect. 1b - Initiate the charger detection sequence. If the sequence is already running, writes of this value have no effect.
23-18 —	Reserved
17 BC12	Battery Charging Revision 1.2 Compatibility BC1.2 compatibility. This field cannot be changed after start detection. 0b - Compatible with BC1.1 (default) 1b - Compatible with BC1.2
16 IE	Interrupt Enable Enables/disables interrupts to the system. 0b - Disable interrupts to the system. 1b - Enable interrupts to the system.
15-9 —	Reserved
8 IF	Interrupt Flag Determines whether an interrupt is pending. 0b - No interrupt is pending. 1b - An interrupt is pending.
7-1 —	Reserved
0 IACK	Interrupt Acknowledge Determines whether the interrupt is cleared. 0b - Do not clear the interrupt. 1b - Clear the IF field (interrupt flag).

53.7.3 Clock (CLOCK)

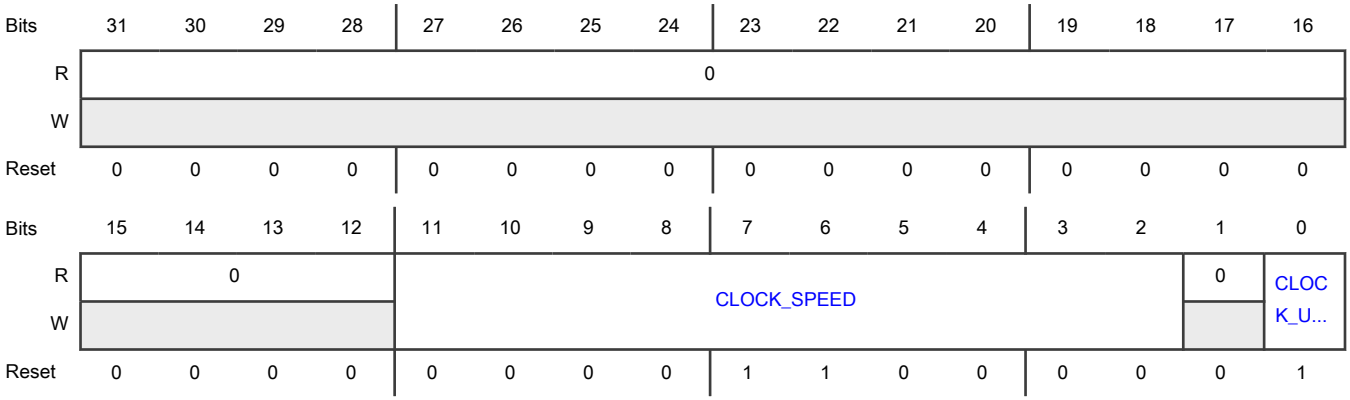
Offset

Register	Offset
CLOCK	4h

Function

Controls the clock speed.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-2 CLOCK_SPEED	Numerical Value of Clock Speed in Binary The unit of measure is programmed in CLOCK_UNIT. The valid range is from 1 to 1023 when the clock unit is MHz and 4 to 1023 when the clock unit is kHz. Examples with CLOCK_UNIT = 1b: • For 48 MHz: 00110000b (48) • For 24 MHz: 00011000b (24) Examples with CLOCK_UNIT = 0b: • For 100 kHz: 01100100b (100) • For 500 kHz: 000111110100b (500)
1 —	Reserved
0	Unit of Measurement Encoding for Clock Speed

Table continues on the next page...

Table continued from the previous page...

Field	Function
CLOCK_UNIT	Specifies the unit of measure for the clock speed. 0b - kHz Speed (between 4 kHz and 1023 kHz) 1b - MHz Speed (between 1 MHz and 1023 MHz)

53.7.4 Status (STATUS)

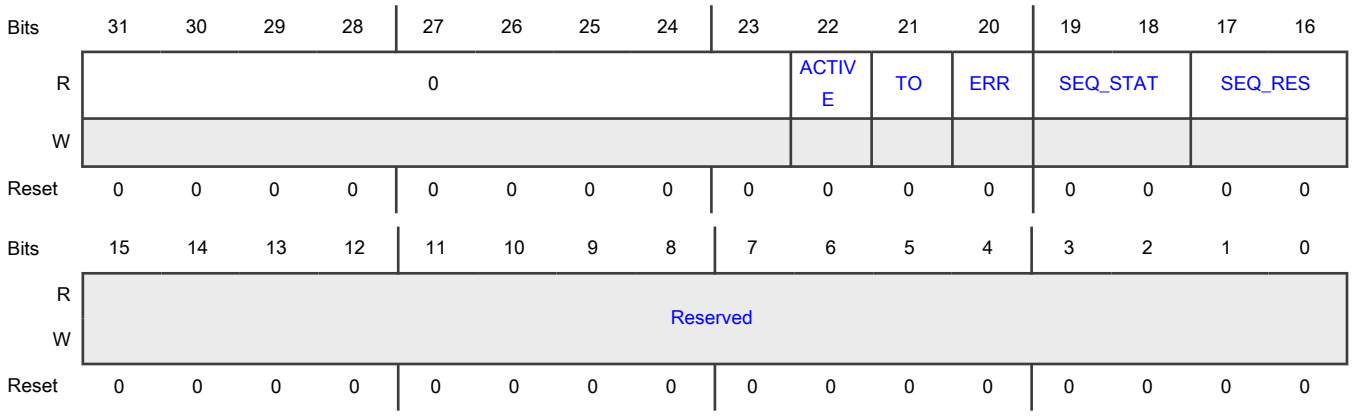
Offset

Register	Offset
STATUS	8h

Function

Stores the current state of the module for system software monitoring.

Diagram



Fields

Field	Function
31-23 —	Reserved
22 ACTIVE	Active Status Indicator Indicates whether the sequence is running. 0b - The sequence is not running. 1b - The sequence is running.
21	Timeout Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
TO	Indicates whether the detection sequence is passed the timeout threshold. 0b - The detection sequence is not running for over 1 s. 1b - It is over 1 s since the data pin contact was detected and debounced.
20 ERR	Error Flag Indicates whether there is an error in the detection sequence. 0b - No sequence errors. 1b - Error in the detection sequence. See STATUS[SEQ_STAT]to determine the phase in which the error occurred.
19-18 SEQ_STAT	Charger Detection Sequence Status Indicates the status of the charger detection sequence. 00b - The module is either not enabled, or the module is enabled but the data pins have not yet been detected. 01b - Data pin contact detection is complete. 10b - Charging port detection is complete. 11b - Charger type detection is complete.
17-16 SEQ_RES	Charger Detection Sequence Results Reports how the charger detection is attached. 00b - No results to report. 01b - Attached to an SDP. Must comply with USB 2.0 by drawing only 2.5 mA (max) until connected. 10b - Attached to a charging port. The exact meaning depends on the STATUS[SEQ_STAT] field (value 0: Attached to either a CDP or a DCP. The charger type detection has not completed. value 1: Attached to a CDP. The charger type detection has completed.) 11b - Attached to a DCP.
15-0 —	Reserved **NOTE** Fields do not always read as 0.

53.7.5 Signal Override (SIGNAL_OVERRIDE)

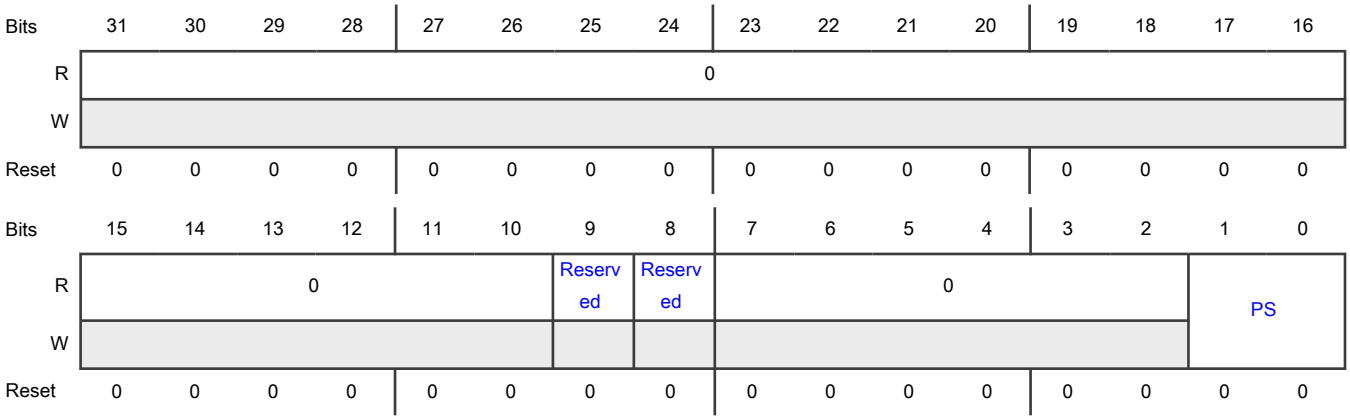
Offset

Register	Offset
SIGNAL_OVERRIDE	Ch

Function

Provides a way for the customer to enable signaling required by the USB BC Rev. 1.2 Specification after the battery charger detection sequences is completed.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 —	Reserved Reserved, not for customer use. The value of this field may change during module operation.
8 —	Reserved Reserved, not for customer use. The value of this field may change during module operation.
7-2 —	Reserved
1-0 PS	Phase Selection Enables specified voltage and current source circuits on the USB_DP and USB_DM pins. Use of the override value below enables the charger detection bias circuits without needing to set CONTROL[START]. Customers may set this field to 10b for required signaling if attached to a Dedicated Charging Port, or during operation under the Dead Battery Provision. 00b - No overrides. Field must remain at this value during normal USB data communication to prevent unexpected conditions on USB_DP and USB_DM pins. (Default) 01b - Reserved, not for customer use. 10b - Enables VDP_SRC voltage source for the USB_DP pin and IDM_SINK current source for the USB_DM pin. 11b - Reserved, not for customer use.

53.7.6 TIMER0 (TIMER0)

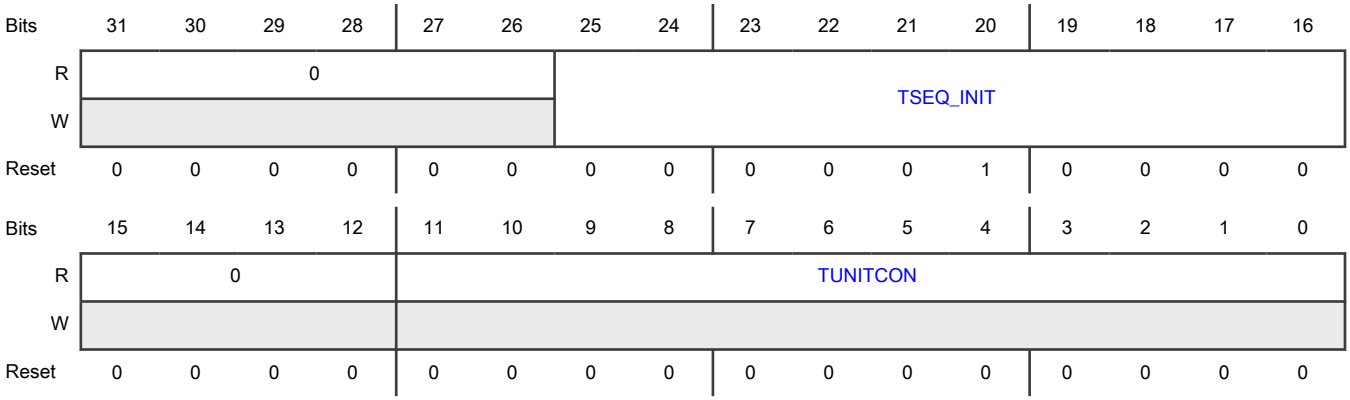
Offset

Register	Offset
TIMER0	10h

Function

Represents the system latency in ms using the TSEQ_INIT field.

Diagram



Fields

Field	Function
31-26 —	Reserved
25-16 TSEQ_INIT	Sequence Initiation Time TSEQ_INIT represents the system latency (in ms) measured from the time VBUS goes active to the time system software initiates the charger detection sequence in the USBDCD module. When software sets CONTROL[START], the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT. Valid values are 0-1023, but the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1 s or less. 00_0000_0000b-11_1111_1111b - 0 ms - 1023 ms
15-12 —	Reserved
11-0 TUNITCON	Unit Connection Timer Elapse (in ms) Displays the amount of elapsed time since the event of setting CONTROL[START] plus the value of TSEQ_INIT. The timer is automatically initialized with the value of TSEQ_INIT before starting to count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This timer enables compliance with the maximum time allowed to connect T_{UNIT_CON} under the USB Battery Charging Specification. If the timer reaches the one second limit, the module triggers an interrupt and sets the error flag STATUS[ERR]. The timer continues counting throughout the charger detection sequence, even when control has been passed to software. As long as the module is active, the timer continues to count until it reaches the maximum value of FFFh (4095 ms). The timer does not roll over to zero. A software reset clears the timer.

53.7.7 TIMER1 (TIMER1)

Offset

Register	Offset
TIMER1	14h

Function

Contains timing parameters.

NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						TDCD_DBNC									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TVDPSRC_ON									
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Fields

Field	Function
31-26 —	Reserved
25-16	Time Period to Debounce D+ Signal

Table continues on the next page...

Table continued from the previous page...

Field	Function
TDCD_DBNC	Sets the time period (ms) to debounce the D+ signal during the data pin contact detection phase. See Debouncing the data pin contact. Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 10 ms. 00_0000_0001b-11_1111_1111b - 1 ms - 1023 ms
15-10 —	Reserved
9-0 TVDPSRC_ON	Time Period Comparator Enabled This timing parameter is used after detection of the data pin. See Charging port detection. Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 40 ms. 00_0000_0001b-11_1111_1111b - 1 ms - 1023 ms

53.7.8 TIMER2_BC11 (TIMER2_BC11)

Offset

Register	Offset
TIMER2_BC11	18h

Function

TIMER2_BC11 contains timing parameters for *USB Battery Charging Specification, Rev 1.1.*

NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TVDPSRC_CON							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CHECK_DM			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-26 —	Reserved
25-16 TVDPSRC_CON	Time Period Before Enabling D+ Pullup Sets the time period (ms) that the module waits after charging port detection before system software must enable the D+ pullup to connect to the USB host. Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 40 ms. 00_0000_0001b-11_1111_1111b - 1 ms - 1023 ms
15-4 —	Reserved
3-0 CHECK_DM	Time Before Check of D- Line Sets the amount of time (in ms) that the module waits after the device connects to the USB bus until checking the state of the D- line to determine the type of charging port. See Charger type detection. Valid values are 1-15 ms. 0001b-1111b - 1 ms - 15 ms

53.7.9 TIMER2_BC12 (TIMER2_BC12)

Offset

Register	Offset
TIMER2_BC12	18h

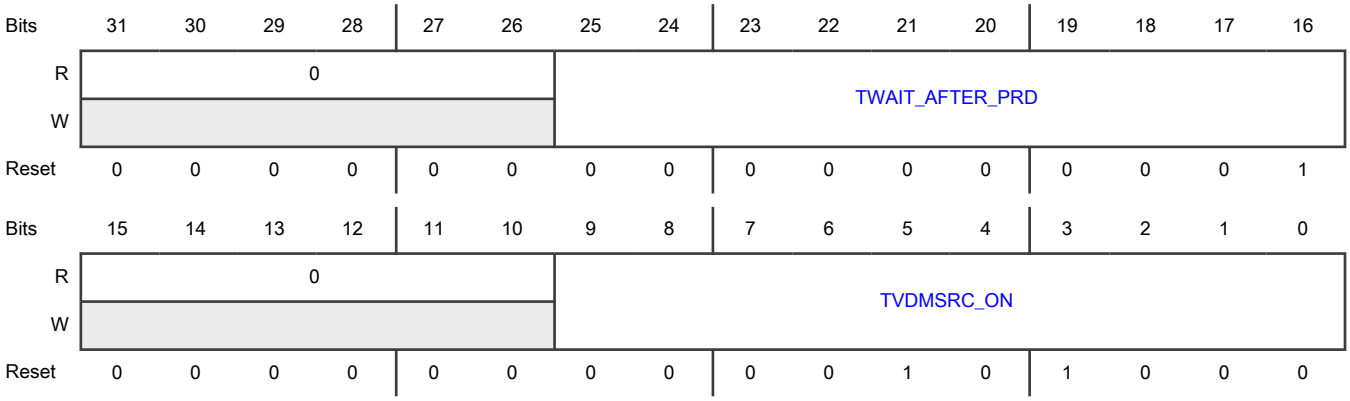
Function

Contains timing parameters for *USB Battery Charging Specification, Rev 1.2.*

NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

Diagram



Fields

Field	Function
31-26 —	Reserved
25-16 TWAIT_AFTER_PRD	TWAIT_AFTER_PRD Sets the amount of time (in ms) that the module waits after primary detection before start to secondary detection. Valid values are 1-1023ms. 00_0000_0001b-11_1111_1111b - 1 ms - 1023 ms
15-10 —	Reserved
9-0 TVDMSRC_ON	TVDMSRC_ON Sets the amount of time (in ms) that the module enables the V_{DM_SRC}. Valid values are 0-40ms. 00_0000_0000b-00_0010_1000b - 0 ms - 40 ms

Chapter 54
Mobile Industry Processor Interface and Display Serial Interface Controller (MIPI_DSI_HOST)

54.1 Chip-specific MIPI_DSI_HOST information

Table 524. Reference links to related information

Topic	Related module	Reference
Full description	MIPI_DSI_HOST	MIPI_DSI_HOST
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Reset controller	RSTCTL	RSTCTL
System Controller	SYSCON	Control the DSI shutdown and color mode and indicate DSI transmit status by SYSCON4.MIPI_DSI_CTRL register in SYSCON

54.1.1 Module instances

This device has one instance of the MIPI_DSI_HOST module.

Table 525. MIPI_DSI_HOST instance

	MIPI_DSI_HOST
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

54.1.2 Reset and initialization

NOTE

Initialize the LCDIF module before the MIPI_DSI_HOST module.

Chip-specific Reset and Initialization procedure for the MIPI_DSI_HOST:

- Deassert all resets through RSTCTL3_PRSTCTL0[MIPIDSI].
- Setup MIPI_DSI_HOST related clocks through chip-specific registers provided for this purpose.
- Setup MIPI DSI application by programming all DSI Host Controller and DSI DPHY required parameters (refer to MIPI_DSI_HOST Memory Map/Register for details).

MIPI_DSI_HOST is ready for use.

54.2 Overview

MIPI_DSI_HOST controller is a flexible, high-performance, and easy-to-use digital core that implements all protocol functions defined in the MIPI_DSI_HOST Specification. The MIPI_DSI_HOST controller provides an interface that allows communication with MIPI_DSI_HOST-compliant peripherals.

The MIPI_DSI_HOST D-PHY is a high frequency, low power, low-cost, source-synchronous, physical layer supporting the MIPI Alliance standard for D-PHY.

NOTE

MIPI_DSI_HOST in this chip does not have an embedded PLL, hence disregard any PLL related contents in this chapter.

54.2.1 Features

Key features of the MIPI_DSI_HOST Controller Core include:

- Implements all three DSI Layers (Pixel to Byte packing, Low Level Protocol, Lane Management)
- Support for Command and Video Modes
- Host Version
- Scalable data lane support, 1 to 2 Data Lanes
  - Optional bidirectional support on lane 0
- Supports High Speed and Low Power operation
- Support for all DSI data types and formats
- Virtual Channel support
- Supports ULPS mode
- Full Low-Level Protocol Error and Contention detection and reporting
- Supports continuous and non-continuous Clock Lane operation
- Supports multiple packets per transmission
- Support for all three Video Mode packet sequences
  - Non-Burst Mode with Sync Pulses
  - Non-Burst Mode with Sync Events
  - Burst mode
- Support for bus turnaround signaling
- Flexible packet based user interface
  - APB interface option (status and control)
  - Display Pixel Interface Core (DPI-2) option
  - Display Bus Interface Core (DBI-2) option
- Supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs
- MIPI Alliance Specification for Display Serial Interface Version 1.1 compliant

The following are the key features of the MIPI_DSI_HOST D-PHY:

- 1 Clock lane
- 2 data lanes
- Supports MIPI Standard for D-PHY

- Supports both high speed and low power modes
- High Speed Serializers and Deserializers

54.3 DSI Host Controller Core

The figure below illustrates the DSI Host Controller Core structure. The DSI Host Controller Core operates on the host (transmit) side of a DSI link.

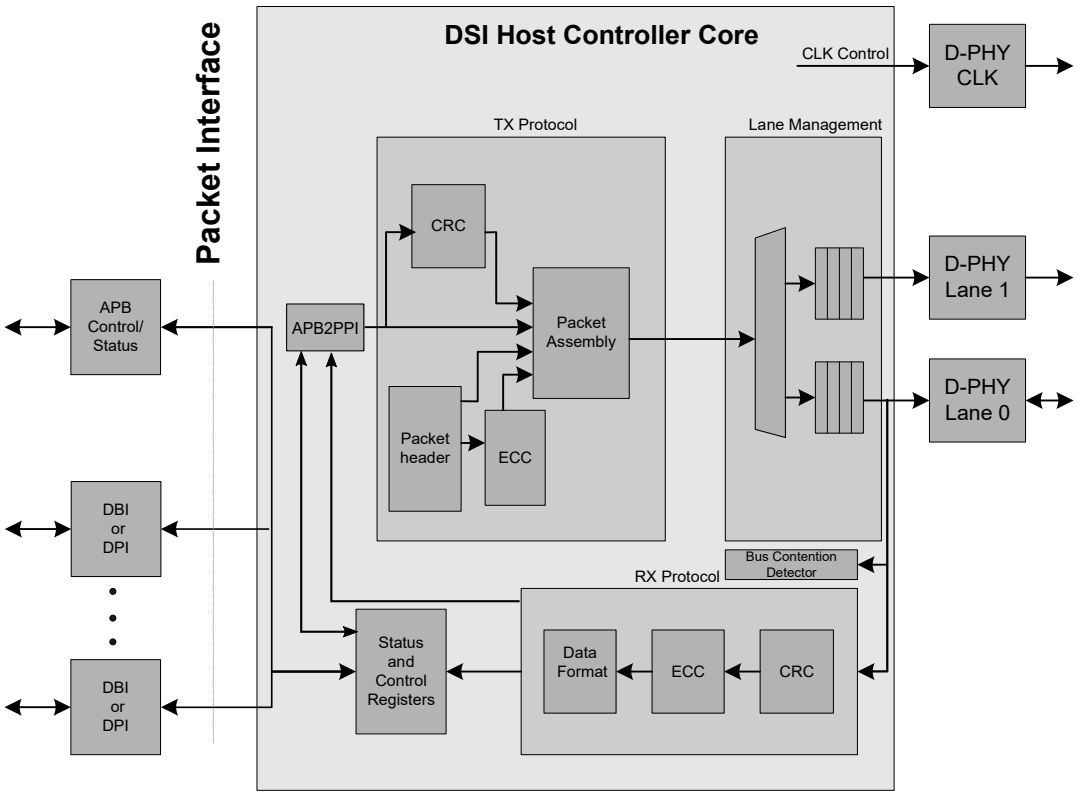


Figure 186. DSI Host Controller Core Block Diagram

The DSI Host Controller Core implements all three layers defined by the DSI Specification: Pixel to Byte Packing in the Application layer, Low Level Protocol, and Lane Management. The DSI Host Controller Core sends and receives DSI commands via the Packet Interface. The Packet Interface can be connected to a DBI/DPI translator or to an APB to PPI gasket.

The D-PHY interface of the DSI Host Controller Core supports up to four PHY Protocol Interface (PPI) compatible MIPI D-PHYs.

The Packet Interface is an easy-to-use data interface that accepts commands and data, and sends it over the DSI link. It supports 1 to 4 virtual channels, and the use of 1-2 D-PHY lanes. The DSI Host Controller Core takes care of all packet formatting details and transmission over the MIPI bus.

The DBI/DPI Translator connects to the DSI Host Controller Core via the Packet Interface. DBI/DPI controllers may connect directly to the DBI/DPI Translator to send commands across the DSI link.

Sending commands and receiving information through the DSI link is also possible through the APB to PPI gasket. See MIPI_DSI_HOST APB PKT IF Memory Map/Register section for details about the DSI registers which allow the APB to PPI setup and communication.

54.3.1 D-PHY Interface

The DSI Host Controller Core D-PHY Interface connects directly to MIPI PPI compliant D-PHYs. The DSI Host Controller Core uses this interface to transmit data, send Escape sequences, receive and transmit triggers, and detect and report D-PHY error conditions.

54.3.1.1 High-Speed Transmit Interface

The High-Speed Transmit interface is used to transmit High-Speed data across the MIPI interface. All signals are synchronous to the PPI TX byte clock, TxByteClkHS, and are PPI compliant.

54.3.1.2 Escape Mode Transmit Interface

The DSI Host Controller Core uses the Escape Mode Transmit Interface to generate Escape sequences on the MIPI Interface via the D-PHY. Escape sequences are used to change the DPHY transmit mode, go into low-power mode, or initiate a bus turnaround. The Controller generates Escape sequence requests when necessary to implement the DSI protocol. These signals are synchronous to the clk_esc.

54.3.2 Packet Interface

The DSI Host Controller Core Packet Interface consists of Transmit, Receive, and Control and Status sections. Through these interfaces, the user application can take complete control of the DSI interface, sending all video timing, sending DSI commands, receiving DSI reads, monitoring the status of the interface and responding to error reporting.

User application may have access to DSI Host Controller Core Packet Interface through DSI_HOST_APB_PKT_IF registers, DSI_HOST_TX_PAYLOAD and DSI_HOST_PKT_CONTROL. For details refer to the section MIPI DSI Host register descriptions

54.3.2.1 Transmit Ports

The Transmit Packet Interface is the mechanism with which the user creates packets to send over the MIPI Interface.

For Long Packets, the user provides the Virtual Channel (VC) number, Data Type (DT), and Word Count (WC) to the controller. The Controller then creates a packet header and pulls the packet data from the Packet Interface and out to the D-PHY to transmit.

For Short Packets, the user provides the Virtual Channel (VC) number, Data Type (DT), and required parameters (if any) to the controller. The Controller then creates the short packet and sends it to the D-PHY to transmit. This interface enables the user application to transmit and receive any type of DSI packet.

The Packet Interface Transmit Interface ports are listed in the table below. Signals are synchronous to the Tx Byte clock.

NOTE

Only a subset of the Packet Interface Signals are accessible by user application through DSI_HOST_APB_PKT_IFx (DSI_HOST_TX_PAYLOAD and DSI_HOST_PKT_CONTROL) registers. For reset_byte_n and reset_esc_n signals, refer to chip-specific MIPI_DSI_HOST information. For tx_payload, tx_cmd_data_type, tx_cmd_vc, and tx_cmd_byte_count signals, refer to DSI_HOST_TX_PAYLOAD register.

Table 526. DSI Host Controller Core Transmit Packet Interface

PORT	TYPE	DESCRIPTION
clk_byte	Input	Byte clock input. The D-PHY PPI interface, the tx_cmd and tx_payload interfaces are synchronous to clk_byte. This clock can be independent and unrelated to clk_esc.
clk_esc	Input	The low-speed D-PHY Escape Mode clock, which becomes TxClkEsc on the D-PHY interface. This clock can be independent and unrelated to clk_byte.
reset_byte_n	Input	Asynchronous reset, active low. This reset applies to all logic in the clk_byte clock domain.
reset_esc_n	Input	Asynchronous reset, active low. This reset applies to all logic in the clk_esc clock domain.
tx_payload[31:0]	Input	Packet data input.

Table continues on the next page...

Table 526. DSI Host Controller Core Transmit Packet Interface (continued)

PORT	TYPE	DESCRIPTION
tx_payload_en	Output	Packet data read enable. This active high signal indicates that the controller requires a valid packet during the next clk_byte period.
tx_payload_en_last	Output	Last packet read enable, active high signals last cycle of tx_payload_en.
tx_cmd_data_type[5:0]	Input	Transmit packet DSI data type. It is written into the command buffer when tx_cmd_ack is asserted high.
tx_cmd_vc[1:0]	Input	Transmit packet command virtual channel. It is written into the command buffer when tx_cmd_ack is asserted high.
tx_cmd_byte_count[15:0]	Input	Transmit packet payload byte count. It is written into the command buffer when tx_cmd_ack is asserted high. For DSI Long packet types, tx_cmd_byte_count defines the number of bytes of packet data to pull from the tx_payload port. For DSI Short packets, the format of tx_cmd_byte_count contains any optional parameters. If the SDI Short packet type does not have any parameters, it is recommended to set tx_cmd_byte_count to all 0s.
tx_cmd_req	Input	Transmit packet command request. This active high signal informs the controller that the packet command is valid. The packet command consists of the ports tx_cmd_data_type, tx_cmd_vc, and tx_cmd_byte_count. The controller will assert tx_cmd_ack when it accepts the command, after which, the user should either update port values for the next transmit packet command or deassert tx_cmd_req.
tx_cmd_ack	Output	Transmit packet command request acknowledge. This active high signal indicates that the controller has accepted the TX packet request and the user logic should either submit a new request or deassert tx_cmd_req on the next rising edge of clk_byte.

54.3.2.2 Receive Ports

The Receive Packet Interface returns data from the Peripheral to the user. The user is provided the Virtual Channel (VC) number, Data Type (DT), Word Count (WC), and Rx Payload Data.

This interface enables the user application to receive any type of DSI packet. These signals are synchronous to the clk_byte clock. The Packet Interface Receive Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals are accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- rx_payload
- rx_cmd_vc
- rx_cmd_data_type
- rx_cmd_byte_count
- ecc_one_bit_error
- ecc_two_bit_error
- ecc_one_bit_error_pos[4:0]
- crc_err

See DSI_HOST_APB_PKT_IFx and DSI_HOST device sections in MIPI DSI Host register descriptions for details.

Table 527. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
rx_payload[31:0]	Output	Received packet data output. The Host Receive Packet Interface presents 4 bytes at a time. Bytes are valid in this interface according to the rx_cmd_byte_count signal, beginning with the lowest byte.
rx_payload_valid	Output	Packet data valid. This active high signal indicates that the controller is presenting valid packet during the next clk_byte period.
rx_payload_valid_last	Output	This data is the last of the packet, active high signals last cycle of rx_payload_valid.
rx_cmd_valid	Output	Packet header data is valid on the packet header ports below when this signal is asserted.
rx_cmd_vc[1:0]	Output	Packet virtual channel number, valid when rx_cmd_valid is asserted.
rx_cmd_data_type[5:0]	Output	Packet data type, valid when rx_cmd_valid is asserted. See the MIPI_DSI_HOST-2 specification for a definition of possible values.
rx_cmd_byte_count[15:0]	Output	Packet Word Count (byte count). Contains the number of bytes of data in the received packet. Valid when rx_cmd_valid is asserted.
ecc_one_bit_error	Output	Single bit error in the packet header was detected and corrected. Active high. Valid when rx_cmd_valid is high.
ecc_two_bit_error	Output	Two packet header bit errors were detected and not corrected, active high. Valid when rx_cmd_valid is high.
ecc_one_bit_error_pos[4: 0]	Output	Position of the corrected single bit error in the packet header. Valid when ecc_one_bit_error is high.
crc_err	Output	Asserts high when the CRC calculated on the received data does not reach the end of the packet.

54.3.3 Status Interface Ports

The Status Interface Core provides for reading the Status of the DSI Host Controller Core via a simple interface. All signals in the Status Interface are synchronous to the clk_byte clock. The Status Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals are accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- status_out
- hs_tx_timeout
- lp_rx_timeout

Refer to DSI_HOST_APB_PKT_IFx and DSI_HOST device sections in MIPI DSI Host register descriptions for details.

Table 528. DSI Host Controller Status Interface Port

PORT	TYPE	DESCRIPTION
status_rd	Input	Initiates a read of the status register. Asserting this input high will result in the contents of the status register being presented on status_out[15:0] along with status_out_valid asserting high to indicate that status_out[15:0] contains a valid value.
status_addr[3:0]	Input	When status_rd is asserted, indicates which status register is to be returned. Address 0x0 is a valid address while 0x1-0xF are reserved.
status_out[31:0]	Output	Contains the contents of the addressed status register.
status_out_valid	Output	When asserted high, indicates that the data on status_out is valid.
tx_active	Output	tx_active asserts high when the Host Controller is actively transmitting data or when it has accepted a request from the user but has not yet started transmitting.
hs_tx_timeout	Output	Asserts high for one clk_byte period to indicate that a High Speed transmit has timed out.
lp_rx_timeout	Output	Asserts high for one clk_byte period to indicate that a Low Power RX occurred.

54.3.4 Control and Status Ports

The Control and Status Packet Interface allows the user to control and receive the status of the underlying link between the DSI Host and DSI Peripheral. It works on the same clk_byte clock as the Host Transmit Packet Interface. The Packet Interface Control and Status Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals may be accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- dphy_direction
- tx_hs_mode
- dphy_turnaround

Please refer to DSI_HOST_APB_PKT_IFx and DSI_HOST device sections in MIPI DSI Host register descriptions for details.

Table 529. DSI Host Controller Core Control And Status Packet Interface

PORT	TYPE	DESCRIPTION
tx_hs_mode	Input	Switches the DPHY into High Speed Data Transfer mode or Low Power Data Transfer mode. 1'b1 = request HS mode 1'b0 = request LP mode. The Packet interface will not acknowledge packet commands or data while switching modes.
dphy_turnaround	Input	Requests bus turnaround. 1'b1 = Request reverse direction LP mode, from Host TX to Host RX. 1'b0 = No effect. This signal is ignored if the bus is already in Reverse (Host RX) direction.
dphy_direction	Output	Reports the current bus direction. 1'b1 = Bus is in Reverse direction (Host RX). 1'b0 = Bus is in Forward direction (Host TX).

54.3.5 Interface Cores

There are several interface cores for the DSI Host Controller Core. These interface cores provide industry standard interfaces to the DSI Host Controller Core to simplify interfacing and integration the DSI Host Controller Core to your application. When selected, these interface cores provide additional ports to the DSI Host Controller core.

54.3.5.1 DPI-2 Interface Host Bridge Core

The DPI-2 Interface Host Bridge Core provides a DPI-2 compliant interface to the DSI Host controller. The DPI-2 Interface Host Bridge Core handles converting the DPI-2 interface into a packet format that is compatible with the DSI Host Controller Core's Packet Interface, and that adheres to the DSI specification. The DPI-2 Interface Core handles all pixel-to-byte packing for all DPI data types.

The DSI Host Controller DPI-2 Interface Core provides the following features:

- Support for Type 2,3, and 4 displays
- Support for 16-, 18- and 24- bit Pixel data and all alignment configurations
- Support for 16, 18, 24, and 18 bit loosely packed DSI data types
- Supports DSI video modes
  - Non-Burst Mode with Sync Pulses
  - Non-Burst Mode with Sync Events
  - Burst Mode
- Supports normal or inverted HSYNC and VSYNC signals
- Handles clock domain crossing from DPI Pixel clock to the Host controller TX Byte clock
- Interfaces directly to the Host Controller's DSI Packet Interface
- Comes already integrated with the DSI Host Controller

A block diagram of the DPI-2 Interface Host Bridge Core is shown below. The DPI-2 Host Bridge Core accepts a MIPI compliant DPI-2 set of signals (vsync, hsync, pixel data), creates DSI packets for the video timing events and video data and transmits those packets via the DSI Host Controller's Packet Interface.

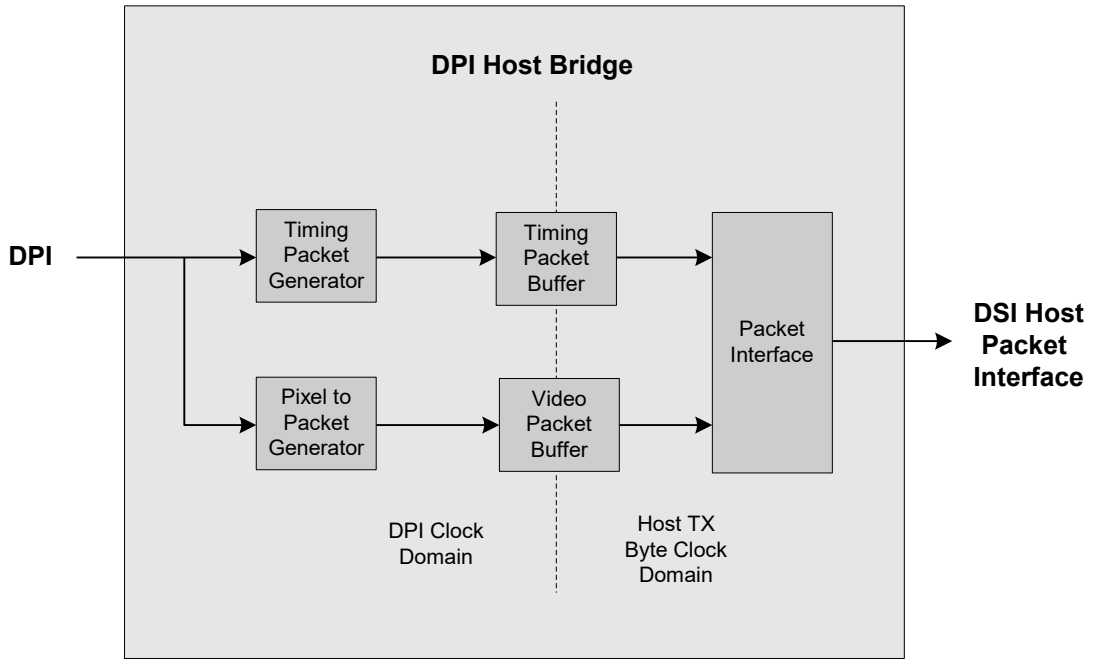


Figure 187. DPI-2 Host Bridge Core Block Diagram

54.3.5.1.1 DPI-2 Interface Ports

The DPI-2 Interface module provides a MIPI compliant DPI-2 interface to the DSI Host Controller Core. In addition to the standard DPI-2 interface signals (prefixed with a dpi_) there are several configuration and status ports that allow the user to adjust the behavior of the DPI-2 Interface module to better suit a particular application.

NOTE

The DPI configuration signals herein described are accessible to User Application through chip-specific registers (please refer to chip-specific MIPI-DSI information section for details) and DSI_HOST_DPI_INTFC device registers (please refer to MIPI_DSI_HOST Memory Map/Register section for details).

Table 530. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
dpi_sd	Input	Shut Down – Control to shutdown display (type 4 only) 1'b1= Send shutdown command. 1'b0= No effect
dpi_cm	Input	Color Mode control. 1'b0== Normal Mode 1'b1== Low-color Mode
cfg_dpi_pixel_payload_size [15:0]	Input	Maximum number of pixels that should send as one DSI packet. Recommended to be evenly divisible by the line size (in pixels).
cfg_dpi_pixel_fifo_send_ level[15:0]	Input	In order to optimize DSI utility, the DPI bridge buffers a certain number of DPI pixels before initiating a DSI packet. This configuration port controls the level at which the DPI Host bridge begins sending pixels.

Table continues on the next page...

Table 530. DSI Host Controller Core Receive Packet interface (continued)

PORT	TYPE	DESCRIPTION
cfg_dpi_interface_color_coding[2:0]	Input	Sets the distribution of RGB bits within the 24-bit d bus, as specified by the DPI specification. 0= 16-bit Configuration 1 1= 16-bit Configuration 2 2= 16-bit Configuration 3 3= 18-bit Configuration 1 4= 18-bit Configuration 2 5= 24-bit
cfg_dpi_pixel_format[1:0]	Input	Sets the DSI packet type of the pixels. 0= 16-bit 1= 18-bit 2= 18-bit loosely packed, 3= 24-bit
dpi_host_underrun_err	Output	During DSI Host transmission of DPI data insufficient DPI data was received. This may indicate that DPI_CLK is too slow, or that the cfg_dpi_* parameters are incorrectly set.
cfg_dpi_vsync_polarity	Input	Sets polarity of dpi_vsync input, 0 – active low, 1 active high
cfg_dpi_hsync_polarity	Input	Sets Polarity of dpi_hsync input, 0 – active low, 1 – active high
cfg_host_dpi_video_mode[1:0]	Input	Select DSI video mode that the host DPI module should generate packets for. 2'b00 – Non-Burst mode with Sync Pulses 2'b01 – Non-Burst mode with Sync Events 2'b10 – Burst mode 2'b11 – Reserved, not valid
cfg_host_dpi_hfp[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal front porch blanking packet.
cfg_host_dpi_hbp[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal back porch blanking packet.
cfg_host_dpi_hsa[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal sync active packet.
cfg_enable_mult_packets	Input	Enable Multiple packets per video line. When enabled, cfg_dpi_pixel_payload_size must be set to exactly half the size of the video line. 0 – Video Line is sent in a single packet 1 – Video Line is sent in two packets
cfg_bllp_mode	Input	Optimize bllp periods to Low Power mode when possible 0 – blanking packets are sent during BLLP periods 1 – LP mode is used for BLLP periods

54.3.5.1.2 DPI-2 Video Timing Recreation over DSI

The DSI Host DPI-2 module is designed to preserve the receive video timing by properly spacing the DSI packets that carry the DPI video information and by inserting blanking packets into the packet stream. In order to achieve this goal, the DSI Host controller needs to have the cfg_host_dpi_hfp, cfg_host_dpi_hbp, and cfg_host_dpi_hsa ports properly set. The values of these ports reflect the size of the hfp, hbp, and has in terms of DSI packet bytes, values such that the number of DSI bytes when transmitted takes approximately the same amount of absolute time as the video event took on the DPI interface.

To relate pixel sizes on the DPI interface to DSI packet bytes, use the following relationship:

```
Time of DPI event = time to transmit x number of bytes on the DSI interface
```

```
dpi_event_size * dpi_pclk_period = number_of_dsi_bytes * 8 * dsi_hs_bit_period / cfg_num_lanes
```

Solving for number_of_dsi_bytes:

```
number_of_dsi_bytes = dpi_event_size * dpi_pclk_period * cfg_num_lanes / (8 * dsi_hs_bit_period)
```

number_of_dsi_bytes	value to set the cfg_host_dpi_hfp (hbp, or hsa) port to
dpi_event_size	size of the video event in pixels (hfp, hbp, hsa)
dpi_pclk_period	period of dpi_pclk
dsi_hs_bit_period	bit period of the data lanes in High Speed mode
cfg_num_lanes	host controller setting for number of active lanes (0 = 1 lane, 1 = 2 lanes, 2 = 3 lanes, 3 = 4 lanes)

54.3.5.2 DBI-2 Interface Core

The DBI-2 Interface Core provides a DBI-2 compliant interface that attaches to a Packet Interface Port on the DSI Host controller. The DBI-2 Interface Core handles converting the DBI-2 interface into a packet format that is compatible with the DSI Host Controller Core's Packet Interface and adheres to the DSI specification. The DBI-2 Interface Core handles all pixel-to-byte packing for all DBI data types.

The DSI Host Controller DBI-2 Interface Core provides the following features:

- Support for Type 1, 2, and 3 displays
- Support for DBI-2 Type B and C Interfaces
- Support for 8-, 9- and 16- bit data (Type B)
- Interfaces directly to the Host Controller's Packet Interface
- Comes already integrated with the DSI Host Controller
- Source Code License available (Verilog)
- Provided with expert technical support
- Customization and integration services available

54.3.5.2.1 DBI-2 Host Bridge Module Interface Ports

The DBI-2 Interface module provides a MIPI compliant DBI-2 interface to the DSI Host Controller Core. In addition to the standard DBI-2 interface signals (prefixed with a dbi_) there are several configuration and status ports that allow the user to tune adjust the behavior of the DBI-2 Interface module to better suit a particular application.

Table 531. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
dbi_csx	Input	DBI Chip Select, active low.
dbi_wrx	Input	DBI Write enable. Write data/command is registered at the rising edge of dbi_wrx.
dbi_rdx	Input	DBI Read enable. Read data is read at the rising edge of dbi_rdx.
dbi_d_write[15:0]	Input	8-,9- or 16-bit Input Data for Writes
dbi_d_read[15:0]	Output	8-,9- or 16-bit Output Data for Reads
dbi_dcx	Input	Data/Command indicator. 0 – Command cycle 1 – Data cycle
dbi_resx	Input	Active low DBI Display Module reset
dbi_te	Output	Tearing Effect (optional support)

Table continues on the next page...

Table 531. DSI Host Controller Core Receive Packet interface (continued)

PORT	TYPE	DESCRIPTION
dbi_host_underrun_err	Output	During DSI Host transmission of DBI data insufficient DBI data was received. This may indicate that the rate of data received on dbi_d_in was insufficient to keep up with the outgoing data rate on the MIPI interface, or that the cfg_dbi_* parameters are incorrectly set.
cfg_dbi_pixel_payload_size[15:0]	Input	Maximum number of pixels that should send as one DSI packet. Recommended to be evenly divisible by the line size (in pixels).
cfg_dbi_pixel_fifo_send_level[15:0]	Input	In order to optimize DSI utility, the DBI bridge buffers a certain number of DBI pixels before initiating a DSI packet. This configuration port controls the level at which the DBI Host bridge begins sending pixels.

54.3.5.2.2 DBI-2 Host Bridge Module Write and Read Waveforms

The figure below shows a DBI-2 Write transaction. The command portion of the write is registered with the rising edge of dbi_wrx when dbi_dcx and dbi_csx are low. The write data is registered with the rising edge of dbi_wrx when dbi_dcx is high and dbi_csx is low.

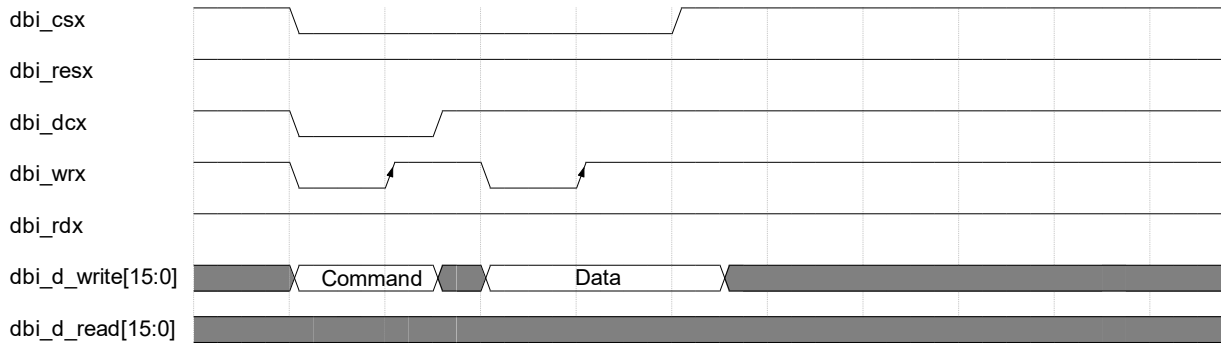


Figure 188. DBI-2 Write Waveform

The figure below shows a DBI-2 Read Transaction. The command portion of the read is registered with the rising edge of dbi_wrx when dbi_dcx and dbi_csx are low. The first cycle of read data is invalid with the first rising edge of dbi_rdx while the second cycle of read data is valid with the rising edge of dbi_rdx.

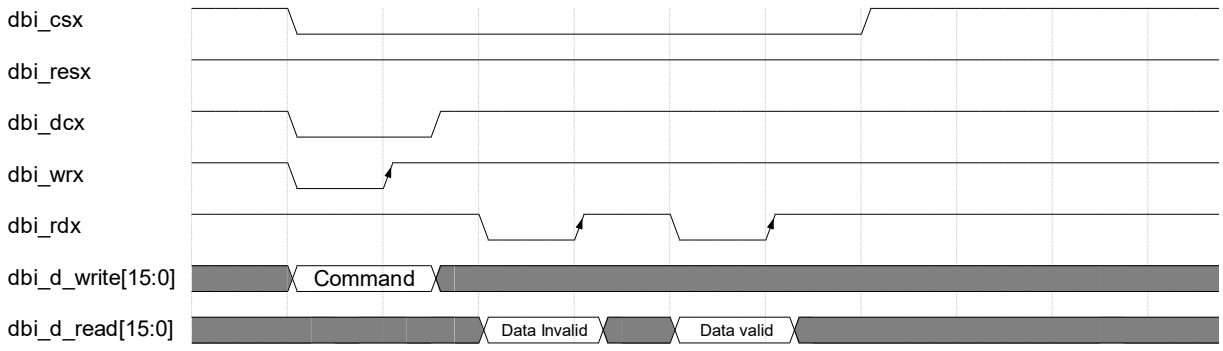


Figure 189. DBI-2 Read Waveform

54.3.6 RAM Usage

The DSI Host Controller Core requires on-chip RAM arrays. These RAM arrays and their corresponding sizes are listed in the table below.

All RAM arrays used in this core are configured as two-port, having the following attributes:

- Two address buses, one for the write port and one for read port
- Two data buses, one for write data on the write port and one for read data on the read port
- Two clock inputs, one for the write port and one for the read port. (Note: In some cases, the core logic will connect the same clock to both the read and write clock inputs of the RAM.)

Ram Purpose	Size (Depth x Width)
DPI Core Pixel Data FIFO	2048 x 32
APB to Packet Interface TX FIFO	64 x 32
APB to Packet Interface RX FIFO	64 x 32

54.3.7 DSI Host Clocking

The DSI Host Controller requires the following clocks:

- Reference clock (ref_clk) - used by the DPHY (DPHY PLL generates byte clock)
- Byte clock (clk_byte) - generated by the DPHY, used for packet generation and transfer to the DPHY
- TX escape mode clock (clk_tx_esc) - for internal LPDT state machines and low-power transmissions;
- Pixel clock (dpi_pclk)
- Serial clock (dbi_c_scl)

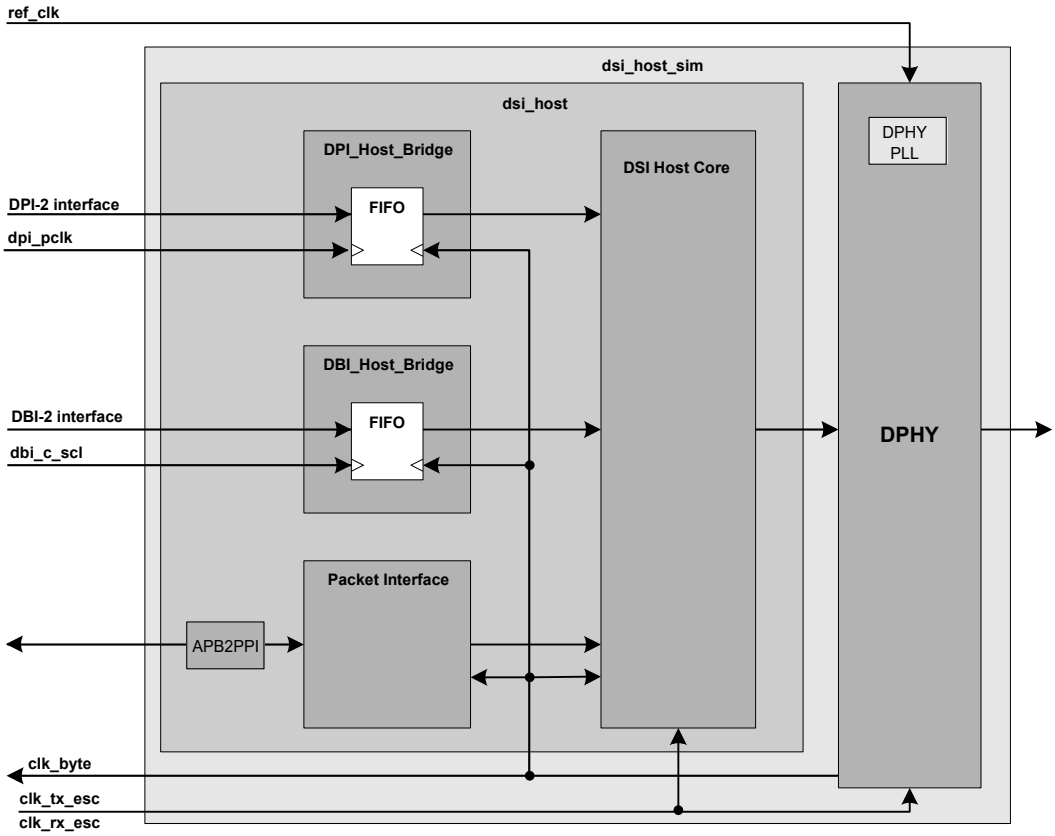


Figure 190. DSI Host Controller Clocking Block Diagram

ref_clk

The reference clock is used by the DPHY (DPHY PLL) to generate the High Speed MIPI clock and High Speed Data Lane signaling. The frequency of the resulting clock generated from DPHY PLL is typically equal to the high speed bit rate of the data lanes. The DPHY will also use this clock to generate the clk_byte clock output used by the Host DSI Controller and user application logic.

The MIPI DPHY PLL must be setup through the MIPI_DSI_HOST_FSL_IP1_DPHY INTFCx registers, prior to starting DSI Host Controller and DPHY operation. For details, please refer to MIPI_DSI_HOST IP1 DPHY INTFC Memory section.

clk_byte

The clk_byte clock is generated by the DPHY (DPHY PLL) and the frequency is 1/8th the data rate of the DPHY Data Lane High Speed data rate. A DPHY configured to generate High Speed data at 1Gbps/lane would generate a 125MHz clk_byte clock.

The clk_byte clock is used by the DSI Host Controller to interface to the MIPI Tx DPHY High Speed PPI interface. The DSI Host Controller's Packet Interface is synchronous to the clk_byte clock.

dpi_pclk

The dpi_pclk clock is used on the Host DPI-2 interface. All of the Host DPI-2 signals are synchronous to this clock. The DSI Host Controller's DPI Bridge module handles transferring video data received in the dpi_pclk clock domain over to the clk_byte clock domain.

The dpi_pclk and clk_byte frequencies are related by the following formula: clk_byte_freq >= dpi_pclk_freq * DPI_pixel_size / (8 * (cfg_num_lanes + 1))

cfg_num_lanes = the configuration port setting that selects the number of active MIPI DPHY data lanes clk_byte_freq = frequency of clk_byte which is 1/8th the High Speed data lane rate. dpi_pclk_freq = frequency of the dpi_pclk clock on the DPI-2 interface. DPI_pixel_size = size of pixels, in bits, on the DPI-2 interface

If the clk_byte frequency does not meet this requirement, the MIPI interface will not be able to keep up with the video stream on the DPI-2 interface resulting in dropped video lines.

clk_tx_esc

The clk_tx_esc clock is used by the MIPI Tx DPHY for state control and low power data transmission. The DSI Host Controller also uses clk_tx_esc for the portion of controller logic that interfaces to the MIPI Tx DPHY that are synchronous to clk_tx_esc. The frequency of clk_tx_esc is defined by the requirements of the MIPI interface and the MIPI Tx DPHY. For details about the clk_tx_esc source, please refer to the chip-specific MIPI-DSI clocking section.

dbi_c_scl

The dbi_c_scl clock is used by the optional DBI-C interface. The MIPI DBI-2 specification defines the minimum period for dbi_c_scl to be 80ns. The DSI Host Controller DBI-2 bridge module handles the clock domain crossing from dbi_c_scl clock domain to the clk_byte clock domain.

54.3.8 Reset and Initialization

Reset and Initialization procedure for the MIPI_DSI_HOST Controller is as follows:

1. Assert all resets through the chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n
2. Setup MIPI_DSI_HOST related clocks through chip-specific registers provided for this purpose.
3. Setup MIPI_DSI_HOST application by programming all MIPI_DSI_HOST Controller and DSI DPHY required parameters (refer to MIPI_DSI_HOST Memory Map/Register for details).
4. Deassert all remaining resets through the Chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n
5. DSI Host Controller should be ready for use

54.4 DSI D-PHY

The figure below shows the block diagram of the MIPI_DSI_HOST D-PHY. It consists of a clock lane module and two data lane modules. Each of these PHY Lane Modules communicates through a differential line to a complementary PHY at the other side of the lane interconnect.

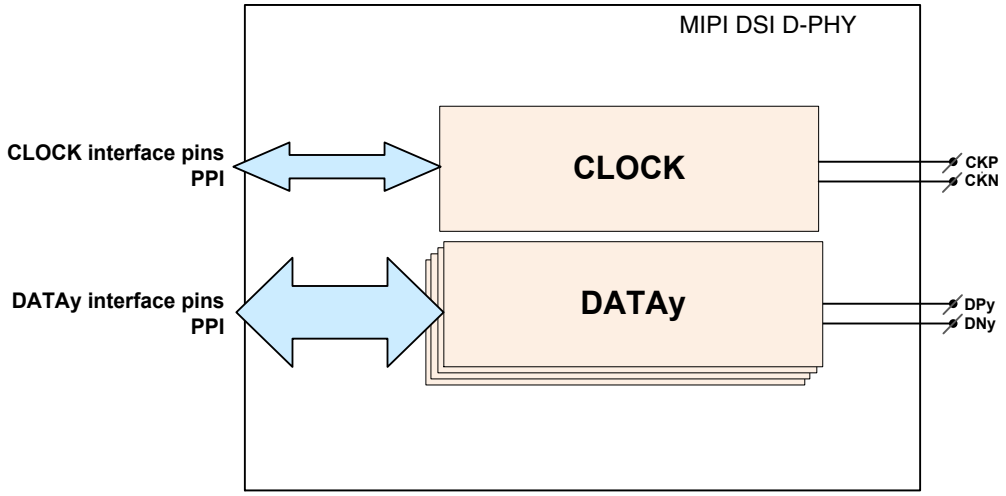


Figure 191. MIPI D-PHY Block Diagram

The figure below shows the D-PHY Lane components:

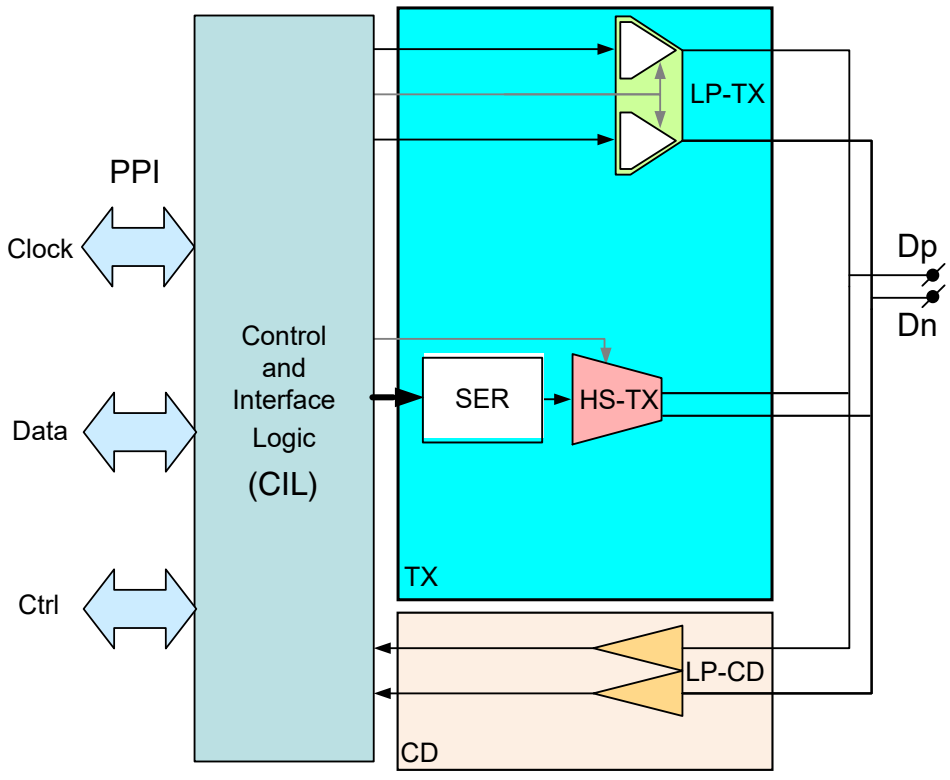


Figure 192. D-PHY Lane Diagram

The DATAy are bidirectional lanes and consist of High-Speed Transmitter (HS-TX), Low-Power Transmitter (LP-TX), Serializer (SER), and Low- Power Contention Detector (LP-CD).

The Clock lane is a bidirectional lane and consists of High-Speed Transmitter (HS-TX), and Low-Power Transmitter (LP-TX).

The Control and Interface Logic (CIL) module interfaces with the Protocol and determines the global operation of the Lane Module. The interface between the D-PHY and the protocol is called the PHY-Protocol Interface (PPI).

During normal operation, a Lane switches between Low-Power and High-Speed mode. Bidirectional Lanes can also switch communication direction. The change of operating mode or direction requires enabling and disabling of certain electrical functions. These enable and disable events do not cause glitches on the Lines that would result in a detection of an incorrect signal level. Therefore, all mode and direction changes are smooth to always ensure a proper detection of the Line signals.

54.4.1 High-Speed Transmitter (HS-TX)

The differential output voltage VOD is defined as the difference of the voltages V_{DP} and V_{DN} at the Dp and Dn pins, respectively.

$VOD = V_{DP} - V_{DN}$

The output voltages V_{DP} and V_{DN} at the Dp and Dn pins will not exceed the high-speed output high voltage V_{OHHS}. The common-mode voltage V_{CMTX} is defined as the arithmetic mean value of the voltages at the Dp and Dn pins:

$V_{CMTX} = (V_{DP} + V_{DN})/2$

V_{OD} and V_{CMTX} are graphically shown in Figure 193 for ideal HS signals. Figure 194 shows single ended HS signals with the possible kinds of distortion of the differential output and common mode voltages. V_{OD} and V_{CMTX} may be slightly different for driving a Differential-1 or a Differential-0 on the pins.

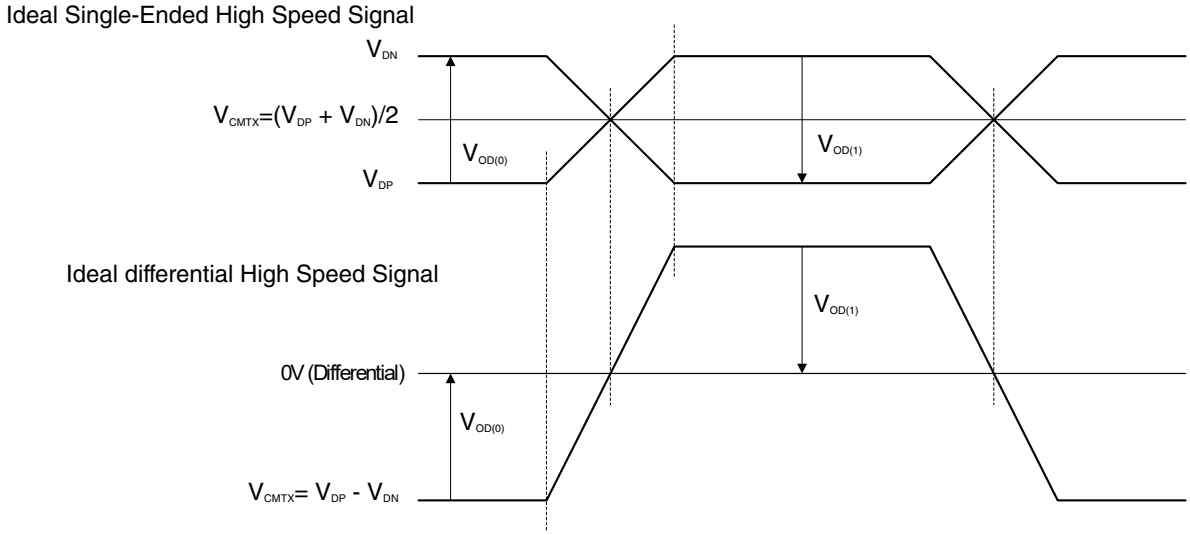


Figure 193. Ideal Single-ended and Resulting Differential HS Signals

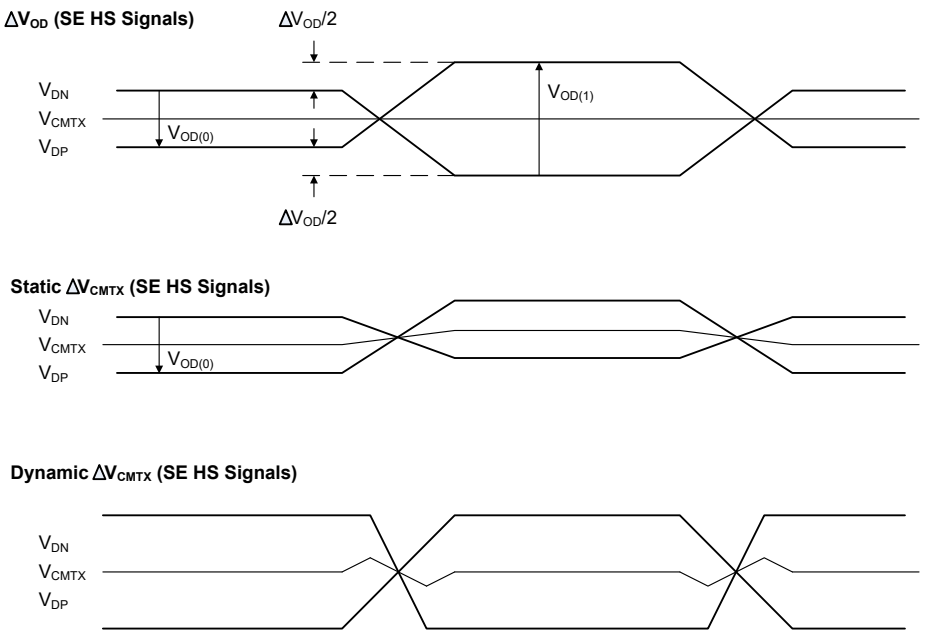


Figure 194. Possible ΔV_{CMTX} and ΔV_{OD} Distortions of the Single-Ended HS Signals

The output differential voltage mismatch ΔV_{OD} is defined as the difference of the absolute values of the differential output voltage in the Differential-1 state $V_{OD}(1)$ and the differential output voltage in the Differential-0 state $V_{OD}(0)$. This is expressed by:

$\Delta V = |V_{OD(1)}| - |V_{OD(0)}|$

If $V_{CMTX(1)}$ and $V_{CMTX(0)}$ are the common-mode voltages for static Differential - 1 and Differential - 0 states respectively, then the common-mode reference voltage is defined by:

$V_{CMTX,\ REF} = (V_{\ CMTX(1)} + V_{\ CMTX(0)})/2$

The transient common-mode voltage variation is defined by:

$\Delta V_{CMTX\ (t)} = V_{CMTX\ (t)} - V_{\ CMTX\ ,REF(t)}$

The static common-mode voltage mismatch between the Differential-1 and Differential-0 state is given by:

$\Delta V_{CMTX\ (1,0\ )} = (V_{CMTX\ (1)} - V_{CMTX\ (0)})/2$

A test circuit for the measurement of V_{OD} and V_{CMTX} is shown:

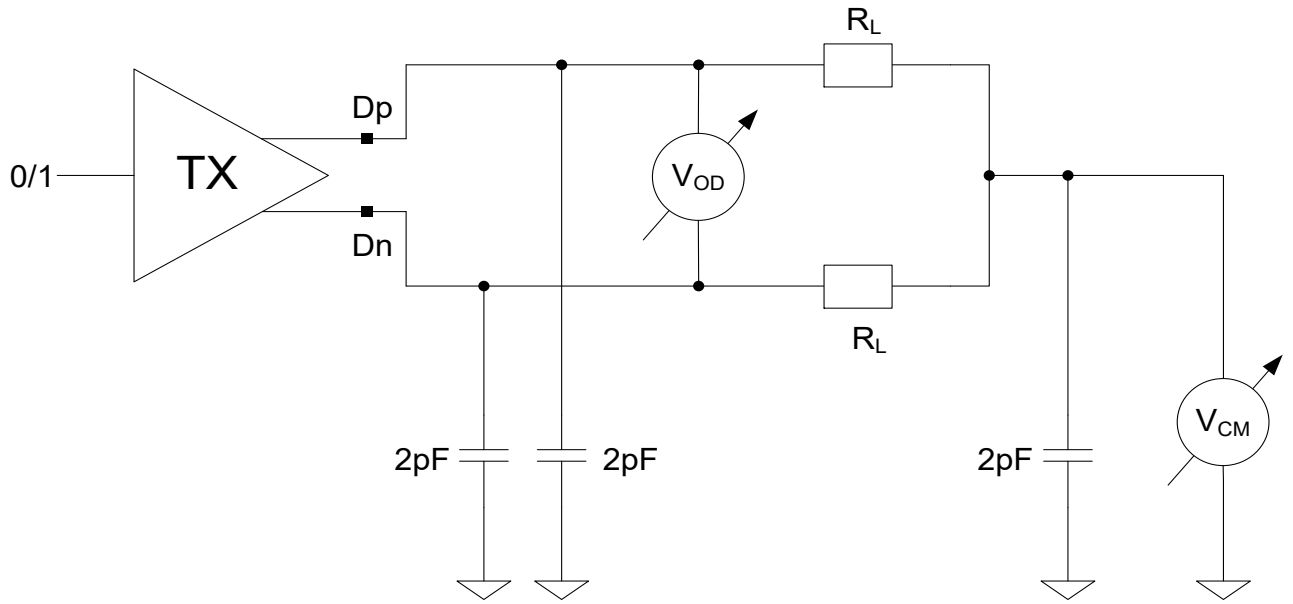


Figure 195. Test Circuit for V_{CMTX} and V_{OD} measurement

The single-ended output impedance of the transmitter at both the Dp and Dn pins is denoted by Z_{OS}. ΔZ_{OS} is the mismatch of the single ended output impedances at the Dp and Dn pins, denoted by Z_{OSDP} and Z_{OSDN} respectively. This mismatch is defined as the ratio of the absolute value of the difference of Z_{OSDP} and Z_{OSDN} and the average of those impedances:

$\Delta Z_{OS} = (2*|Z_{OSDP} - Z_{OSDN}|)/(Z_{OSDP} + Z_{OSDN})$

The output impedance Z_{OS} and the output impedance mismatch ΔZ_{OS} is compliant with High Speed Transmitter DC Specifications for both the Differential-0 and Differential-1 states for all allowed loading conditions. It is recommended the design keep the output impedance during state transitions as close as possible to the steady state value. The output impedance Z_{OS} can be determined by injecting an AC current into the Dp and Dn pins and measuring the peak-to-peak voltage amplitude. The test circuit for such a measurement is shown in the following figure, where R_L are load resistors and $V_{TEST,PP}$ is a test signal applied at the common-mode node via an AC coupling capacitor CAC. The frequency of the test signal is f_{TEST}.

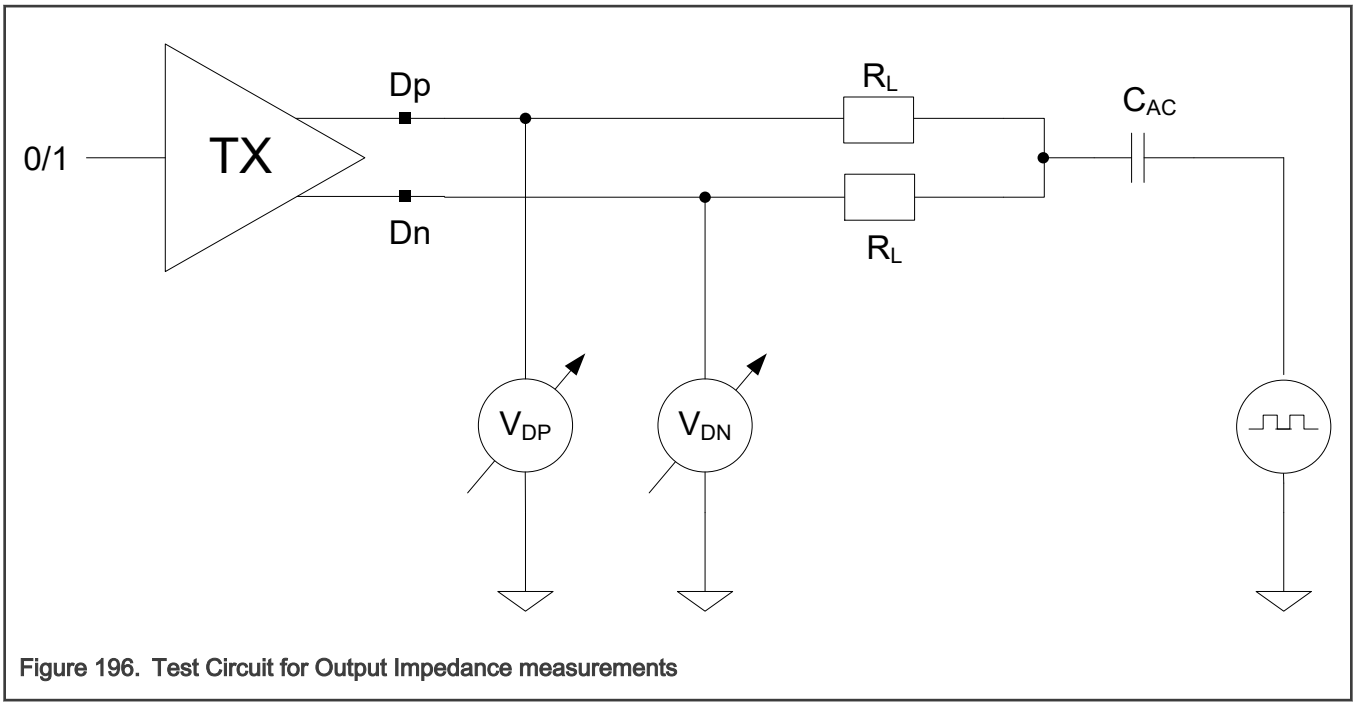


Figure 196. Test Circuit for Output Impedance measurements

Assuming negligible effect of the coupling capacitor, the example circuit gives the following relation between peak-to-peak pin voltage $V_{PIN,PP}$ and Z_{OS}. This equation can be used for both Dn and Dp pins individually.

$Z_{OS} = (V_{PIN,PP} * R_L)/(V_{TEST,PP} – Z_{PIN,PP})$

Where $V_{PIN,PP}$ is the peak-to-peak voltage at the pin Dp or Dn.

54.4.2 Low-Power Transmitter (LP-TX)

The Low-Power transmitter is a slew-rate controlled push-pull driver. It is used for driving the Lines in all Low-Power operating modes. The LP transmitter has been optimized to minimize static power consumption. The slew-rate of signal transitions is bounded in order to keep EMI low. The LP transmitter will not drive the pad pin potential statically beyond the maximum value of V_{OH}. V_{OH} is the thevenin output, high-level voltage in the high-level state, when the pad pin is not loaded. V_{OL} is the thevenin output, low-level voltage in the LP transmit mode. This is the voltage at an unloaded pad pin in the low-level state.

54.4.3 Low-Power Contention Detector (LP-CD)

Contention can be inferred from any of the following conditions:

- LP high fault is detected when the LP transmitter is driving high and the pin voltage is less than VIL.
- LP low fault is detected when the LP transmitter is driving low and the pad pin voltage is greater than VIHCD. An LP low fault will not be detected when the pin voltage is less than VILCD.

The general operation of a contention detector is similar to that of an LP receiver with lower threshold voltages. Although the DC specifications differ, the AC specifications of the LP-CD are defined to match those of the LP receiver. The LP-CD sufficiently filters the input signal to avoid false triggering on short events

The LP-CD threshold voltages (VILCD, VIHCD) are shown along with the normal signaling voltages:

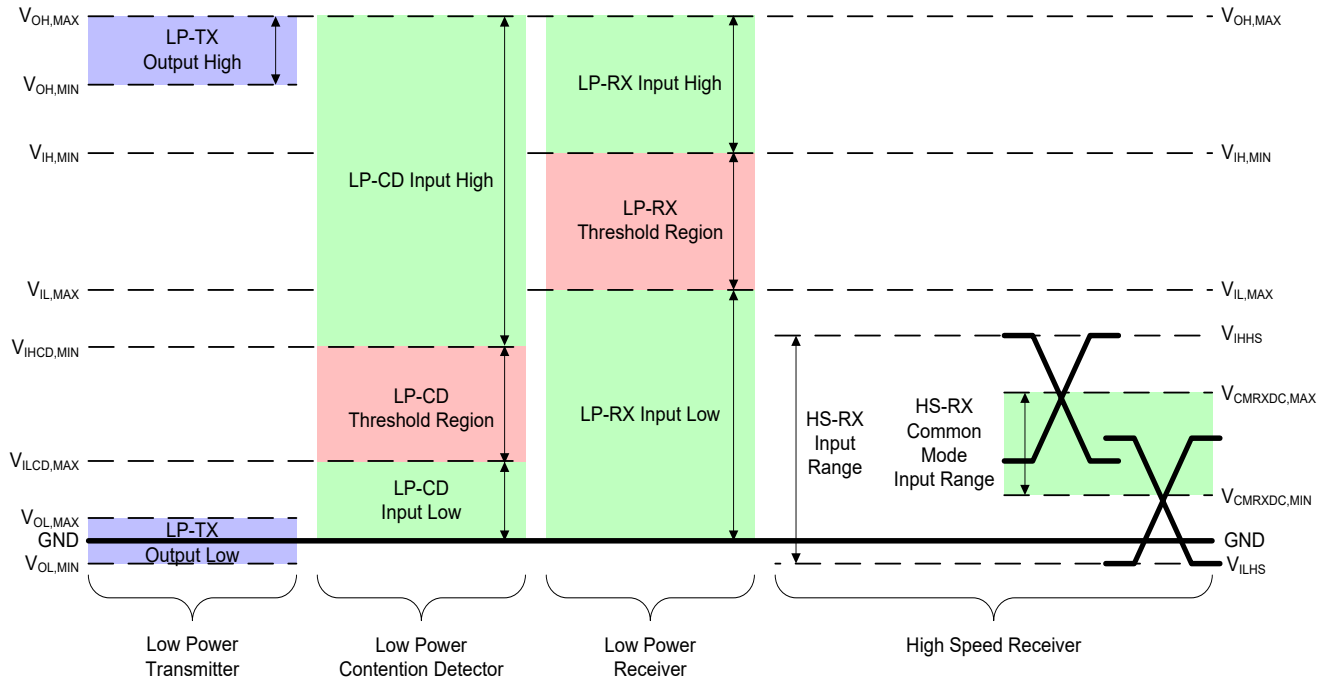


Figure 197. Signaling and MIPI Contention Voltage Levels

54.4.4 Input Characteristics

No structure within the PHY may be damaged when a DC signal that is within the signal voltage range VPIN is applied to a pad pin for an indefinite period of time. $V_{PIN(absmax)}$ is the maximum transient output voltage at the transmitter pin. The voltage on the transmitter's output pin will not excel $V_{PIN,MAX}$ for a period greater than $T_{VPIN(absmax)}$. When the PHY is in the low-power receive mode the pad pin leakage current will be I_{LEAK} when the pad signal voltage is within the signal voltage range of V_{PIN}. The specification of ILEAK assures interoperability of any PHY in the LP mode by restricting the maximum load current of an LP transmitter. The test circuit for leakage current measurement is shown:

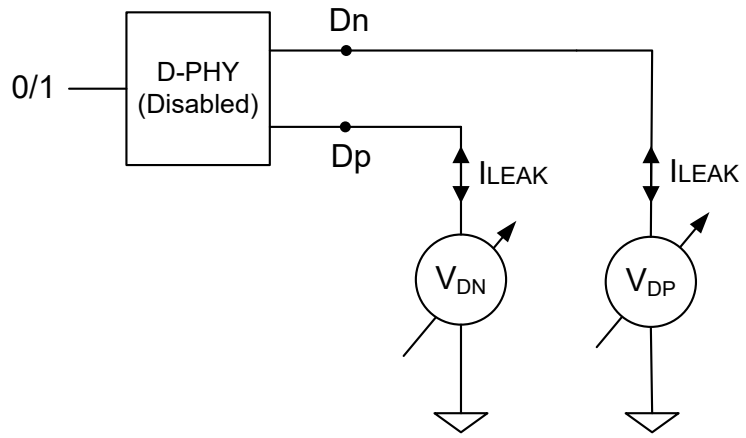


Figure 198. Pin-Leakage Measurement DC specification

54.4.5 High-Speed Serializer

The HS-Serializer converts 8-bits parallel data into 1 bit data stream to be transmitted by HSTX module. The CIL engine should clock out 8-bits parallel data, HSTX_DATA[7:0], on D0_HS_BYTE_CLKS rising edges. Then the Serializer will capture HSTX_DATA[7:0] on D0_HS_BYTE_CLKS rising edges and the signal connected to HSTX_DATA[0] is transmitted first. There is setup timing constraint and hold timing constraint designated to the rising edge of D0_HS_BYTE_CLKS as constrained in the liberty view provided.

The figure shows the relationship between HS_TXCLKP, HS_TXCLKN, HS_SER_LD, HS_BYTE_CLKS and HSTX_DATA[7:0]. It also shows how the 8-bits of parallel bits are mapped into the serial link:

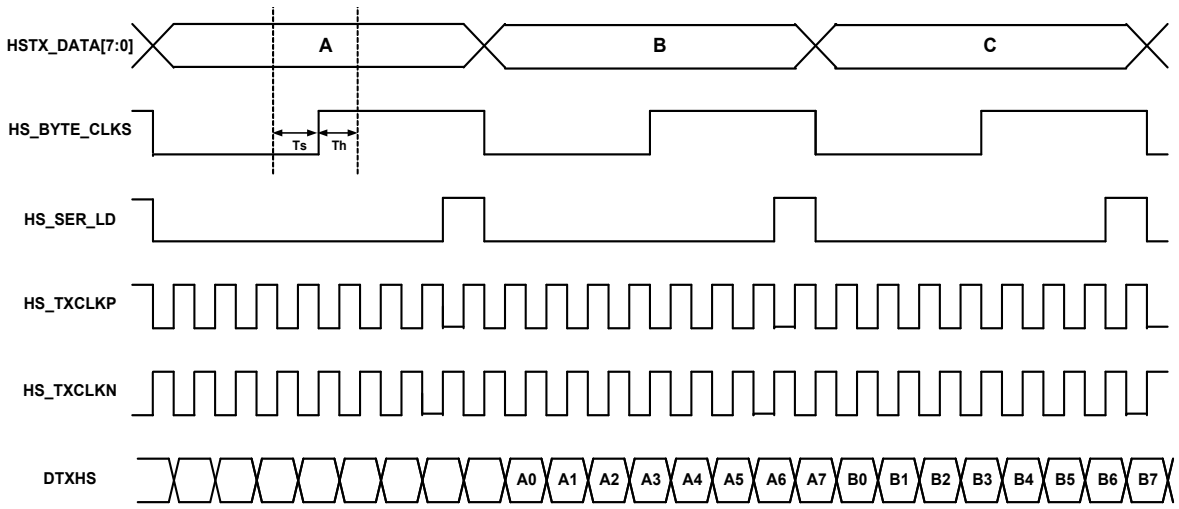


Figure 199. Serializer Timing Diagram

54.4.6 DSI D-PHY Signals

This section details the interface signals for MIPI DSI D-PHY. For detailed signal spec, please refer to the chip data sheet.

Table 532. DSI DPHY PPI Signals

Signal	Type	Description
TxClkEsc	Clock	Escape mode Transmit Clock. This clock is directly used to generate escape sequences. The period of this clock determines the symbol time for low power signals. It is therefore constrained by the normative part of the DPHY specification. **NOTE** The max frequency of TxClkEsc is 20 MHz **NOTE** The Min frequency of TxClkEsc is 12 MHz.

Table continues on the next page...

Table 532. DSI DPHY PPI Signals (continued)

Signal	Type	Description
TxByteClkHS	Clock	High-Speed Transmit Byte Clock. This is used to synchronize PPI signals in the High-Speed transmit clock domain. It is recommended that all transmitting Data Lane Modules share on TxByteClkHS signal. The frequency of TxByteClkHS is exactly 1/8 the High-Speed bit rate.

54.4.7 Operating Modes

54.4.7.1 High-Speed Transmission

The figure shows an example of a High-Speed transmission. While TxRequestHS is low, the Lane Module ignores the value of TxDataHS. To begin transmission, the protocol drives TxDataHS with the first byte of data and asserts TxRequestHS. This data byte is accepted by the PHY on the first rising edge of TxByteClkHS with TxReadyHS also asserted. At this point, the protocol logic drives the next data byte onto TxDataHS. After every rising clock cycle with TxReadyHS active, the protocol supplies a new valid data byte or ends the transmission. After the last data byte has been transferred to the Lane Module, TxRequestHS is driven low to cause the Lane Module to stop the transmission and enter Stop state. The minimum number of bytes transmitted could be as small as one.

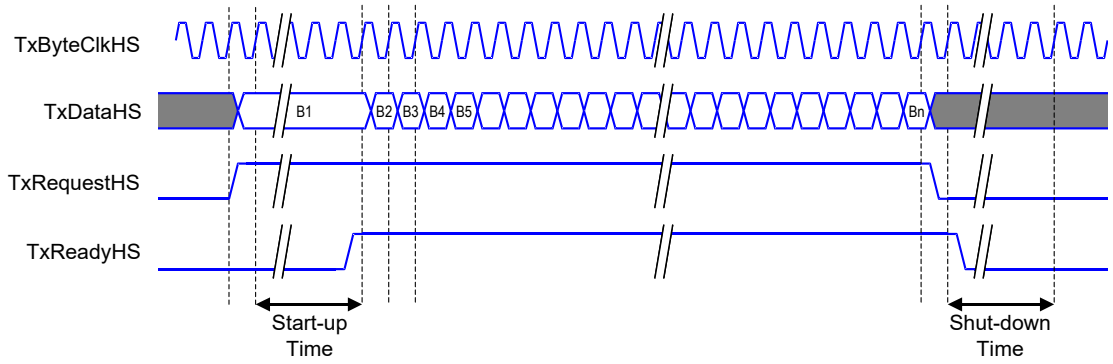


Figure 200. High-Speed Transmission

54.4.7.2 Low-Power Data Transmission

For Low-Power data transmission the TxClkEsc is used. The Protocol directs the Data Lane to enter Low-Power data transmission Escape mode by asserting TxRequestEsc with TxLpdtEsc high. The Low-Power transmit data is transferred on the TxDataEsc lines when TxValidEsc and TxReadyEsc are both active at a rising edge of TxClkEsc. The byte is transmitted in the time after the TxDataEsc is accepted by the Lane Module (TxValidEsc = TxReadyEsc = high). The Protocol knows the byte transmission is finished when TxReadyEsc is asserted. After the last byte has been transmitted, the protocol de-asserts TxRequestEsc to end the Low-Power data transmission. This causes TxReadyEsc to return low. Whenever TxRequestEsc transitions from high-to-low, it always remains in the low state for a minimum of two TxClkEsc clock cycles. The figure shows an example Low-Power data transmission operation.

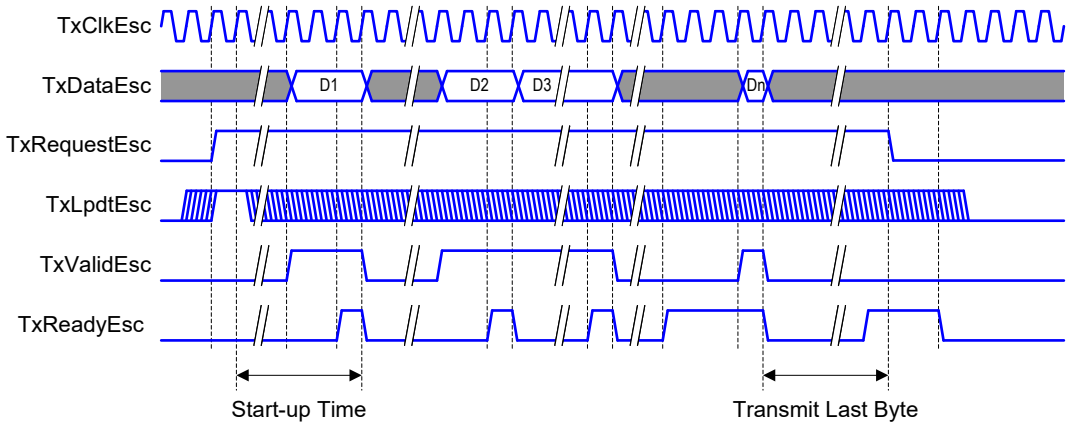


Figure 201. Low-Power Data Transmission

54.4.7.3 Turn-around

If the local side and remote side Lane Modules are both bi-directional, it is possible to turn around the Link for Escape mode signaling. Which side is allowed to transmit is determined by passing a “token” back and forth. That is, the side currently transmitting passes the token to the receiving side. If the receiving side acknowledges the turn-around request, as indicated by driving the appropriate line state, the direction is switched. The figure shows an example of two turn-around events. At the beginning, the local side is the transmitter, as shown by Direction=0. When the protocol on this side wishes to turn the Lane around (i.e. give the token to the other side), it asserts TurnRequest for at least one cycle of TxClkEsc. This initiates the turn around procedure. The remote side acknowledges the turn-around request by driving the appropriate states on the Lines. When this happens, the local Direction signal changes from transmit (0) to receive (1). The remote side initiates a turn-around request, passing the token back to the local side. When this happens, the local Direction signal changes back to transmit (0). The transmitter is in control of the Link direction and decides when to turn the Link around, passing control to the receiver. If the remote side does not acknowledge the turnaround request, the Direction signal does not change.

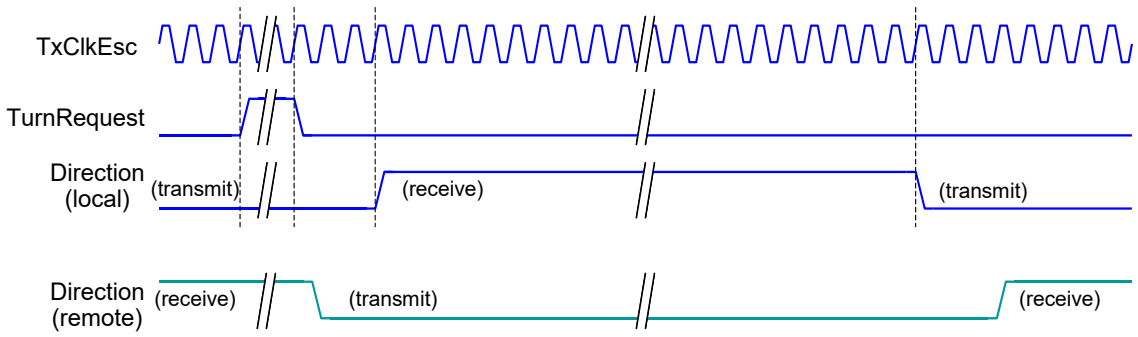


Figure 202. Turn-around Actions Transmit-to-Receive and Back to Transmit

54.5 DPHY Programming

The following sections detail the programming of the MIPI_DSI_HOST DPHY high-speed timers.

High-Speed Prepare Timer ($T_{HS-PREPARE}$)

Calculate the value for clock lane high-speed prepare timer (mc_PRG_HS_PREPARE) using the formula below:

Assuming TxClkEsc = 20 MHz (T_{per} = 50 ns)

When mc_PRG_HS_PREPARE = 0, $T_{CLK\text{-}PREPARE}$= 1 * TxClkEsc Period = 50 ns

When mc_PRG_HS_PREPARE = 1, $T_{CLK\text{-}PREPARE}$ = 1.5 * TxClkEsc Period = 75 ns

Calculate the value for data lane high-speed prepare timer (m_PRG_HS_PREPARE) using the formula below:

Assuming TxClkEsc (TLPX) = 20 MHz (T_{per} = 50 ns)

When m_PRG_HS_PREPARE = 00, $T_{HS\text{-}PREPARE}$ = 1 * TxClkEsc Period = 50 ns

When m_PRG_HS_PREPARE = 01, $T_{HS\text{-}PREPARE}$ = 1.5 * TxClkEsc Period = 75 ns

When m_PRG_HS_PREPARE = 10, $T_{HS\text{-}PREPARE}$ = 2 * TxClkEsc Period = 100 ns

When m_PRG_HS_PREPARE = 11, $T_{HS\text{-}PREPARE}$ = 2.5 * TxClkEsc Period = 125 ns

High-Speed Zero Timer($T_{HS\text{-}ZERO}$)

Calculate the value for clock lane high-speed zero timer (mc_PRG_HS_ZERO) using the formula below:

$T_{CLK\text{-}ZERO}$ = (mc_PRG_HS_ZERO + 4) * (TxByteClkHS Period)

TxByteClkHS = 1/8 Data Rate

When mc_PRG_HS_ZERO = 000110 @ 250 Mb/s, $T_{CLK\text{-}ZERO}$ = (6+4) * 32 ns = 320 ns

When mc_PRG_HS_ZERO = 100000 @ 1 Gb/s, $T_{CLK\text{-}ZERO}$ = (32+4) * 8 ns = 288 ns

Calculate the value for data lane high-speed zero timer (m_PRG_HS_ZERO) using the formula below:

$T_{HS\text{-}ZERO}$ = (m_PRG_HS_ZERO + 6)*(TxByteClkHS Period)

TxByteClkHS = 1/8 Data Rate

When m_PRG_HS_ZERO = 00001 @ 250 Mb/s, $T_{HS\text{-}ZERO}$ = (1+6) * 32 ns = 224 ns

When m_PRG_HS_ZERO = 01001 @ 1 Gb/s, $T_{HS\text{-}ZERO}$ = (9+6) * 8 ns = 120 ns

High-Speed Trail Timer ($T_{HS\text{-}TRAIL}$)

Calculate the value for the high-speed trail timer (PRG_HS_TRAIL) using the formula below:

$T_{HS\text{-}TRAIL}$= (PRG_HS_TRAIL)*(TxByteClkHS Period)

TxByteClkHS = 1/8 Data Rate

When PRG_HS_TRAIL = 0100 @ 250 Mb/s, >$T_{HS\text{-}TRAIL}$ = 4 * 32 ns = 128 ns

When PRG_HS_TRAIL = 1100 @ 1 Gb/s, >$T_{HS\text{-}TRAIL}$ = 12 * 8 ns = 96 ns

54.6 Packet Data and Pixel Formats

The DSI Host Controller Core and the external DSI Peripheral explicitly handle and exchanges 32-bit words encapsulated as packet data. The mapping between pixel formats and pixel data must be done on both sides by user application on both sides of the DSI link. The DSI spec gives many examples of common pixel formats. The DSI Host Controller Core and external DSI Peripheral have no preference or limitation on which pixel formats it will handle. The IP which provides video source packet data through DPI-2 interface (e.g. LCDIF) onto DSI should also be configured by the user application to match the pixel formats expected by the DSI Host Controller.

The DPI-2/DBI-2 interface core handles pixel to byte packing and byte to pixel unpacking according to the MIPI_DSI_HOST specification.

54.7 MIPI_DSI_HOST Memory Map/Register Definition

MIPI_DSI_HOST Controller Register Groups	Offset
DSI_HOST	0x00
DBI_INTFC	0x100
DPI_INTFC	0x200
APB_PKT_IF	0x280
DPHY_INTFC	0x300

54.7.1 MIPI DSI Host register descriptions

54.7.1.1 MIPI_DSI_HOST memory map

MIPI_DSI_HOST base address: 4041_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Configure Number of Lanes (CFG_NUM_LANES)	32	RW	0000_0000h
4h	Configure Non-continuous Clock (CFG_NONCONTINUOUS_CLK)	32	RW	0000_0000h
8h	Configure Pre Clock Periods (CFG_T_PRE)	32	RW	0000_0000h
Ch	Configure Post Clock Periods (CFG_T_POST)	32	RW	0000_0000h
10h	Configure Gap Clock Periods (CFG_TX_GAP)	32	RW	0000_0000h
14h	Configure Autoinsert EOTP (CFG_AUTOINSERT_EOTP)	32	RW	0000_0000h
18h	Configure Extra Commands after EOTP (CFG_EXTRA_CMDS_AFTER_EOTP)	32	RW	0000_0000h
1Ch	Configure High Speed Tx Timeout Count (CFG_HTX_TO_COUNT)	32	RW	0000_0000h
20h	Configure Low Power Rx Timeout Count (CFG_LRX_H_TO_COUNT)	32	RW	0000_0000h
24h	Configure Bus Turn Around Timeout Count (CFG_BTA_H_TO_COUNT)	32	RW	0000_0000h
28h	Configure Twakeup (CFG_TWAKEUP)	32	RW	0000_0000h
2Ch	Configure Status Out (CFG_STATUS_OUT)	32	R	0000_0000h
30h	Receive Error Status (RX_ERROR_STATUS)	32	R	0000_0000h
34h	ULPS Enable (ULPS_ENABLE)	32	RW	0000_0000h
38h	ULPS Active (ULPS_ACTIVE)	32	R	0000_0000h
3Ch	High Speed Mode Enable (HS_MODE_ENABLE)	32	RW	0000_0000h
40h	Host Turnaround (HOST_TURNAROUND)	32	RW	0000_0000h
44h	PHY Direction (PHY_DIRECTION)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
48h	PHY Ready (PHY_RDY)	32	R	0000_0001h
100h	Pixel Payload Size (CFG_DBI_PIXEL_PAYLOAD_SIZE)	32	RW	0000_0000h
104h	Configure DBI Pixel FIFO Send Level (CFG_DBI_PIXEL_FIFO_SEND_LEVEL)	32	RW	0000_0000h
108h	DBI Pixel Format (CFG_DBI_PIXEL_FORMAT)	32	RW	0000_0000h
10Ch	DBI Underrun Error (DBI_UNDERRUN_ERR)	32	R	0000_0000h
110h	DBI Overflow Error (DBI_OVERFLOW_ERR)	32	R	0000_0000h
200h	Configure DPI Pixel Payload Size (CFG_DPI_PIXEL_PAYLOAD_SIZE)	32	RW	0000_0000h
204h	Configure DPI Pixel FIFO Send Level (CFG_DPI_PIXEL_FIFO_SEND_LEVEL)	32	RW	0000_0000h
208h	Configure DPI Interface Color Coding (CFG_DPI_INTERFACE_COLOR_CODING)	32	RW	0000_0000h
20Ch	Configure DPI Pixel Format (CFG_DPI_PIXEL_FORMAT)	32	RW	0000_0000h
210h	Configure DPI vsync Polarity (CFG_DPI_VSYNC_POLARITY)	32	RW	0000_0001h
214h	Configure DPI hsync Polarity (CFG_DPI_HSYNC_POLARITY)	32	RW	0000_0001h
218h	Configure DPI Video Mode (CFG_DPI_VIDEO_MODE)	32	RW	0000_0000h
21Ch	Configure DPI horizontal front porch (CFG_DPI_HFP)	32	RW	0000_0000h
220h	Configure DPI Horizontal Back Porch (CFG_DPI_HBP)	32	RW	0000_0000h
224h	Configure DPI Horizontal Sync Width (CFG_DPI_HSA)	32	RW	0000_0000h
228h	Enable Multiple Packets (CFG_DPI_ENABLE_MULT_PKTS)	32	RW	0000_0000h
22Ch	Configure DPI Vertical Back Porch (CFG_DPI_VBP)	32	RW	0000_0000h
230h	Configure DPI Vertical Front Porch (CFG_DPI_VFP)	32	RW	0000_0000h
234h	Configure DPI BLLP Mode (CFG_DPI_BLLP_MODE)	32	RW	0000_0000h
238h	Configure DPI Blank Packet in BLLP (CFG_DPI_USE_NULL_PKT_BLLP)	32	RW	0000_0000h
23Ch	Configure DPI Vertical Active (CFG_DPI_VACTIVE)	32	RW	0000_0000h
240h	Configure DPI Virtual Channel (CFG_DPI_VC)	32	RW	0000_0000h
244h	DPI Underrun Error (DPI_UNDERRUN_ERR)	32	R	0000_0000h
280h	Transmit Payload (TX_PAYLOAD)	32	RW	0000_0000h
284h	Packet Control (PKT_CONTROL)	32	RW	0000_0000h
288h	Send Packet (SEND_PACKET)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
28Ch	Packet Status (PKT_STATUS)	32	R	0000_0000h
290h	Packet FIFO Write (PKT_FIFO_WR_LEVEL)	32	R	0000_0000h
294h	Packet FIFO Read (PKT_FIFO_RD_LEVEL)	32	R	0000_0000h
298h	Packet Rx Payload (PKT_RX_PAYLOAD)	32	R	0000_0000h
29Ch	Packet Rx Packet Header (PKT_RX_PKT_HEADER)	32	R	0000_0000h
2A0h	Interrupt Status (IRQ_STATUS)	32	R	0000_0000h
2A4h	Interrupt Status 2 (IRQ_STATUS2)	32	R	0000_0000h
2A8h	Mask Interrupt (IRQ_MASK)	32	RW	0000_0000h
2ACh	Interrupt Mask 2 (IRQ_MASK2)	32	RW	0000_0000h
300h	DPHY Power Down (PD_DPHY)	32	RW	0000_0001h
304h	Program T_HS_PREPARE (M_PRG_HS_PREPARE)	32	RW	0000_0000h
308h	Program HS T_CLK_PREPARE (MC_PRG_HS_PREPARE)	32	RW	0000_0000h
30Ch	Program T_HS_ZERO (M_PRG_HS_ZERO)	32	RW	0000_0000h
310h	Program T_CLK_ZERO (MC_PRG_HS_ZERO)	32	RW	0000_0000h
314h	Program T_HS_TRAIL (M_PRG_HS_TRAIL)	32	RW	0000_0000h
318h	Program T_CLK_TRAIL (MC_PRG_HS_TRAIL)	32	RW	0000_0000h
31Ch	DPHY TST Input (TST)	32	RW	0000_0025h
320h	RTERM Select (RTERM_SEL)	32	RW	0000_0000h
324h	Power Down Auto Enable (AUTO_PD_EN)	32	RW	0000_0000h
328h	DPHY RXLPRP Input (RXLPRP)	32	RW	0000_0000h
32Ch	DPHY RXCDRP Input (RXCDRP)	32	RW	0000_0000h

54.7.1.2 Configure Number of Lanes (CFG_NUM_LANES)

Offset

Register	Offset
CFG_NUM_LANES	0h

Function

Sets the number of active lanes used for transmitting data.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														CFG_NUM_LANES	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 CFG_NUM_LANES	Configure Number of Lanes Sets the number of active lanes used for transmitting data. 00b - 1 Lane 01b - 2 Lanes 10b - 3 Lanes 11b - 4 Lanes

54.7.1.3 Configure Non-continuous Clock (CFG_NONCONTINUOUS_CLK)

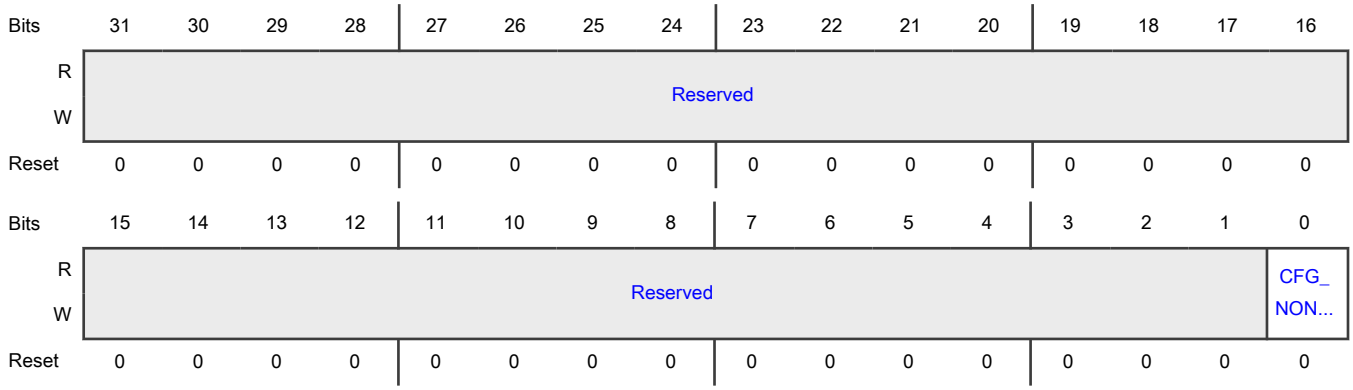
Offset

Register	Offset
CFG_NONCONTINUOUS_CLK	4h

Function

Sets the Host Controller into non-continuous MIPI clock mode.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_NONCONTINUOUS_CLK	Configure Non-continuous Clock When in non-continuous clock mode, the high speed clock will transition into low power mode between transmissions. 0b - Continuous high speed clock 1b - Non-Continuous high speed clock

54.7.1.4 Configure Pre Clock Periods (CFG_T_PRE)

Offset

Register	Offset
CFG_T_PRE	8h

Function

Sets the number of byte clock periods pre clock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CFG_T_PRE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 CFG_T_PRE	Configure Pre Clock Periods Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after enabling the clock lane for HS operation before enabling the data lanes for HS operation.

54.7.1.5 Configure Post Clock Periods (CFG_T_POST)

Offset

Register	Offset
CFG_T_POST	Ch

Function

Sets the number of byte clock periods post clock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CFG_T_POST							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 CFG_T_POST	Configure Post Clock Periods Sets the number of byte clock periods ('clk_byte' input) to wait before putting the clock lane into LP mode after the data lanes have been detected to be in Stop State.

54.7.1.6 Configure Gap Clock Periods (CFG_TX_GAP)

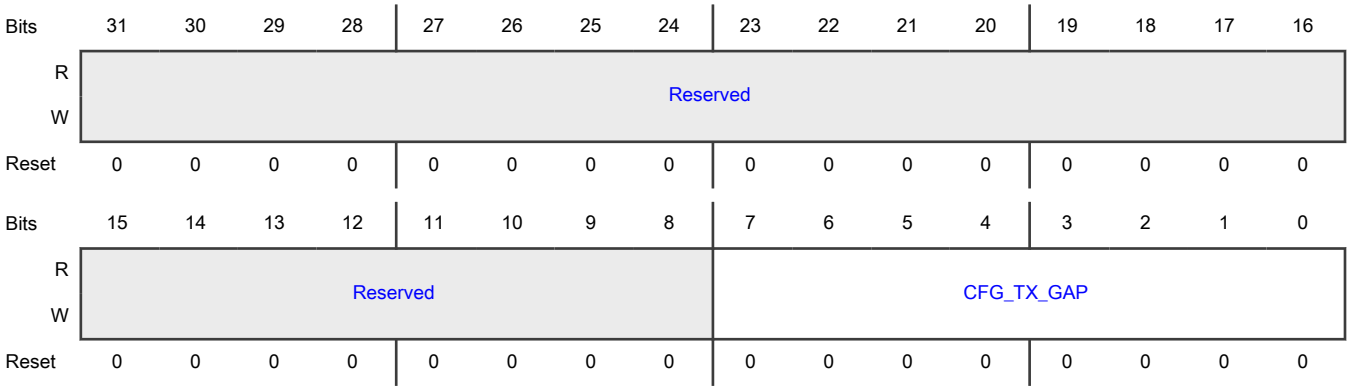
Offset

Register	Offset
CFG_TX_GAP	10h

Function

Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after the clock lane has been put into LP mode before enabling the clock lane for HS mode again.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 CFG_TX_GAP	Configure Gap Clock Periods This setting represents the THS-EXIT DPHY timing parameter. The minimum value for this port is 1.

54.7.1.7 Configure Autoinsert EOTP (CFG_AUTOINSERT_EOTP)

Offset

Register	Offset
CFG_AUTOINSERT_EOTP	14h

Function

Enables the Host Controller to automatically insert an EOTP short packet when switching from HS to LP mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															CFG_AUT...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 CFG_AUTOINSERT_EOTP	Configure Autoinsert EOTP 0b - EOTP is not automatically inserted 1b - EOTP is automatically inserted

54.7.1.8 Configure Extra Commands after EOTP (CFG_EXTRA_CMDS_AFTER_EOTP)

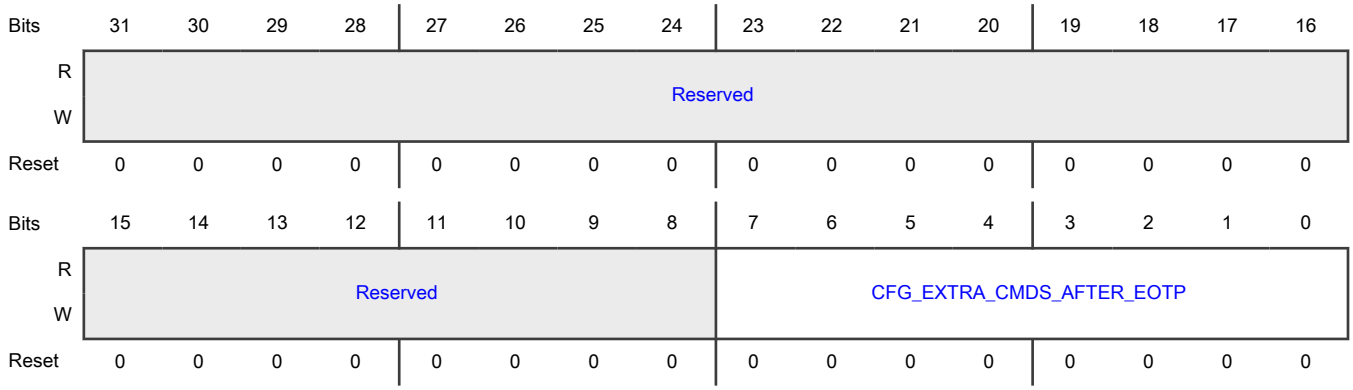
Offset

Register	Offset
CFG_EXTRA_CMDS_AFTER_EOTP	18h

Function

Configures the DSI Host Controller to send extra End Of Transmission Packets after the end of a packet.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 CFG_EXTRA_CMDS_AFTER_EOTP	Configure Extra Commands after EOTP The value is the number of extra EOTP packets sent.

54.7.1.9 Configure High Speed Tx Timeout Count (CFG_HTX_TO_COUNT)

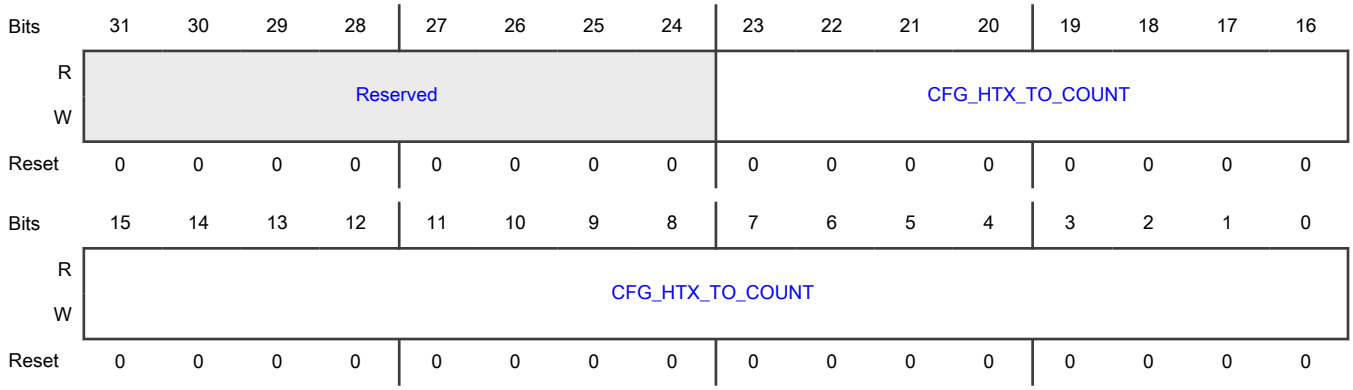
Offset

Register	Offset
CFG_HTX_TO_COUNT	1Ch

Function

Sets the value of the DSI Host High Speed TX timeout count.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 CFG_HTX_TO_COUNT	Configure High Speed Tx Timeout Count Sets the value of the DSI Host High Speed TX timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification.

54.7.1.10 Configure Low Power Rx Timeout Count (CFG_LRX_H_TO_COUNT)

Offset

Register	Offset
CFG_LRX_H_TO_COUNT	20h

Function
Sets the value of the DSI Host low power Rx timeout count.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								CFG_LRX_H_TO_COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_LRX_H_TO_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 CFG_LRX_H_TO_COUNT	Configure Low Power Rx Timeout Count Sets the value of the DSI Host low power Rx timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification.

54.7.1.11 Configure Bus Turn Around Timeout Count (CFG_BTA_H_TO_COUNT)

Offset

Register	Offset
CFG_BTA_H_TO_COUNT	24h

Function
Sets the value of the DSI Host Bus Turn Around (BTA) timeout.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								CFG_BTA_H_TO_COUNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CFG_BTA_H_TO_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 CFG_BTA_H_TO_COUNT	Configure Bus Turn Around Timeout Count Sets the value of the DSI Host Bus Turn Around (BTA) timeout in clk_byte clock periods that once reached will initiate a timeout error.

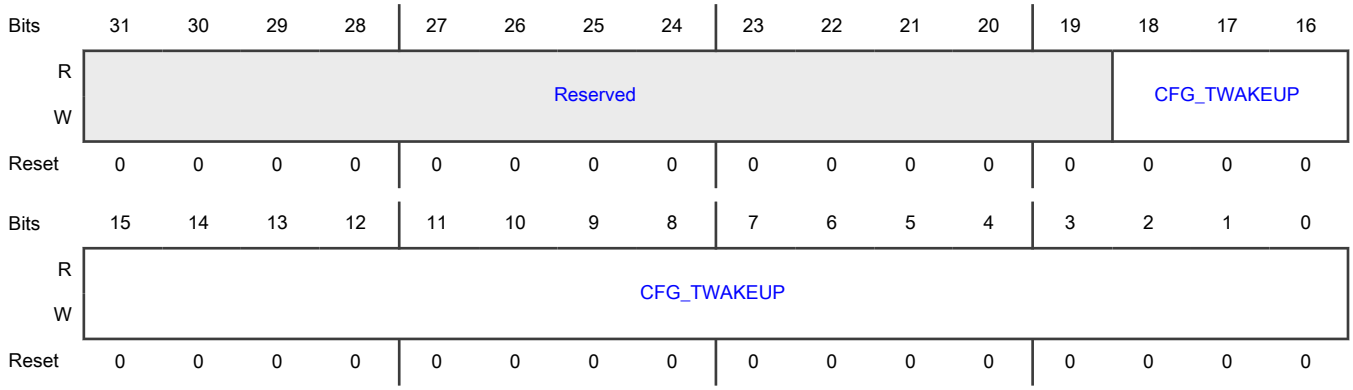
54.7.1.12 Configure Twakeup (CFG_TWAKEUP)

Offset

Register	Offset
CFG_TWAKEUP	28h

Function
Sets the number of clk_esc clock periods to keep a clock or data lane in Mark-1 state after exiting ULPS.

Diagram



Fields

Field	Function
31-19 —	Reserved
18-0 CFG_TWAKEUP	DPHY Twakeup Timing Parameter Sets the number of clk_esc clock periods to keep a clock or data lane in Mark-1 state after exiting ULPS. The MIPI DPHY spec requires a minimum of 1ms in Mark-1 state after leaving ULPS.

54.7.1.13 Configure Status Out (CFG_STATUS_OUT)

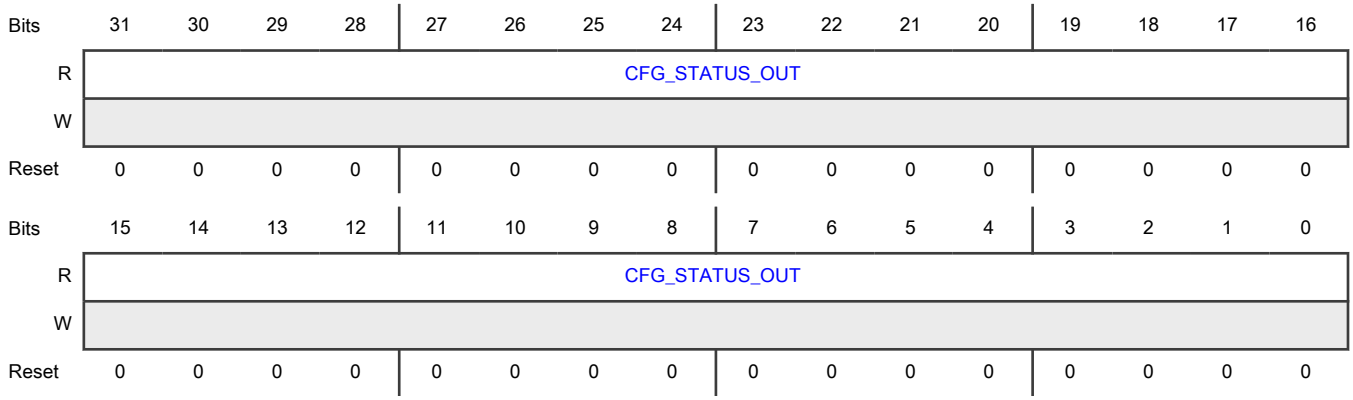
Offset

Register	Offset
CFG_STATUS_OUT	2Ch

Function

Sets status out.

Diagram



Fields

Field	Function
31-0 CFG_STATUS_OUT	Configure Status Out

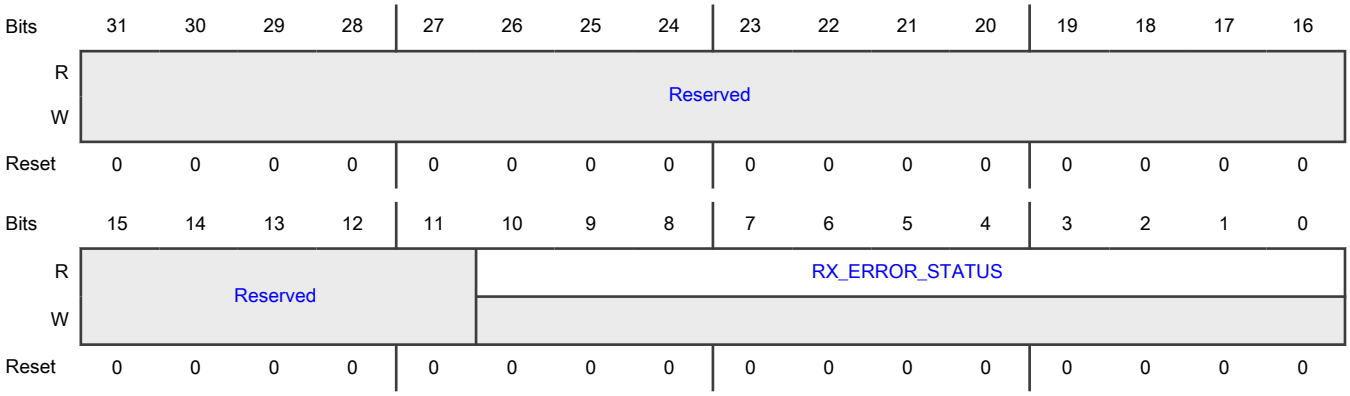
54.7.1.14 Receive Error Status (RX_ERROR_STATUS)

Offset

Register	Offset
RX_ERROR_STATUS	30h

Function
Status for Host receive error detection and timeout indicators.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-0 RX_ERROR_STATUS	Receive Error Status Status for Host receive error detection, ECC errors, CRC errors and for timeout indicators [0] ECC single bit error detected [1] ECC multi bit error detected [6:2] Errored bit position for single bit ECC error [7] CRC error detected

Table continues on the next page...

Table continued from the previous page...

Field	Function
	[8] High Speed forward TX timeout detected [9] Reverse Low power data receive timeout detected [10] BTA timeout detected

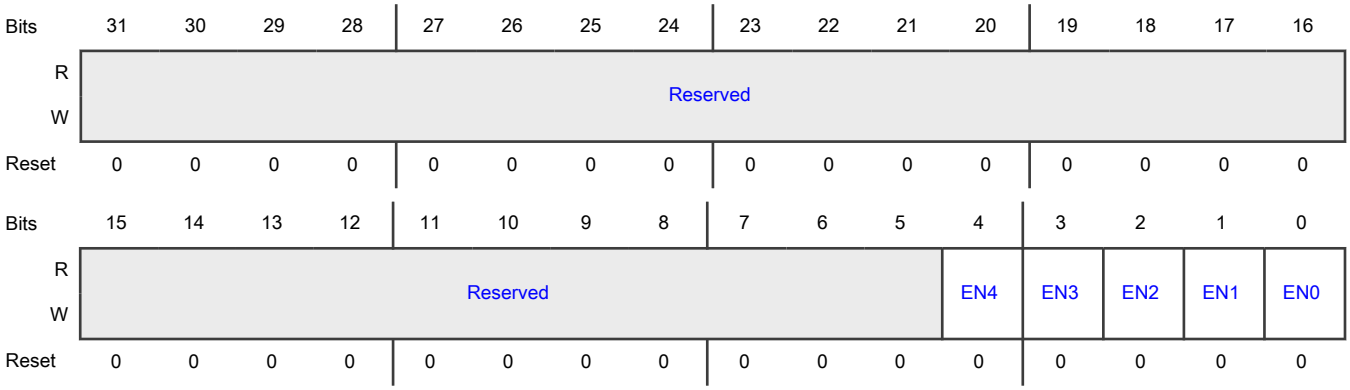
54.7.1.15 ULPS Enable (ULPS_ENABLE)

Offset

Register	Offset
ULPS_ENABLE	34h

Function
Controls the Ultra Low Power State (ULPS) mode.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 EN4	Enable Data Lane 3 Puts data lane 3 into ULPS mode. 0b - Disabled 1b - Enabled
3	Enable Data Lane 2

Table continues on the next page...

Table continued from the previous page...

Field	Function
EN3	Puts data lane 2 into ULPS mode. 0b - Disabled 1b - Enabled
2 EN2	Enable Data Lane 1 Puts data lane 1 into ULPS mode. 0b - Disabled 1b - Enabled
1 EN1	Enable Data Lane 0 Puts data lane 0 into ULPS mode. 0b - Disabled 1b - Enabled
0 EN0	Enable Clock Lane Puts clock lane into ULPS mode. 0b - Disabled 1b - Enabled

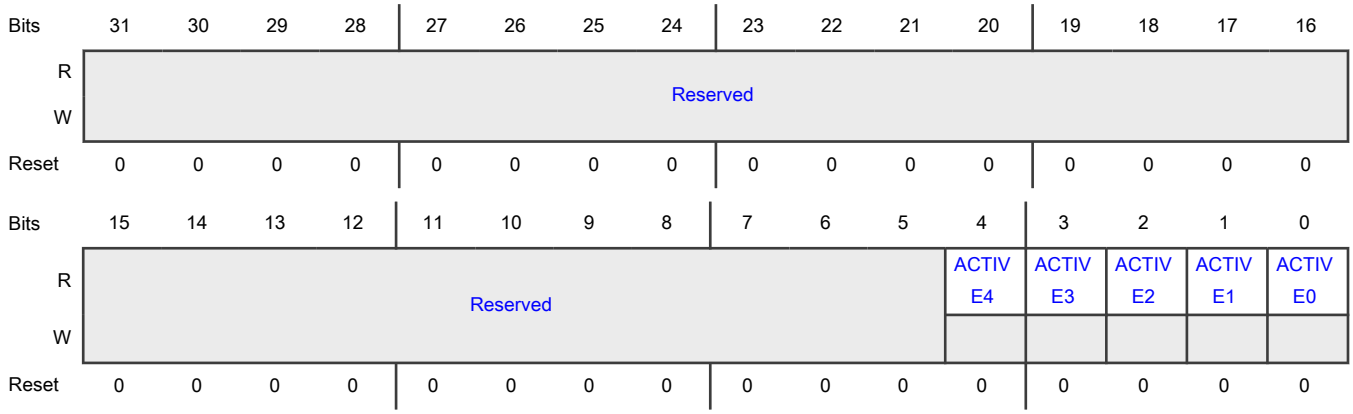
54.7.1.16 ULPS Active (ULPS_ACTIVE)

Offset

Register	Offset
ULPS_ACTIVE	38h

Function
Shows status of Ultra Low Power State (ULPS) mode when active.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 ACTIVE4	Data Lane 3 ULPS ACTIVE Shows data lane 3 is in ULPS mode. 0b - ULPS inactive 1b - ULPS active
3 ACTIVE3	Data Lane 2 ULPS ACTIVE Shows data lane 2 is in ULPS mode. 0b - ULPS inactive 1b - ULPS active
2 ACTIVE2	Data Lane 1 ULPS ACTIVE Shows data lane 1 is in ULPS mode. 0b - ULPS inactive 1b - ULPS active
1 ACTIVE1	Data Lane 0 ULPS ACTIVE Shows data lane 0 is in ULPS mode. 0b - ULPS inactive 1b - ULPS active
0 ACTIVE0	Clock Lane ULPS Active Shows clock lane is in ULPS mode. 0b - ULPS inactive 1b - ULPS active

54.7.1.17 High Speed Mode Enable (HS_MODE_ENABLE)

Offset

Register	Offset
HS_MODE_ENABLE	3Ch

Function

Controls transmission mode.

When the APB CSR is delivered as part of the Host Controller HS_MODE_ENABLE register is logically OR with the controller tx_hs_mode input to the controller.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															ENABLE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 ENABLE	Enable If High Speed (HS) mode is enabled, the APB packet interface always sends packets in HS mode regardless of the setting in Packet Control (PKT_CONTROL). 0b - All transmissions in Low Power Data transmission mode. 1b - All transmissions in High Speed mode.

54.7.1.18 Host Turnaround (HOST_TURNAROUND)

Offset

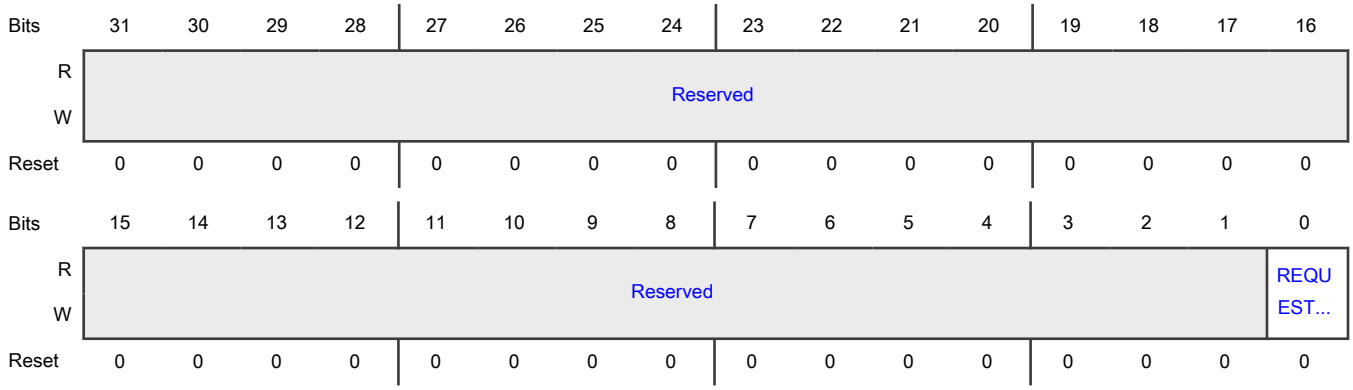
Register	Offset
HOST_TURNAROUND	40h

Function

Controls Bus Turn Around (BTA).

When the APB CSR is delivered as part of the Host Controller, HOST TURNAROUND register is logically OR with the controller dphy_turnaround input to the controller.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 REQUEST_BTA	Request BTA 0b - No Request 1b - Request BTA

54.7.1.19 PHY Direction (PHY_DIRECTION)

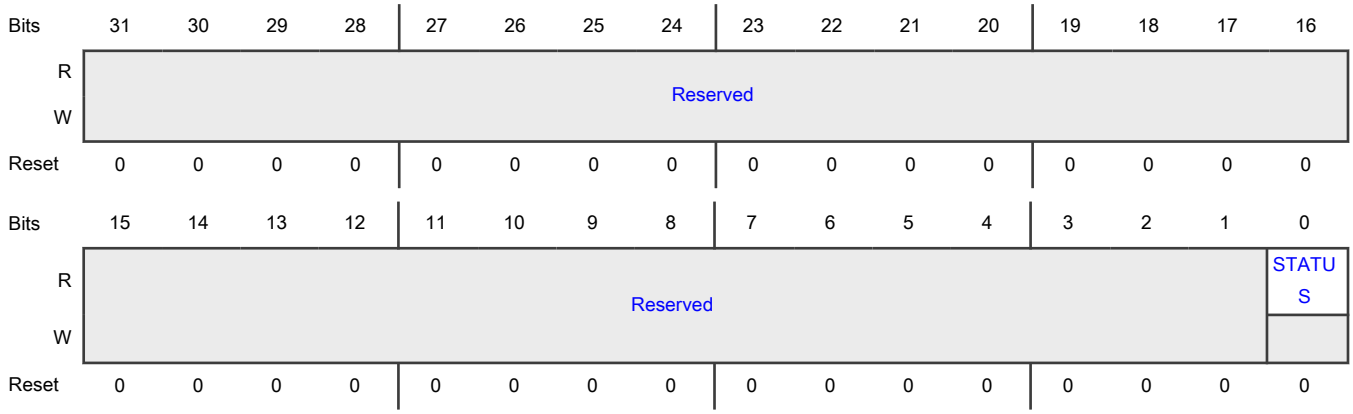
Offset

Register	Offset
PHY_DIRECTION	44h

Function

Shows PHY Status.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 STATUS	PHY Direction 0b - Default PHY direction 1b - Reversed PHY direction

54.7.1.20 PHY Ready (PHY_RDY)

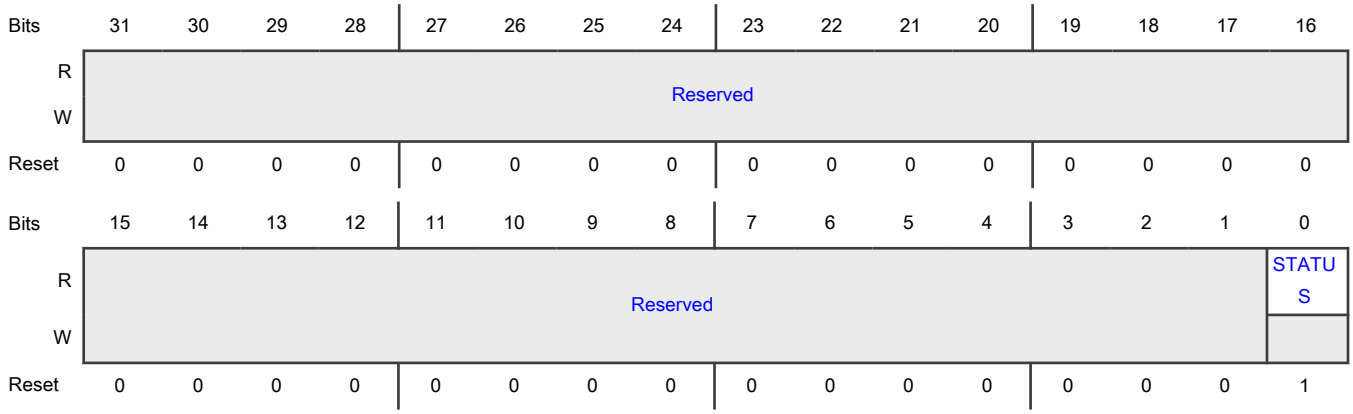
Offset

Register	Offset
PHY_RDY	48h

Function

Indicates Host DPHY ready status.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 STATUS	PHY Ready Status 0b - DPHY not ready 1b - DPHY ready

54.7.1.21 Pixel Payload Size (CFG_DBI_PIXEL_PAYLOAD_SIZE)

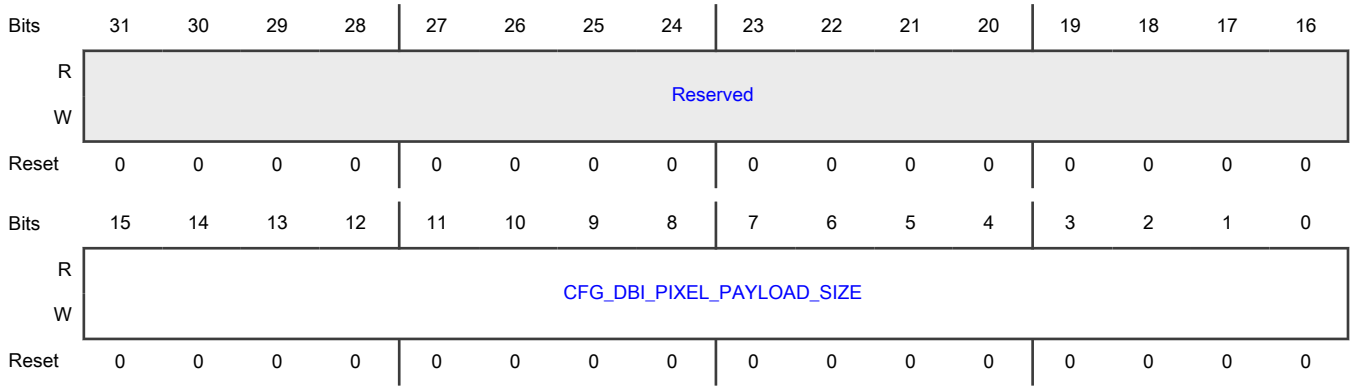
Offset

Register	Offset
CFG_DBI_PIXEL_PAYLOAD_SIZE	100h

Function

Maximum number of pixels that should be sent as one DSI packet.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DBI_PIXEL_PAYLOAD_SIZE	Pixel Payload Size Recommended to be evenly divisible by the line size (in pixels).

54.7.1.22 Configure DBI Pixel FIFO Send Level (CFG_DBI_PIXEL_FIFO_SEND_LEVEL)

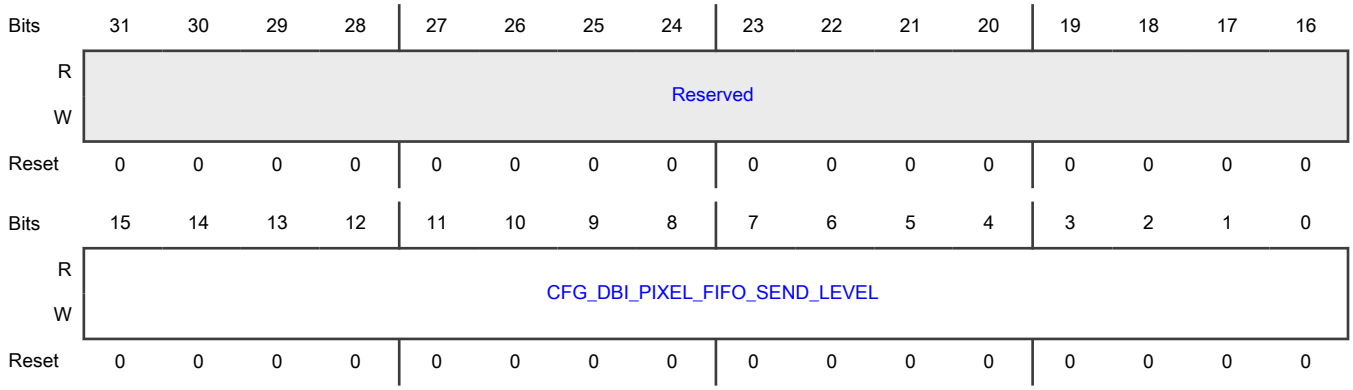
Offset

Register	Offset
CFG_DBI_PIXEL_FIFO_SEND_LEVEL	104h

Function

Controls the level at which the DBI Host bridge begins sending pixels.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DBI_PIXEL_FIFO_SEND_LEVEL	Configure DBI Pixel FIFO Send Level In order to optimize DSI utility, the DBI bridge buffers a certain number of DBI pixels before initiating a DSI packet.

54.7.1.23 DBI Pixel Format (CFG_DBI_PIXEL_FORMAT)

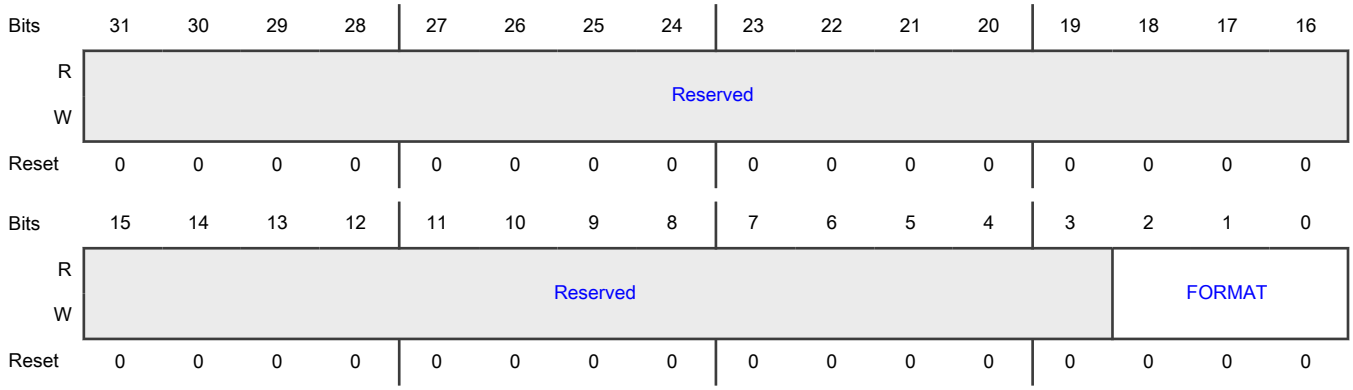
Offset

Register	Offset
CFG_DBI_PIXEL_FORMAT	108h

Function

Configures DBI Pixel Format.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 FORMAT	DBI Pixel Format Options **NOTE** For Option 4: RGB444, both display controller and DSI should work in 8-bit mode, and other configurations should work in 16-bit mode. 000b - Default. No pixel to byte mapping 001b - Option 1: RGB888 010b - Option 2: RGB666 011b - Option 3: RGB565 100b - Option 4: RGB444 101b - Option 5: RGB332 110b-111b - Reserved

54.7.1.24 DBI Underrun Error (DBI_UNDERRUN_ERR)

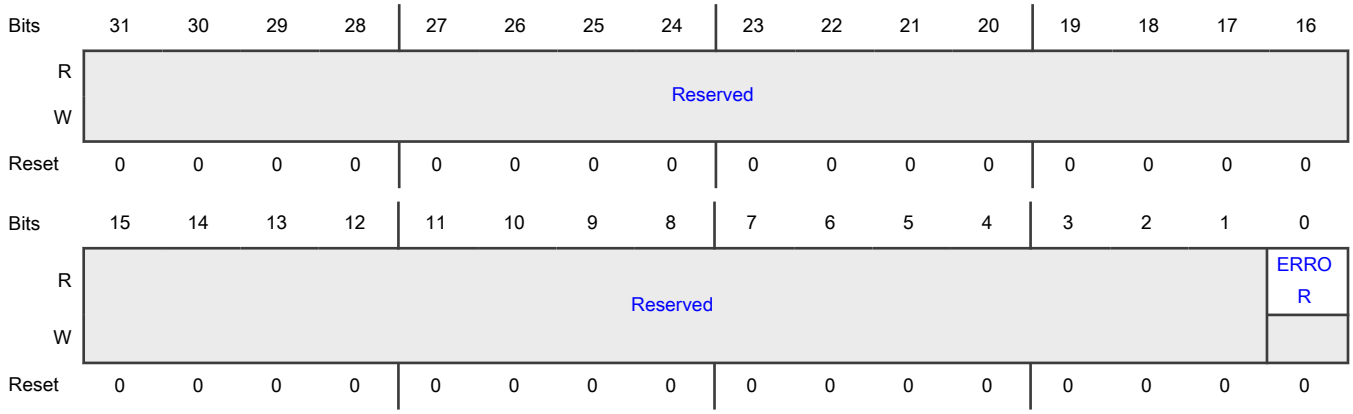
Offset

Register	Offset
DBI_UNDERRUN_ERR	10Ch

Function

Indicates host DBI FIFO underrun error.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 ERROR	Error 0b - No Error 1b - Error

54.7.1.25 DBI Overflow Error (DBI_OVERFLOW_ERR)

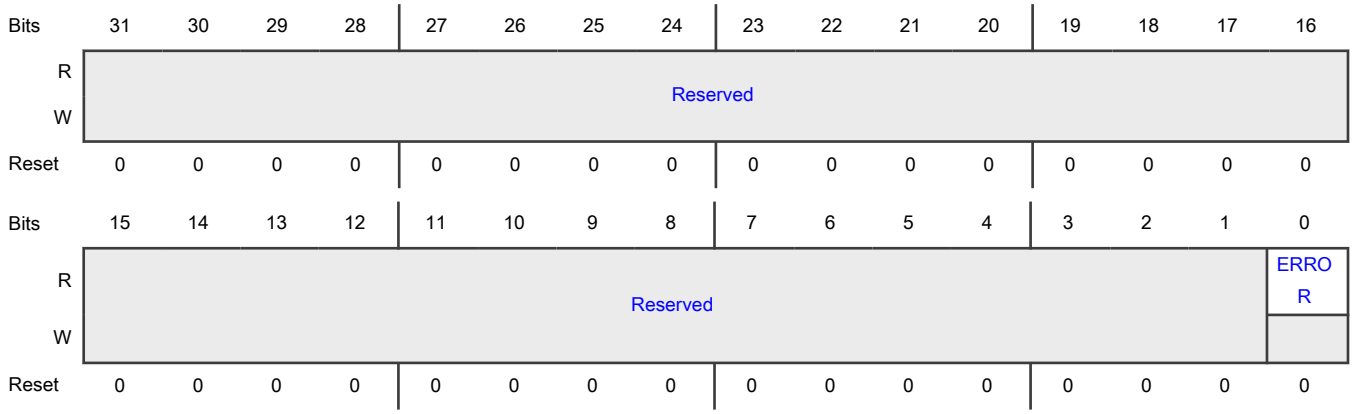
Offset

Register	Offset
DBI_OVERFLOW_ERR	110h

Function

Indicates host DBI FIFO overflow error.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 ERROR	Error 0b - No Error 1b - Error

54.7.1.26 Configure DPI Pixel Payload Size (CFG_DPI_PIXEL_PAYLOAD_SIZE)

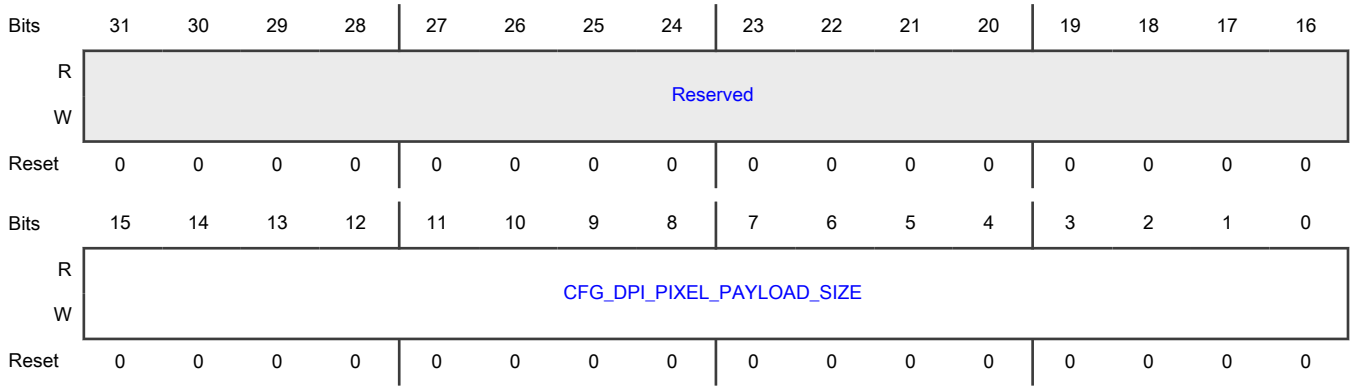
Offset

Register	Offset
CFG_DPI_PIXEL_PAYLOAD_SIZE	200h

Function

Maximum number of pixels that should be sent as one DSI packet.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DPI_PIXEL_PAYLOAD_SIZE	Configure DPI Pixel Payload Recommended to be evenly divisible by the line size (in pixels).

54.7.1.27 Configure DPI Pixel FIFO Send Level (CFG_DPI_PIXEL_FIFO_SEND_LEVEL)

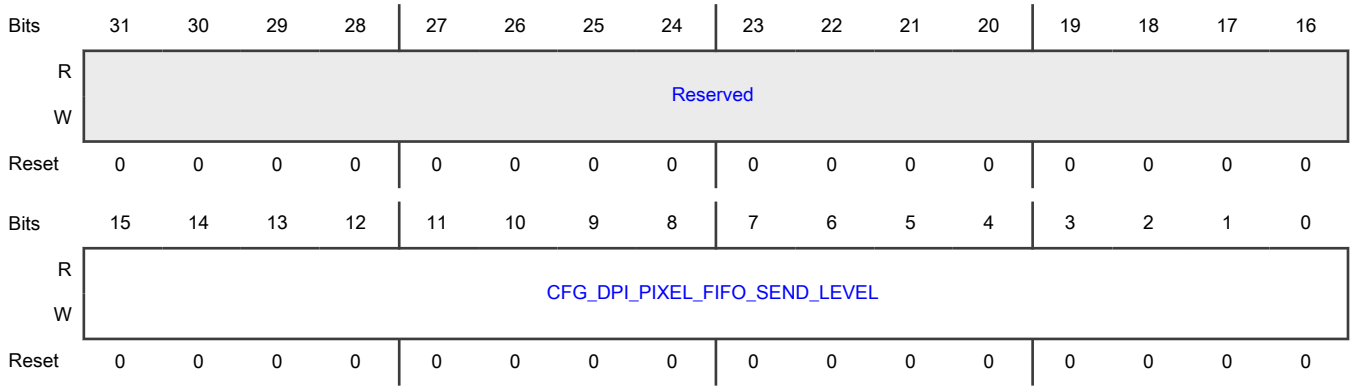
Offset

Register	Offset
CFG_DPI_PIXEL_FIFO_SEND_LEVEL	204h

Function

Controls the level at which the DPI Host bridge begins sending pixels.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DPI_PIXEL_FIFO_SEND_LEVEL	Configure DPI Pixel FIFO Send Level In order to optimize DSI utility, the DPI bridge buffers a certain number of DPI pixels before initiating a DSI packet.

54.7.1.28 Configure DPI Interface Color Coding (CFG_DPI_INTERFACE_COLOR_CODING)

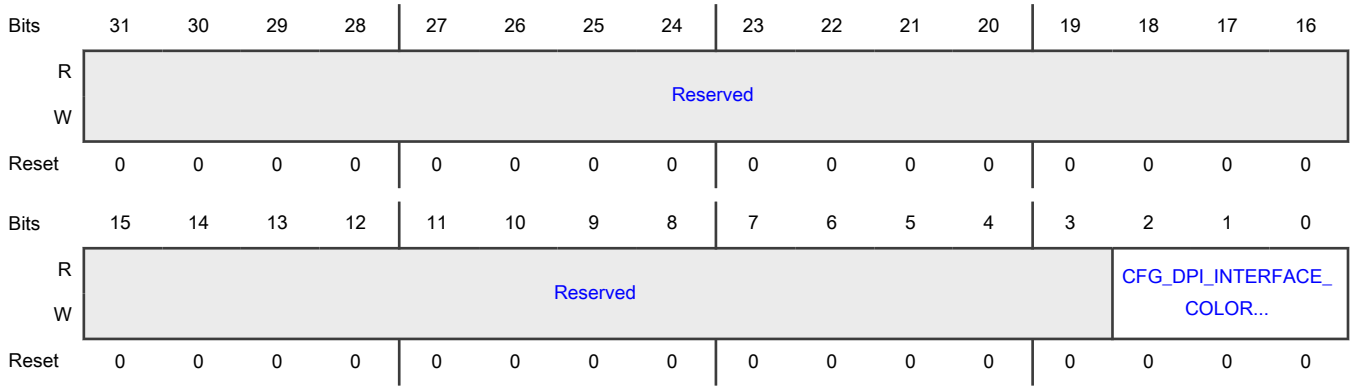
Offset

Register	Offset
CFG_DPI_INTERFACE_COLOR_CODING	208h

Function

Sets the distribution of RGB bits.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 CFG_DPI_INTERFACE_COLOR_CODING	Configure DPI Interface Color Coding Sets the distribution of RGB bits within the 24-bit d bus, as specified by the DPI specification. 0= 16-bit Configuration 1 1= 16-bit Configuration 2 2= 16-bit Configuration 3 3= 18-bit Configuration 1 4= 18-bit Configuration 2 5= 24-bit

54.7.1.29 Configure DPI Pixel Format (CFG_DPI_PIXEL_FORMAT)

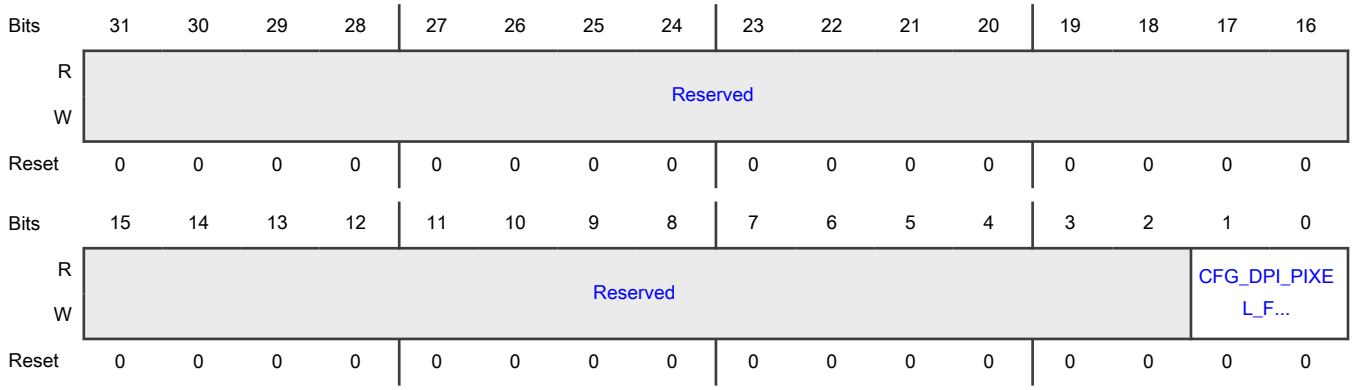
Offset

Register	Offset
CFG_DPI_PIXEL_FORMAT	20Ch

Function

Sets the DSI packet type of the pixels.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 CFG_DPI_PIXEL_FORMAT	Configure DPI Pixel Format Sets the DSI packet type of the pixels. 0 - 16 bit 1 - 18 bit 2 - 18 bit loosely packed 3 - 24 bit

54.7.1.30 Configure DPI vsync Polarity (CFG_DPI_VSYNC_POLARITY)

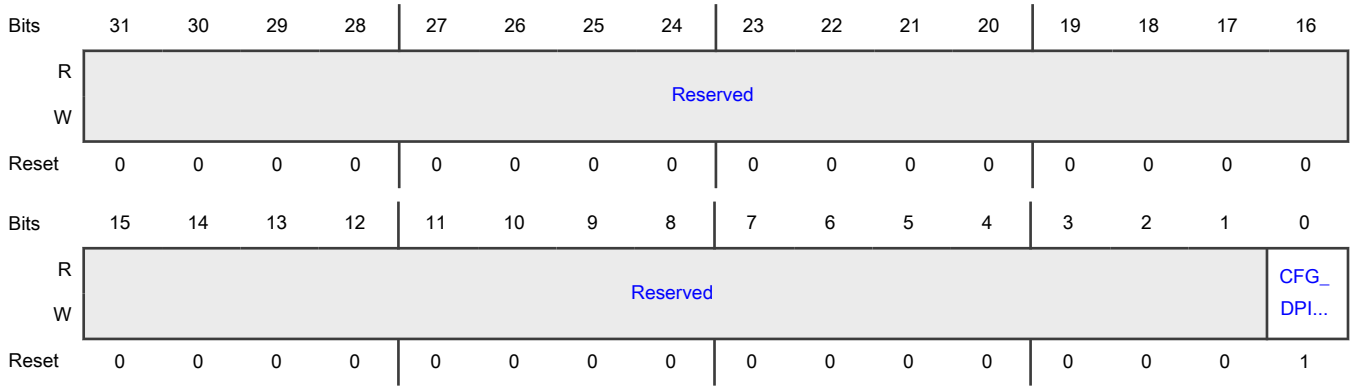
Offset

Register	Offset
CFG_DPI_VSYNC_POLARITY	210h

Function

Sets polarity of dpi_vsync_input

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_DPI_VSYNC_POLARITY	Configure DPI vsync Polarity Sets polarity of dpi_vsync_input. 0 - active low 1 - active high

54.7.1.31 Configure DPI hsync Polarity (CFG_DPI_HSYNC_POLARITY)

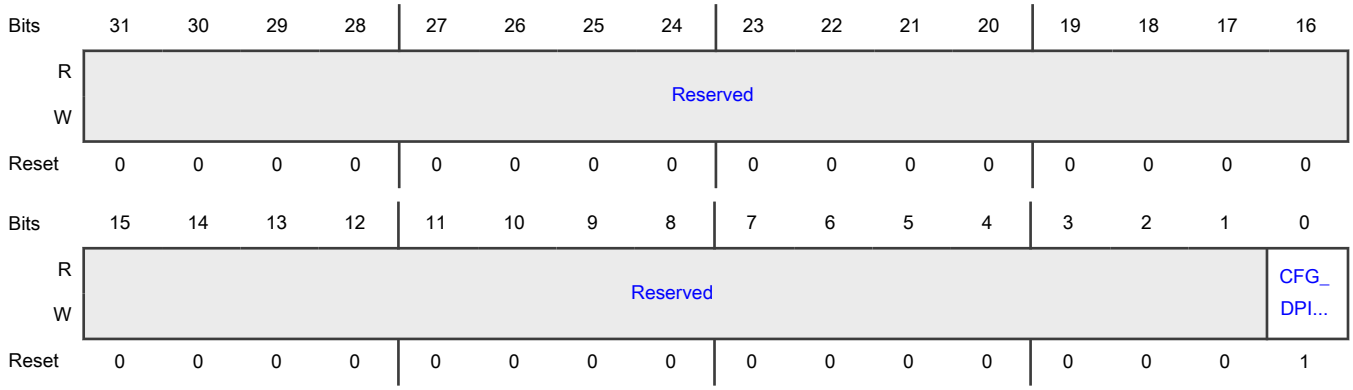
Offset

Register	Offset
CFG_DPI_HSYNC_POLARITY	214h

Function

Sets polarity of dpi_hsync_input.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_DPI_HSYNC_POLARITY	Configure DPI hsync Polarity Sets polarity of dpi_hsync_input. 0 - active low 1 - active high

54.7.1.32 Configure DPI Video Mode (CFG_DPI_VIDEO_MODE)

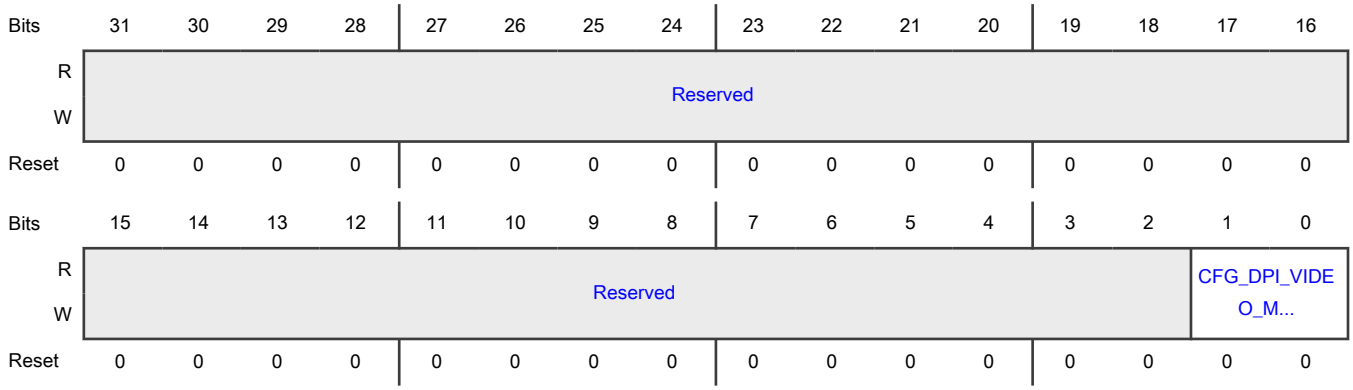
Offset

Register	Offset
CFG_DPI_VIDEO_MODE	218h

Function

Selects DSI video mode for the host DPI module generated packets.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 CFG_DPI_VIDEO_MODE	Configure DPI Video Mode Selects DSI video mode for the host DPI module generated packets. 00b - Non-Burst mode with Sync Pulses 01b - Non-Burst mode with Sync Events 10b - Burst mode 11b - Reserved, not valid

54.7.1.33 Configure DPI horizontal front porch (CFG_DPI_HFP)

Offset

Register	Offset
CFG_DPI_HFP	21Ch

Function

Sets the DSI packet payload size for the HFP blanking packet.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_DPI_HFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DPI_HFP	Configure DPI Horizontal Front Porch Sets the DSI packet payload size, in bytes, of the horizontal front porch blanking packet.

54.7.1.34 Configure DPI Horizontal Back Porch (CFG_DPI_HBP)

Offset

Register	Offset
CFG_DPI_HBP	220h

Function

Sets the DSI packet payload size of the HBP blanking packet.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_DPI_HBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DPI_HBP	Configure DPI Horizontal Back Porch Sets the DSI packet payload size, in bytes, of the horizontal back porch blanking packet.

54.7.1.35 Configure DPI Horizontal Sync Width (CFG_DPI_HSA)

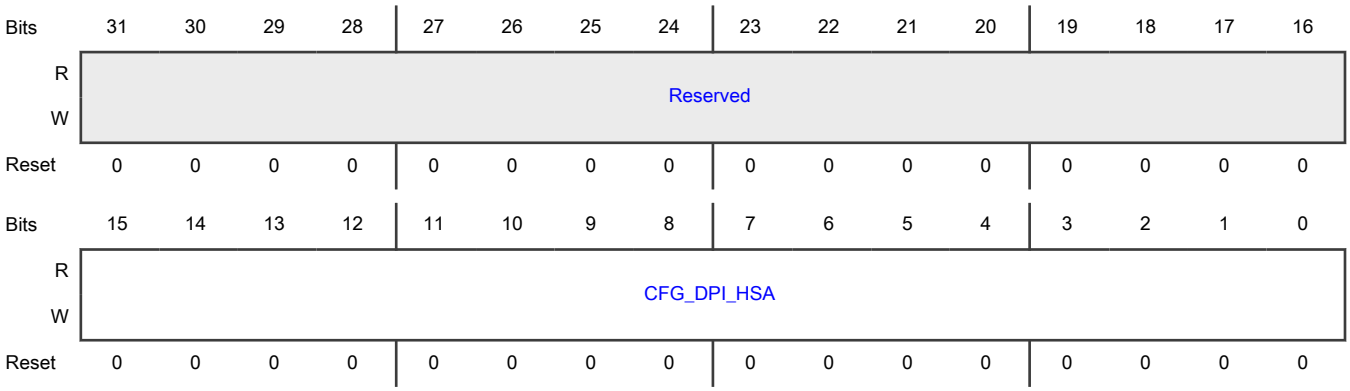
Offset

Register	Offset
CFG_DPI_HSA	224h

Function

Sets the DSI packet payload size of the HSA blanking packet.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CFG_DPI_HSA	Configure DPI Horizontal Sync Width Sets the DSI packet payload size, in bytes, of the horizontal sync width filler blanking packet.

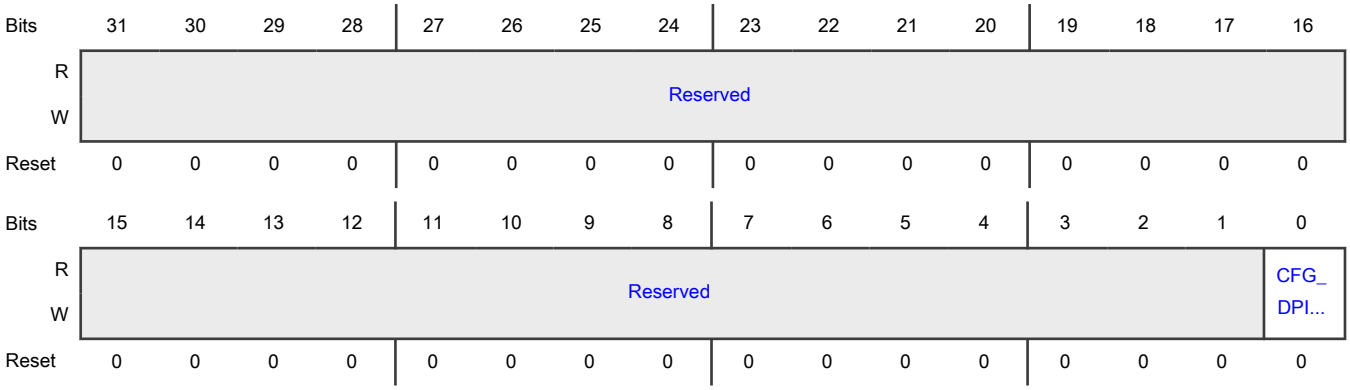
54.7.1.36 Enable Multiple Packets (CFG_DPI_ENABLE_MULT_PKTS)

Offset

Register	Offset
CFG_DPI_ENABLE_MULT_PKTS	228h

Function
Enables video line sent in two packets.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_DPI_ENABLE_MULT_PKTS	Enable Multiple Packets Per Video Line. When enabled, cfg_dpi_pixel_payload_size must be set to exactly half the size of the video line. 0 - Video Line is sent in a single packet 1 - Video Line is sent in two packets

54.7.1.37 Configure DPI Vertical Back Porch (CFG_DPI_VBP)

Offset

Register	Offset
CFG_DPI_VBP	22Ch

Function
Sets the number of lines in the vertical back porch.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				CFG_DPI_VBP											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11-0 CFG_DPI_VBP	Configure DPI Vertical Back Porch

54.7.1.38 Configure DPI Vertical Front Porch (CFG_DPI_VFP)

Offset

Register	Offset
CFG_DPI_VFP	230h

Function
Sets the number of lines in the vertical front porch.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				CFG_DPI_VFP											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11-0 CFG_DPI_VFP	Configure DPI VPP

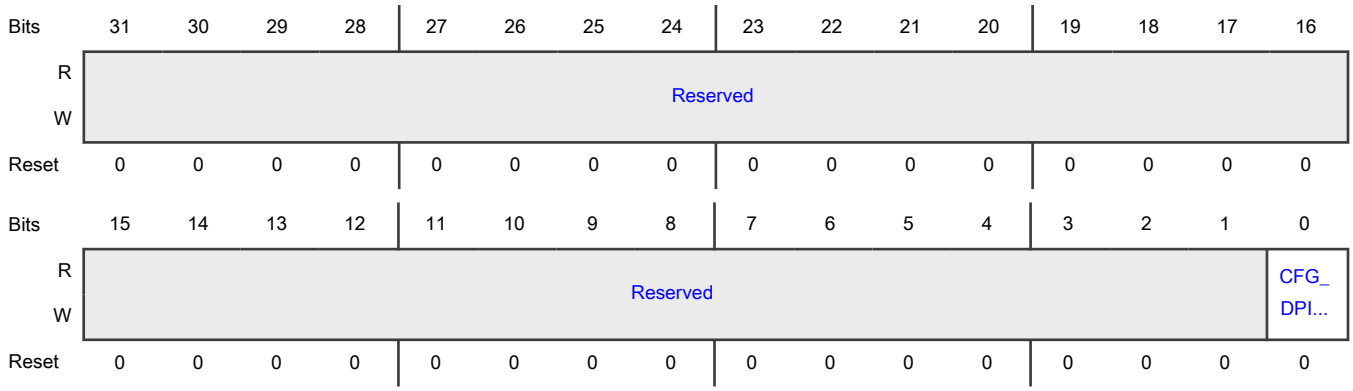
54.7.1.39 Configure DPI BLLP Mode (CFG_DPI_BLLP_MODE)

Offset

Register	Offset
CFG_DPI_BLLP_MODE	234h

Function
Used to optimize bllp periods to Low Power when possible.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_DPI_BLLP_MODE	Configure DPI BLLP Mode Optimize bllp periods to Low Power mode when possible 0 - blanking packets are sent during BLLP periods 1 - LP mode is used for BLLP periods

54.7.1.40 Configure DPI Blank Packet in BLLP (CFG_DPI_USE_NULL_PKT_BLLP)

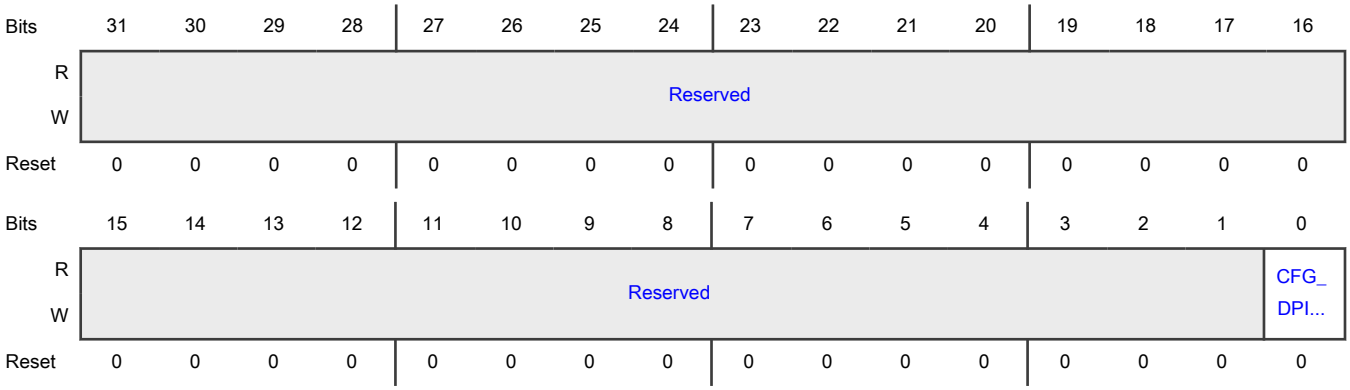
Offset

Register	Offset
CFG_DPI_USE_NULL_PKT_BLLP	238h

Function

Selects type of blanking packet to be sent during bllp region.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 CFG_DPI_USE_NULL_PKT_BLLP	Configure DPI Blank Packet in BLLP Selects type of blanking packet to be sent during bllp region. 0 - Blanking packet used in bllp region 1 - Null packet used in bllp region

54.7.1.41 Configure DPI Vertical Active (CFG_DPI_VACTIVE)

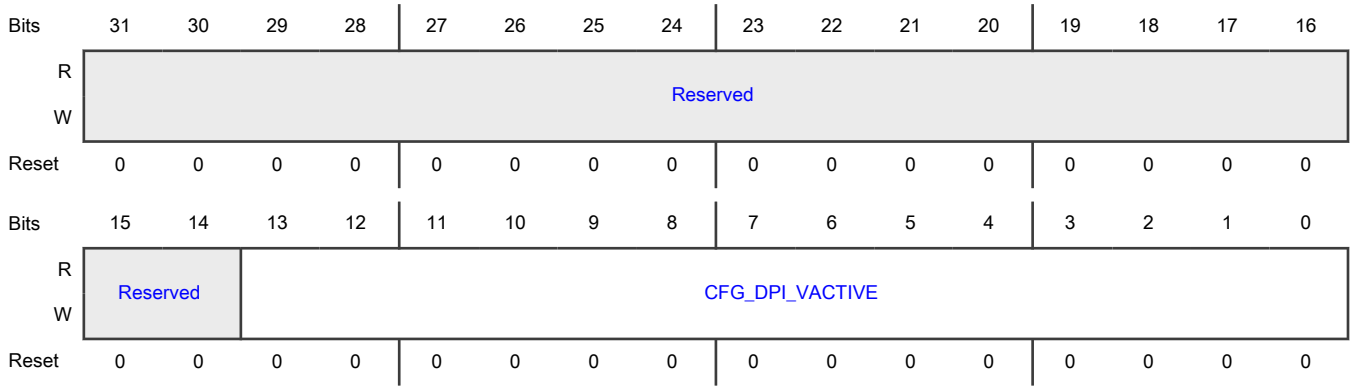
Offset

Register	Offset
CFG_DPI_VACTIVE	23Ch

Function

Sets the number of lines in the Vertical Active.

Diagram



Fields

Field	Function
31-14 —	Reserved
13-0 CFG_DPI_VACTIVE	Configure DPI Vertical Active Sets the number of lines in the vertical active read.

54.7.1.42 Configure DPI Virtual Channel (CFG_DPI_VC)

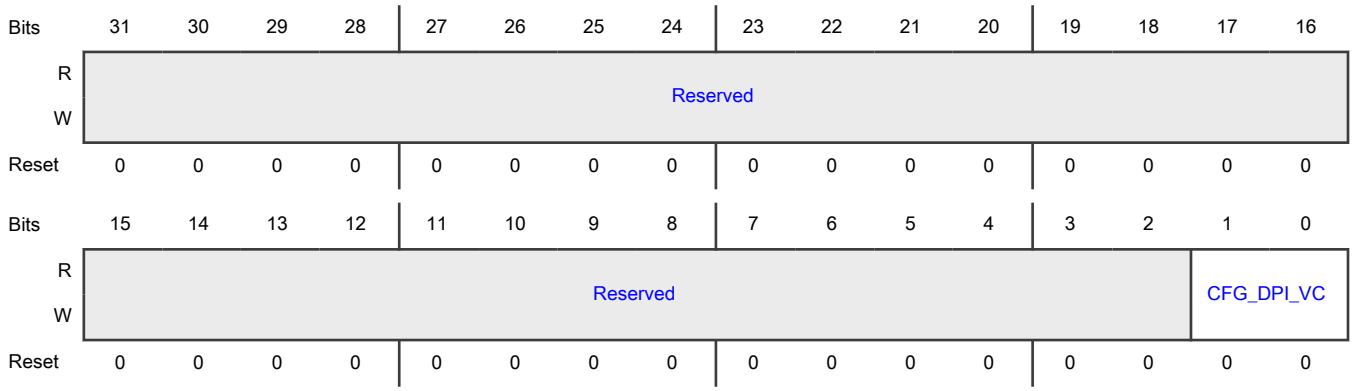
Offset

Register	Offset
CFG_DPI_VC	240h

Function

Sets the Virtual Channel (VC) of packets that will be sent to the receive packet interface.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 CFG_DPI_VC	Configure DPI Virtual Channel Packets with VC not equal to this value are discarded and the "DSI VC ID Invalid" bit (bit 12) in the DSI error report is set.

54.7.1.43 DPI Underrun Error (DPI_UNDERRUN_ERR)

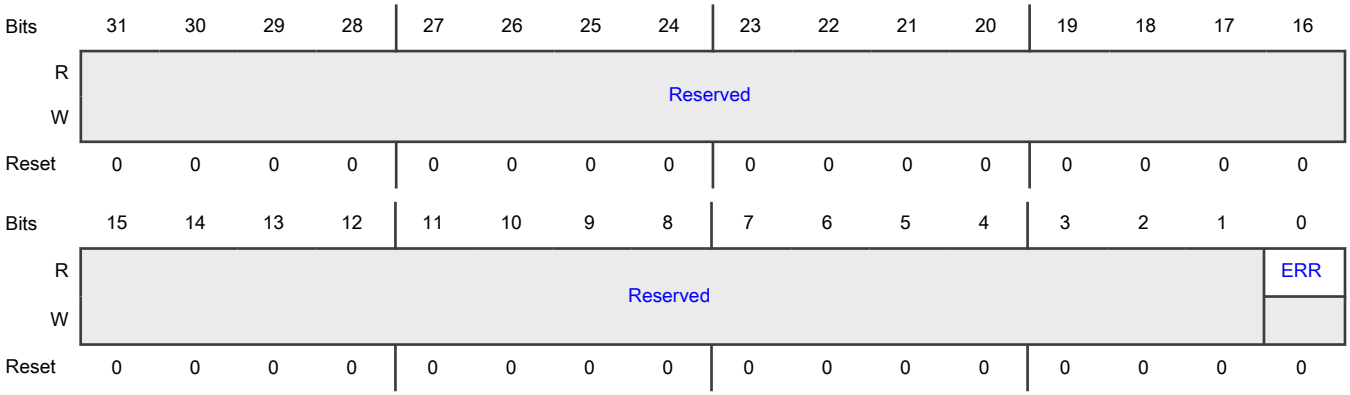
Offset

Register	Offset
DPI_UNDERRUN_ERR	244h

Function

Indicates Host DPI FIFO underrun error detected.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 ERR	Error detected 0b - No error 1b - Error

54.7.1.44 Transmit Payload (TX_PAYLOAD)

Offset

Register	Offset
TX_PAYLOAD	280h

Function

Writes to this registers load the payload fifo with 32 bit values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TX_PAYLOAD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TX_PAYLOAD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TX_PAYLOAD	Tx Payload Data Write

54.7.1.45 Packet Control (PKT_CONTROL)

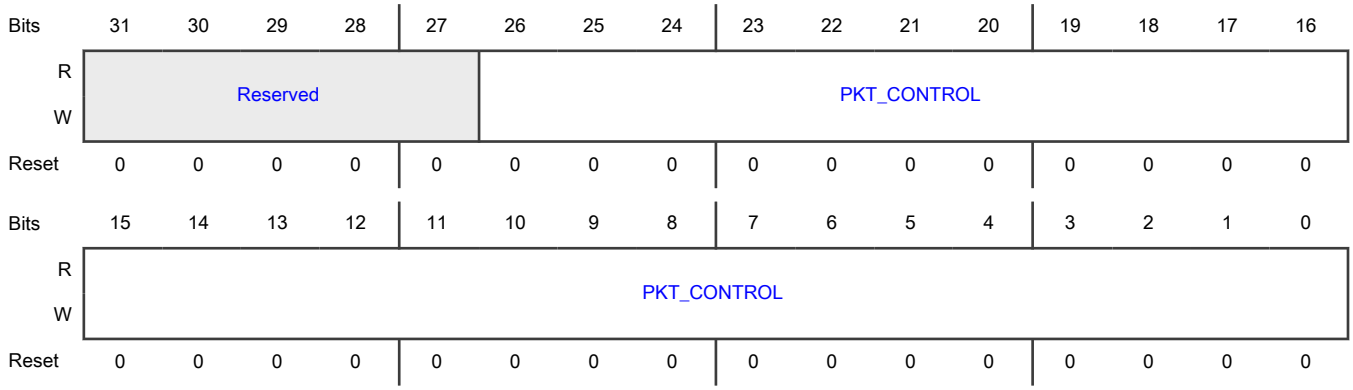
Offset

Register	Offset
PKT_CONTROL	284h

Function

Sets values for packet control.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-0 PKT_CONTROL	Tx Packet Control [15:0] - Packet word count [17:16] - Packet Virtual Channel [23:18] - Packet Header DSI Data Type [24] - LP or HS select. 0 - LP mode, 1 - HS mode [25] - Perform BTA after packet is sent [26] - Perform BTA only, no packet tx

54.7.1.46 Send Packet (SEND_PACKET)

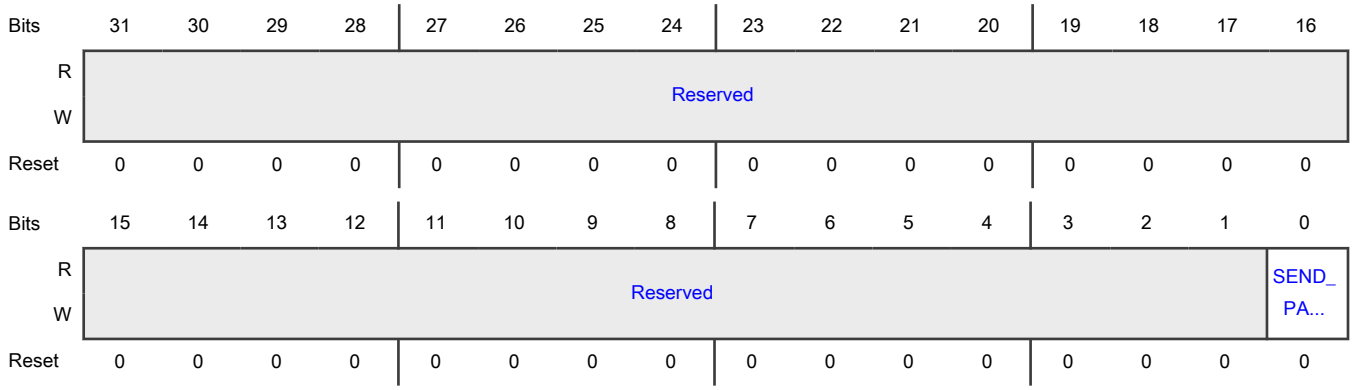
Offset

Register	Offset
SEND_PACKET	288h

Function

Tx send packet, writing to this register causes the packet described in dsi_host_pkt_control to be sent.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 SEND_PACKET	Send Packet

54.7.1.47 Packet Status (PKT_STATUS)

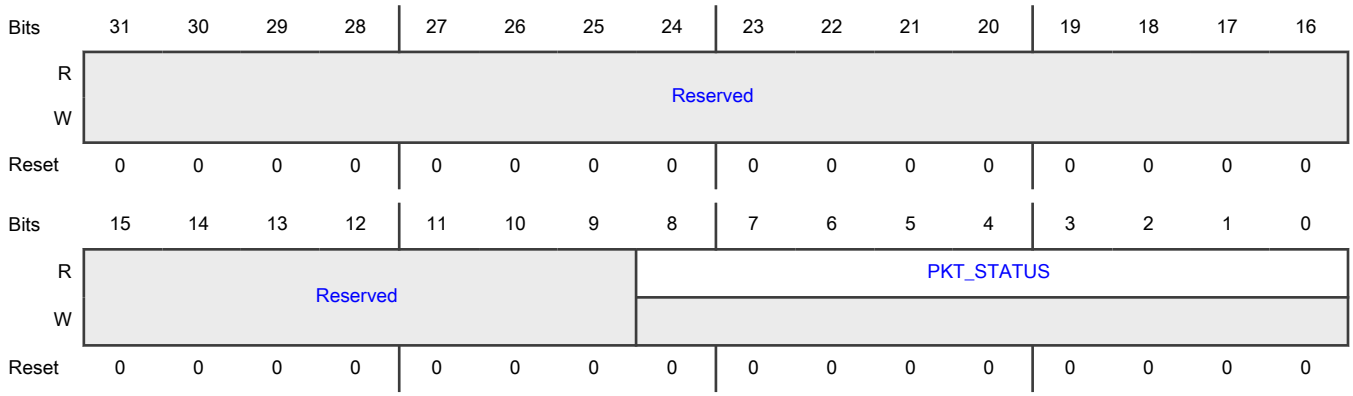
Offset

Register	Offset
PKT_STATUS	28Ch

Function

Status of APB to packet interface

Diagram



Fields

Field	Function
31-9 —	Reserved
8-0 PKT_STATUS	Packet Status [0] - state machine not idle [1] - Tx packet done [2] - dphy direction 0 - tx had control, 1 - rx has control [3] - tx fifo overflow [4] - tx fifo underflow [5] - rx fifo overflow [6] - rx fifo underflow [7] - rx packet header has been received [8] - all rx packet payload data has been received

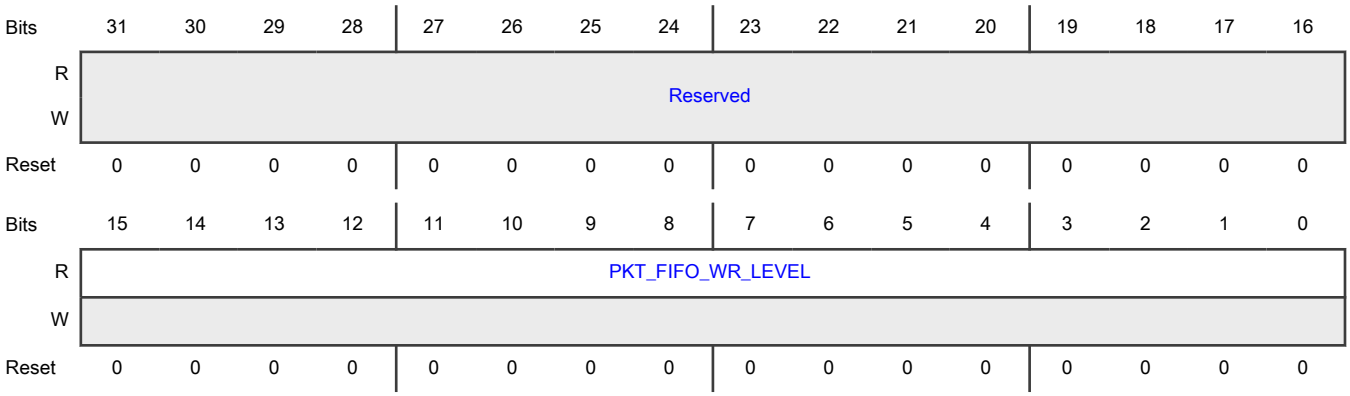
54.7.1.48 Packet FIFO Write (PKT_FIFO_WR_LEVEL)

Offset

Register	Offset
PKT_FIFO_WR_LEVEL	290h

Function
Write level of APB to pkt interface fifo.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PKT_FIFO_WR_LEVEL	Packet FIFO Write Level

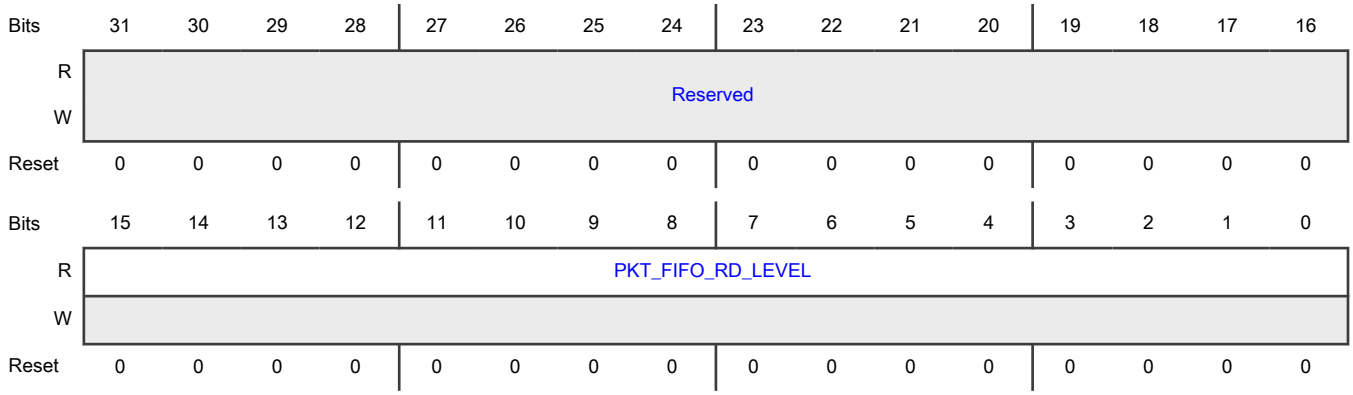
54.7.1.49 Packet FIFO Read (PKT_FIFO_RD_LEVEL)

Offset

Register	Offset
PKT_FIFO_RD_LEVEL	294h

Function
Read level of APB to pkt interface fifo.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PKT_FIFO_RD_LEVEL	Packet FIFO Read Level

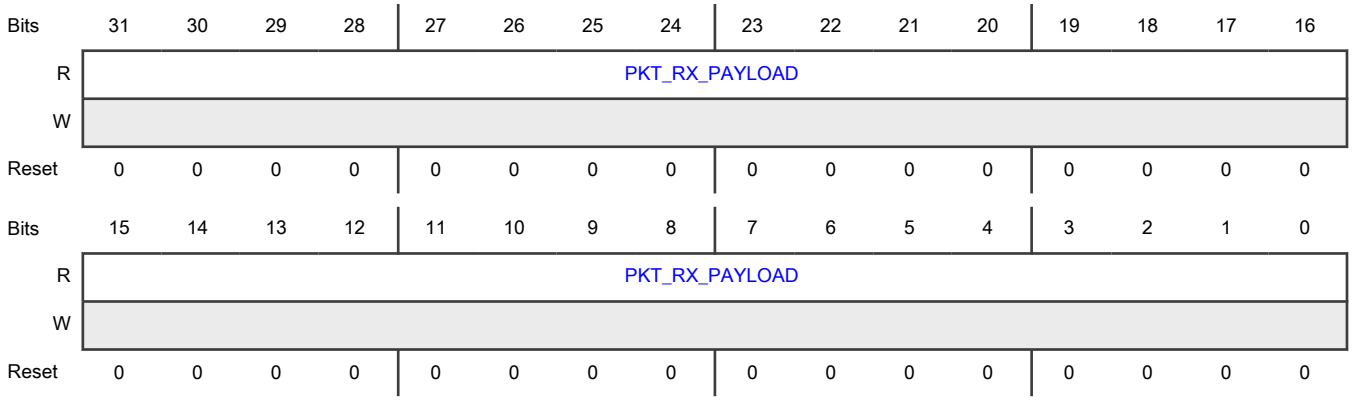
54.7.1.50 Packet Rx Payload (PKT_RX_PAYLOAD)

Offset

Register	Offset
PKT_RX_PAYLOAD	298h

Function
APB to pkt interface rx payload read.

Diagram



Fields

Field	Function
31-0 PKT_RX_PAYLOAD	Packet Rx Payload

54.7.1.51 Packet Rx Packet Header (PKT_RX_PKT_HEADER)

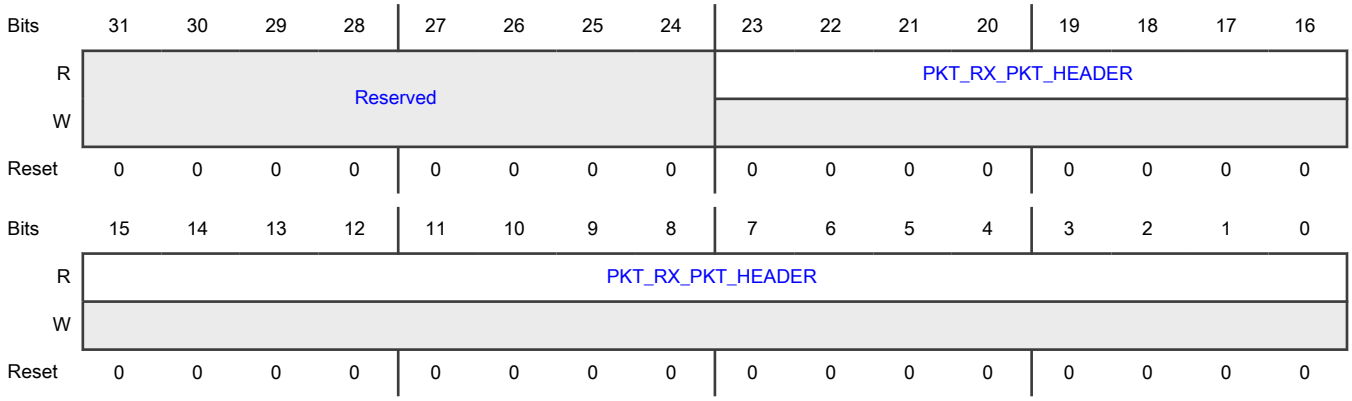
Offset

Register	Offset
PKT_RX_PKT_HEADER	29Ch

Function

APB to pkt interface rx packet header

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 PKT_RX_PKT_HEADER	Packet Rx Packet Header APB to pkt interface rx packet header [15:0] word count [21:16] data type [23:22] Virtual Channel

54.7.1.52 Interrupt Status (IRQ_STATUS)

Offset

Register	Offset
IRQ_STATUS	2A0h

Function

Status of APB to packet interface

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 IRQ_STATUS	Interrupt Status Status of APB to packet interface [0] - state machine not idle [1] - Tx packet done [2] - dphy direction 0 - tx had control, 1 - rx has control [3] - tx fifo overflow [4] - tx fifo underflow [5] - rx fifo overflow [6] - rx fifo underflow [7] - rx packet header has been received [8] - all rx packet payload data has been received [28:9] - map directory to dsi host controller status_out port bit descriptions [29] - host bta timeout, host controller host_bta_timeout port [30] - low power rx timeout, host controller lp_rx_timeout port [31] - high speed tx timeout, host controller hs_tx_timeout port

54.7.1.53 Interrupt Status 2 (IRQ_STATUS2)

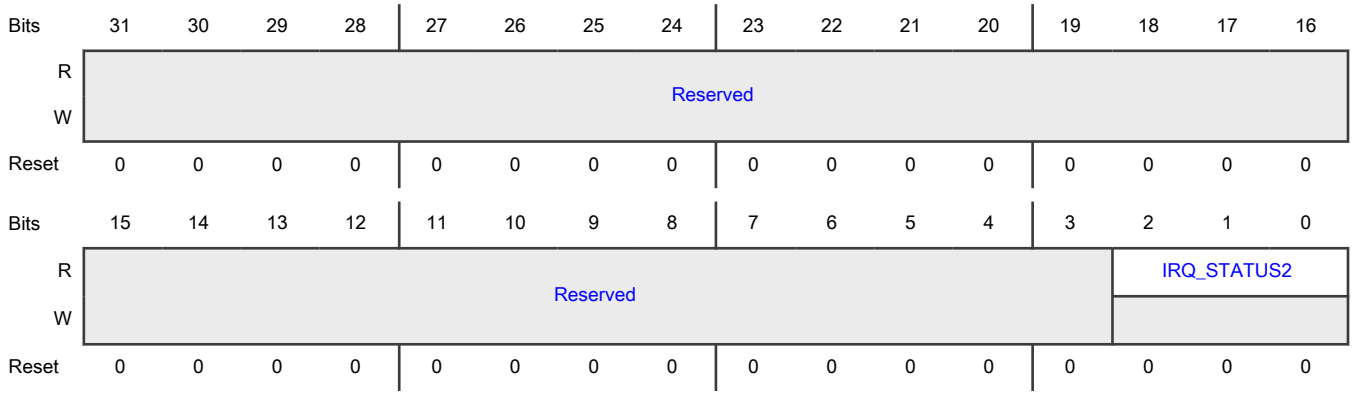
Offset

Register	Offset
IRQ_STATUS2	2A4h

Function

Status of APB to packet interface part 2, read part 2 first then dsi_host_irq_status.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 IRQ_STATUS2	Interrupt Status 2 Reading dsi_host_irq_status will clear both status and status2 [0] - single bit ecc error [1] - multi bit ecc error [2] - crc error

54.7.1.54 Mask Interrupt (IRQ_MASK)

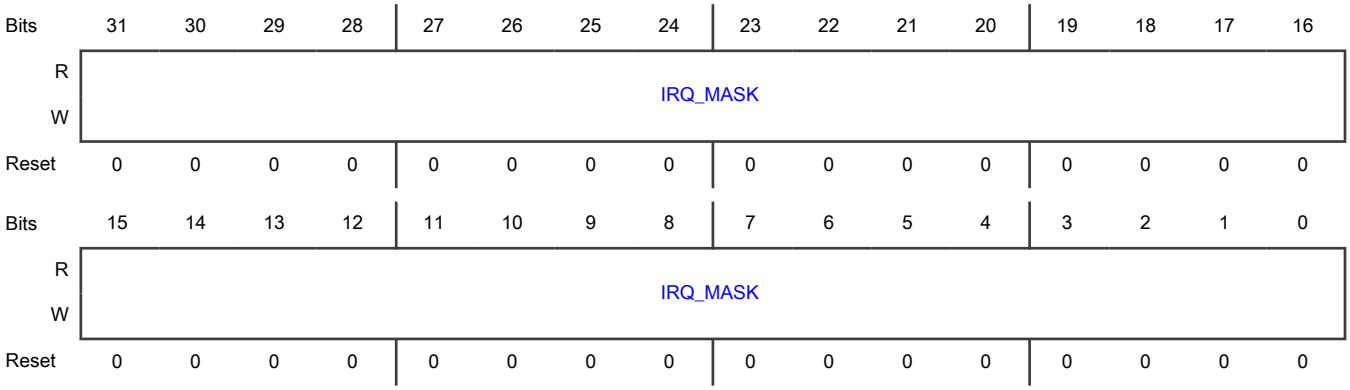
Offset

Register	Offset
IRQ_MASK	2A8h

Function

Contains various mask interrupts.

Diagram



Fields

Field	Function
31-0 IRQ_MASK	Mask Interrupt irq mask [0] - state machine not idle [1] - Tx packet done [2] - dphy direction 0 - tx had control, 1 - rx has control [3] - tx fifo overflow [4] - tx fifo underflow [5] - rx fifo overflow [6] - rx fifo underflow [7] - rx packet header has been received [8] - all rx packet payload data has been received [28:9] - map directory to dsi host controller status_out port bit descriptions [29] - host bta timeout, host controller host_bta_timeout port [30] - low power rx timeout, host controller lp_rx_timeout port [31] - high speed tx timeout, host controller hs_tx_timeout port

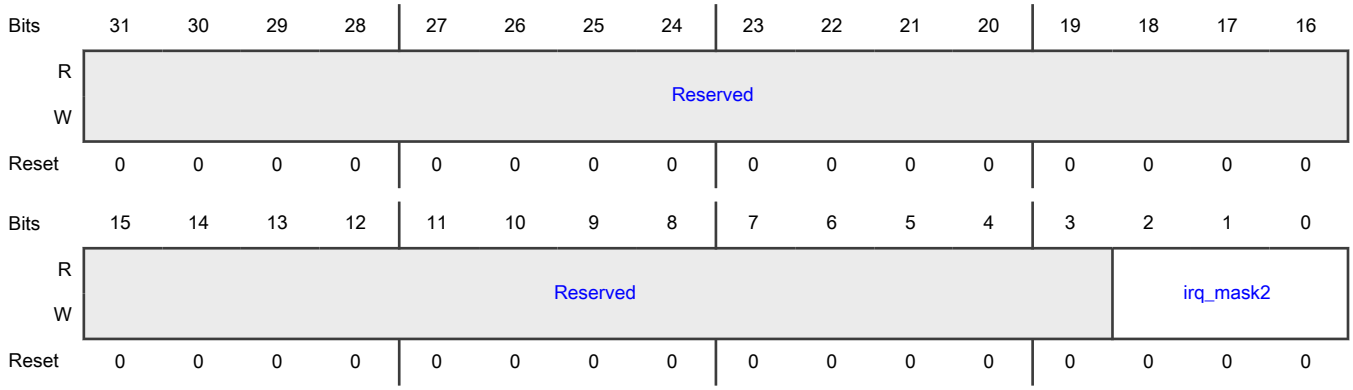
54.7.1.55 Interrupt Mask 2 (IRQ_MASK2)

Offset

Register	Offset
IRQ_MASK2	2ACh

Function
Contains various mask interrupts.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 irq_mask2	Interrupt Mask 2 irq mask 2 [0] - single bit ecc error [1] - multi bit ecc error [2] - crc error

54.7.1.56 DPHY Power Down (PD_DPHY)

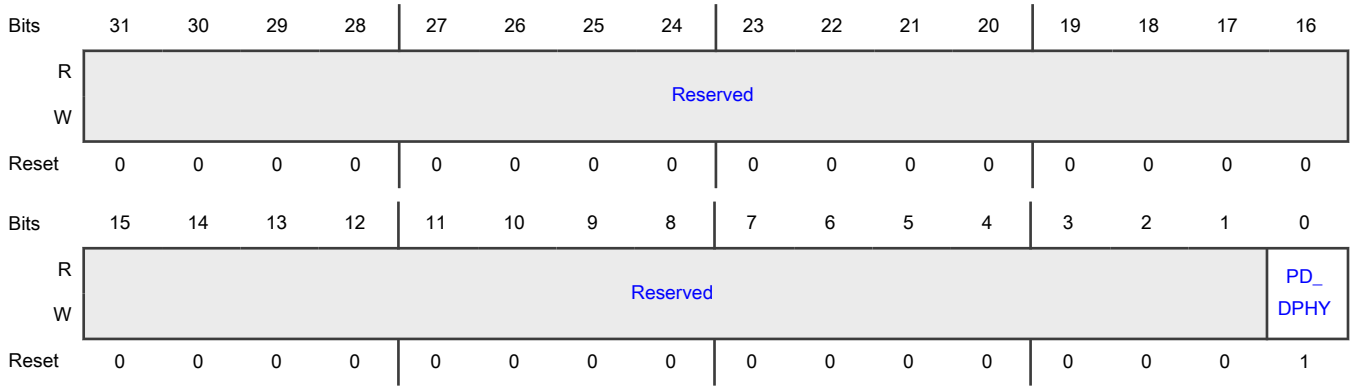
Offset

Register	Offset
PD_DPHY	300h

Function

Power Down Input for D-PHY

Diagram



Fields

Field	Function
31-1 —	Reserved
0 PD_DPHY	Power Down Input for D-PHY 0b - Power up 1b - Power down

54.7.1.57 Program T_HS_PREPARE (M_PRG_HS_PREPARE)

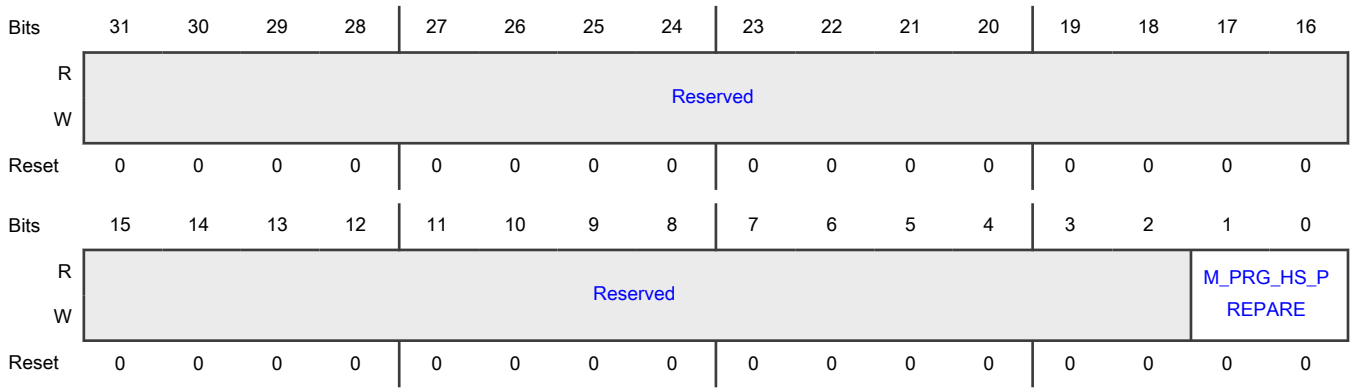
Offset

Register	Offset
M_PRG_HS_PREPARE	304h

Function

Used to program T_HS_PREPARE time.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 M_PRG_HS_PREPARE	DPHY m_PRG_HS_PREPARE Input Bits used to program T_HS_PREPARE time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming.

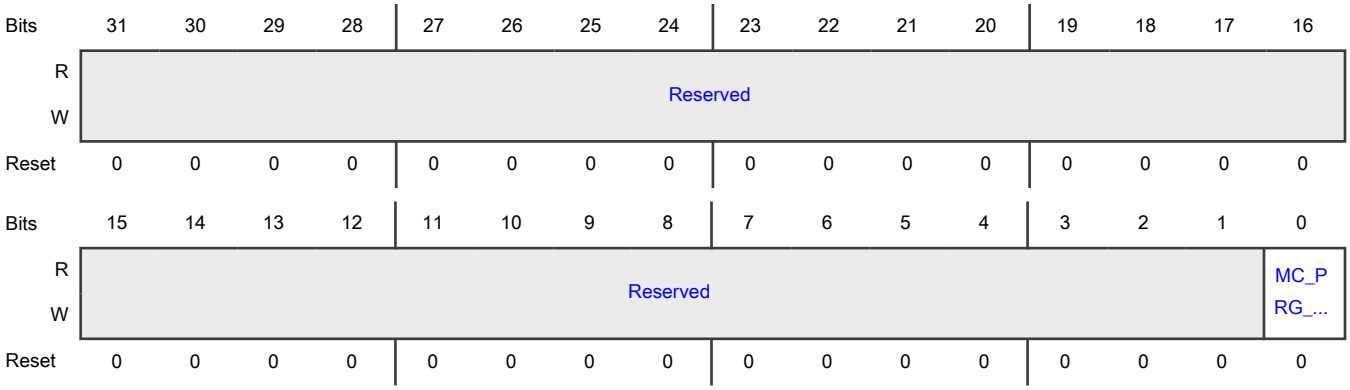
54.7.1.58 Program HS T_CLK_PREPARE (MC_PRG_HS_PREPARE)

Offset

Register	Offset
MC_PRG_HS_PREPARE	308h

Function
Used to program T_CLK_PREPARE time.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 MC_PRG_HS_PREPARE	DPHY mc_PRG_HS_PREPARE input Bits used to program T_CLK_PREPARE time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming.

54.7.1.59 Program T_HS_ZERO (M_PRG_HS_ZERO)

Offset

Register	Offset
M_PRG_HS_ZERO	30Ch

Function
Used to program T_HS_ZERO time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												M_PRG_HS_ZERO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-5 —	Reserved
4-0 M_PRG_HS_ZERO	DPHY m_PRG_HS_ZERO Input Bits used to program T_HS_ZERO time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming.

54.7.1.60 Program T_CLK_ZERO (MC_PRG_HS_ZERO)

Offset

Register	Offset
MC_PRG_HS_ZERO	310h

Function
Used to program T_CLK_ZERO time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										MC_PRG_HS_ZERO					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-6 —	Reserved
5-0 MC_PRG_HS_ZERO	DPHY mc_PRG_HS_ZERO Input Bits used to program T_CLK_ZERO time in the beginning of high speed transmission mode.For more details on how to program this field, see DPHY Programming.

54.7.1.61 Program T_HS_TRAIL (M_PRG_HS_TRAIL)

Offset

Register	Offset
M_PRG_HS_TRAIL	314h

Function
Used to program T_HS_TRAIL time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												M_PRG_HS_TRAIL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3-0 M_PRG_HS_TRAIL	DPHY m_PRG_HS_TRAIL Input Bits used to program T_HS_TRAIL time in the end of high speed transmission mode. For more details on how to program this field, see DPHY Programming.

54.7.1.62 Program T_CLK_TRAIL (MC_PRG_HS_TRAIL)

Offset

Register	Offset
MC_PRG_HS_TRAIL	318h

Function
Used to program T_CLK_TRAIL time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MC_PRG_HS_TRAIL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3-0 MC_PRG_HS_TRAIL	DPHY mc_PRG_HS_TRAIL input Bits used to program T_CLK_TRAIL time in the end of high speed transmission mode.

54.7.1.63 DPHY TST Input (TST)

Offset

Register	Offset
TST	31Ch

Function

Set TST[5:0]=[100101] at normal operation.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TST					
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1

Fields

Field	Function
31-6 —	Reserved
5-0 TST	DPHY TST Input

54.7.1.64 RTERM Select (RTERM_SEL)

Offset

Register	Offset
RTERM_SEL	320h

Function

RTERM_SEL selects voltage levels to enable HS termination.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															RTERM_S...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 RTERM_SEL	DPHY RTERM_SEL Input 0b - LPCD levels enables HS termination (VIL-CD). 1b - LPRX levels enables HS terminations (LP-VIL).

54.7.1.65 Power Down Auto Enable (AUTO_PD_EN)

Offset

Register	Offset
AUTO_PD_EN	324h

Function
Powers down inactive lanes reported by CFG_NUM_LANES input bus.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															AUTO_PD...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 AUTO_PD_EN	Power Down Auto Enable Input Powers down inactive lanes. 0b - Inactive lanes are powered up and driving LP11. 1b - Inactive lanes are powered down.

54.7.1.66 DPHY RXLPRP Input (RXLPRP)

Offset

Register	Offset
RXLPRP	328h

Function
RXLPRP adjusts the threshold voltage of LP-RX.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														RXLPRP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 RXLPRP	DPHY RXLPRP Input Default setting 2’b01.

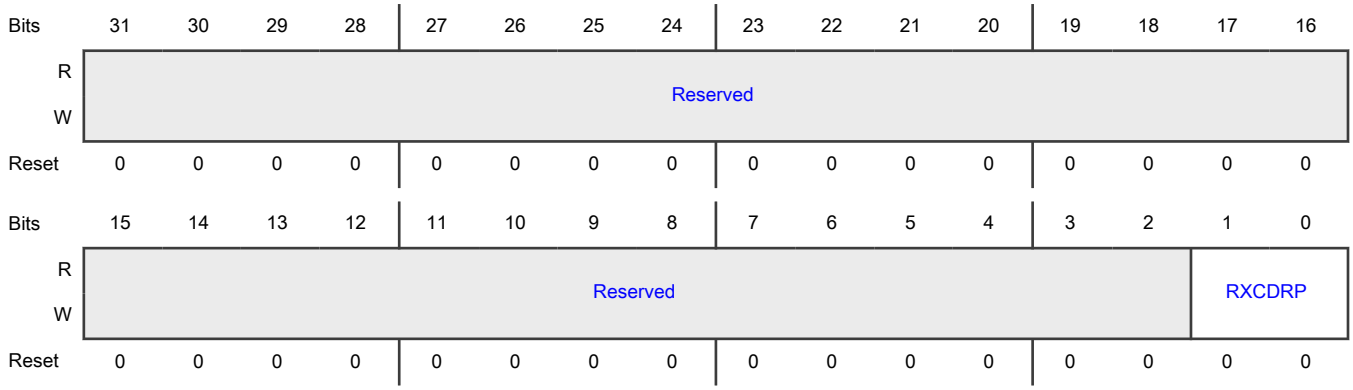
54.7.1.67 DPHY RXCDRP Input (RXCDRP)

Offset

Register	Offset
RXCDRP	32Ch

Function
RXCDRP adjusts the threshold voltage of LP-CD.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 RXCDRP	DPHY RXCDRP Input Default setting 2’b01.

Chapter 55
LCD Interface (LCDIF)

55.1 Chip-specific LCDIF information

Table 533. Reference links to related information

Topic	Related module	Reference
Full description	LCDIF	LCDIF
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

55.1.1 Module instances

This device contains one instance of the LCDIF module.

Table 534. LCDIF instance

	LCDIF
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

55.2 Overview

NOTE

This chapter describes the Vivante® IP DC8000Nano and contains copyright material disclosed with permission of VeriSilicon.

LCDIF is a high-performance, low-power display processor unit (DPU). It outputs video and graphic streams from a single output panel. The single output panel can connect to either a display pixel interface (DPI) device or a display bus interface (DBI) device to support RGB output.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 535. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

55.2.1 Block diagram

Figure 203 shows the function blocks.

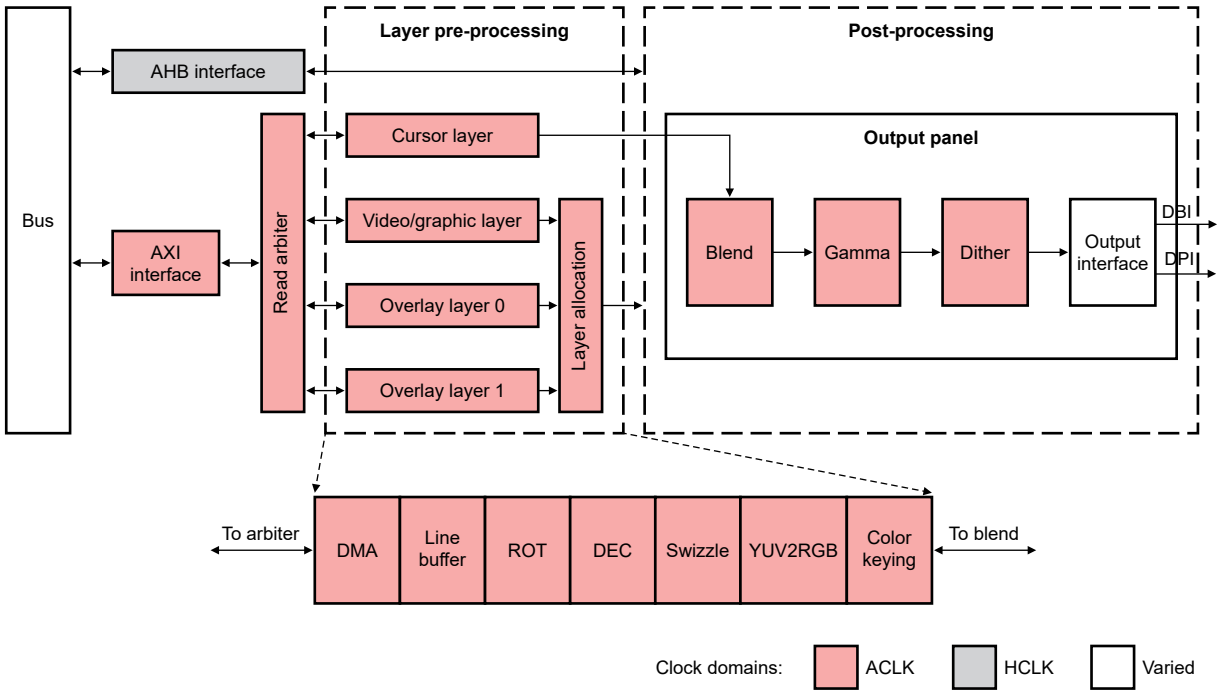


Figure 203. Block diagram

Table 536 summarizes the LCDIF submodules.

Table 536. Submodules

Module name	Description
AHB interface	Provides register access
AXI interface	Provides external memory access and reading of frame buffer data
Cursor	Includes one cursor layer to provide hardware cursor functionality
Layer	Supports: • One layer for video or graphics (VG layer) • Two overlay layers Each of these has DMA to support these pre-processing tasks: • Line buffer • Rotation or flip (ROT) • DECNano decompression and ETC2 decompression (DEC) • Swizzle • YUV-to-RGB color space conversion (YUV2RGB), except for overlay layer 1 • Color keying
Output panel	Provides post-processing computing functions, including:

Table continues on the next page...

Table 536. Submodules (continued)

Module name	Description
	• Alpha blending • Gamma • Dither
Output interface	Provides DPI or DBI for display output. LCDIF uses the DBI interface by default. The supported DBI types include: • Type A (Fixed E mode or Clocked E mode) • Type B These output interfaces reside in different clock domains.

55.2.2 Features

- One AXI initiator interface
  - 64-bit data bus and 32-bit address bus
  - RGB and YUV input formats in linear or tiled mode
- One AHB target interface with a 32-bit data bus and a 32-bit address bus
- One cursor layer
  - 32 × 32 cursor size
  - ARGB8888 and masked cursor formats
- Layers
  - One VG layer with a 4 × 1K - pixel line buffer
  - Two overlay layers (overlay 0 and overlay 1); overlay 0 has a 4 × 1K - pixel line buffer
  - Each layer supports:
    - Rotation and flip
    - DEC
    - Swizzle
    - Color keying
    - YUV-to-RGB conversion (except for overlay layer 1)
  - Post-processing for pixel composition
    - Alpha blending
    - Gamma correction with three separate lookup tables
    - Dither with one lookup table
  - Output interface
    - Single output panel supporting DPI and DBI (default) with RGB output
    - DBI types A and B support RGB swizzle
    - Display resolution at 60 fps:
      - 1024 × 768 in Tiled mode

- 1280 × 720 in Linear mode

— Double-buffered register support for display

— Power management with clock gating for RAM and layers

— Video timing generation with HSYNC, VSYNC, Data Enable (DE) signals programmable timers

55.3 Functional description

55.3.1 Interface features

LCDIF provides AXI and AHB as input interfaces, and supports interfaces to both DPI and DBI at the output.

55.3.1.1 Input formats

LCDIF supports various RGB and YUV input formats for the VG and overlay layers to read data. You can use linear or tiled reads:

- Linear reads access the buffer from left to right and from top to bottom, line by line.
- Tiled reads access the buffer tile by tile. This optimizes space and data readability.

NOTE

Overlay 1 does not support tiled reads.

The following table lists the supported input formats. One layer supports 2 KB of requests before data is returned.

Table 537. Input formats and alignments

Read type	Input format	AXI burst length (bytes)	Maximum read OTs per layer	Stride alignment (bytes)	Base address alignment (bytes)
Linear	RGB888	48	32	64	64
Linear	RGB888 compressed	8, 16, 24, or 32	32	64	64
Linear	ARGB8888	64	32	64	64
Linear	ARGB8888 compressed	8, 16, 24, 32, or 40	32	64	64
Linear	ARGB1555	64	32	64	64
Linear	RGB565	64	32	64	64
Linear	A8RGB565	48	32	64	64
Linear	ARGB4444	64	32	64	64
Tiled	Tiled RGB888	48 (3 bytes × 16 pixels)	32	64	64
Tiled	Tiled RGB888 compressed	16	32	64	64
Tiled	Tiled ARGB8888	64 (4 bytes × 16 pixels)	32	64	64
Tiled	Tiled ARGB8888 compressed	8, 16, 24	32	64	64

Table continues on the next page...

Table 537. Input formats and alignments (continued)

Read type	Input format	AXI burst length (bytes)	Maximum read OTs per layer	Stride alignment (bytes)	Base address alignment (bytes)
Tiled	Tiled A8RGB565	48	32	64	64
Tiled	Tiled YUY2	64 [2 bytes × (16+16) pixels]	32	64	64
Tiled	Tiled NV12	32	64	Y: 64 UV:32	Y: 64 UV:32

NOTE

The total number of read OTs for all the VG and overlay layers cannot exceed 64.

55.3.1.1.1 Linear formats

The pixel data in linear format is stored in the memory line by line, from left to right, and from top to bottom. You can store only RGB formats in Linear mode.

- For the 16-bpp RGB format data, 64 bytes are stored in 32×1 Linear mode
- For the 24-bpp RGB format data, 48 bytes are stored in 16×1 Linear mode
- For the 32-bpp RGB format data, 64 bytes are stored in 16×1 Linear mode.

Figure 204. 16-bpp RGB data stored in 32×1 Linear mode

Figure 205. 24-bpp or 32-bpp RGB pixel data stored in 16×1 Linear mode

55.3.1.1.2 Tiled ARGB8888/RGB888/A8RGB565 and tiled YUY2 formats

The ARGB8888/RGB888/A8RGB565 tiled format and the tiled YUY2 format use the 4×4 tile. The request bursts of the two formats fetch tiles in different ways:

- For the tiled ARGB8888/RGB888/A8RGB565 format, one burst fetches one tile. See Figure 206.

  Each tile is 4×4 with pixel data in 4×4 which are placed in the consecutive addresses as shown in the right diagram below in Figure 206. For ARGB8888, one 4×4 tile takes consecutive 64-byte memory space.

  Figure 206 illustrates n bursts to fetch the first 4 lines, with each burst fetching one tile.

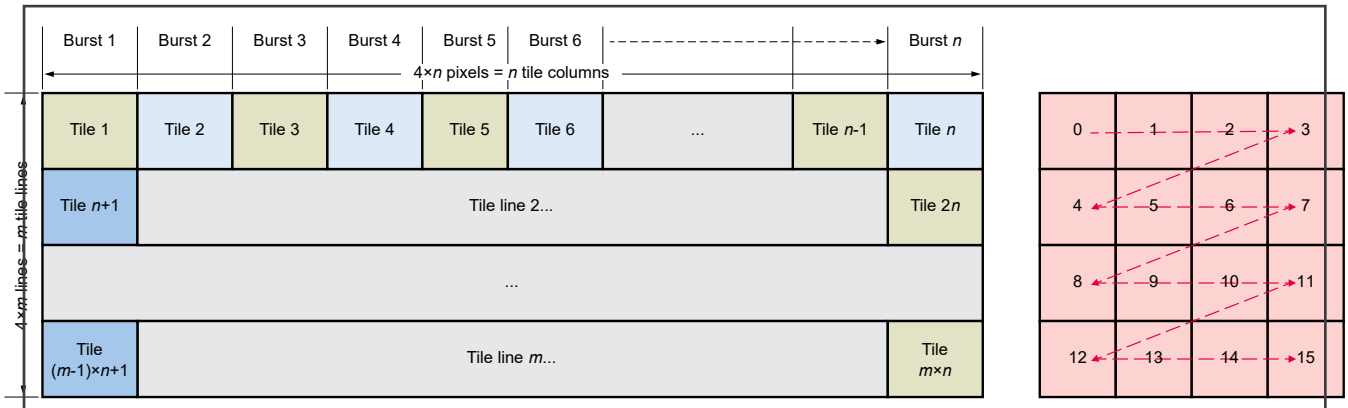


Figure 206. Screen space view: Request burst of the tiled ARGB8888 format (left) and pixel data read inside one 4×4 tile (right)

Although both of linear and tiled formats, LCDIF fetches pixels in consecutive addresses. Since LCDIF output is always linear, it needs to reorder the pixels in tiled format, and ultimately generates the linear output.

- For the tiled YUY2 format, one burst fetches two continuous tiles in the horizontal direction. The horizontal tile number for one frame of the YUY2 tiled format must be an even number. See Figure 207.

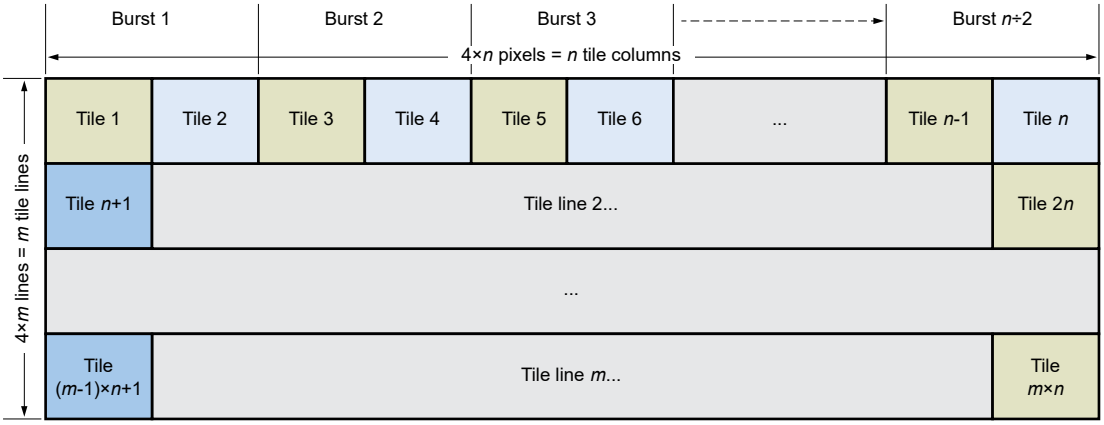


Figure 207. Screen space view: Request burst of the tiled YUY2 format

55.3.1.1.3 Tiled NV12 formats

The tiled NV12 format uses the 8×8 tile for the Y plane and the 4×4 tile for the UV plane. The AXI burst length is 32 bytes for both the Y and UV data.

For the Y plane, one burst fetches half the Y tile (8×4). Figure 208 shows that two bursts are used to fetch one entire 8×8 tile for the Y plane—one is the upper 8×4 part, and the other is the lower 8×4 part. Figure 209 and Figure 210 show the request bursts in one tile line and the tile arrangement, respectively.

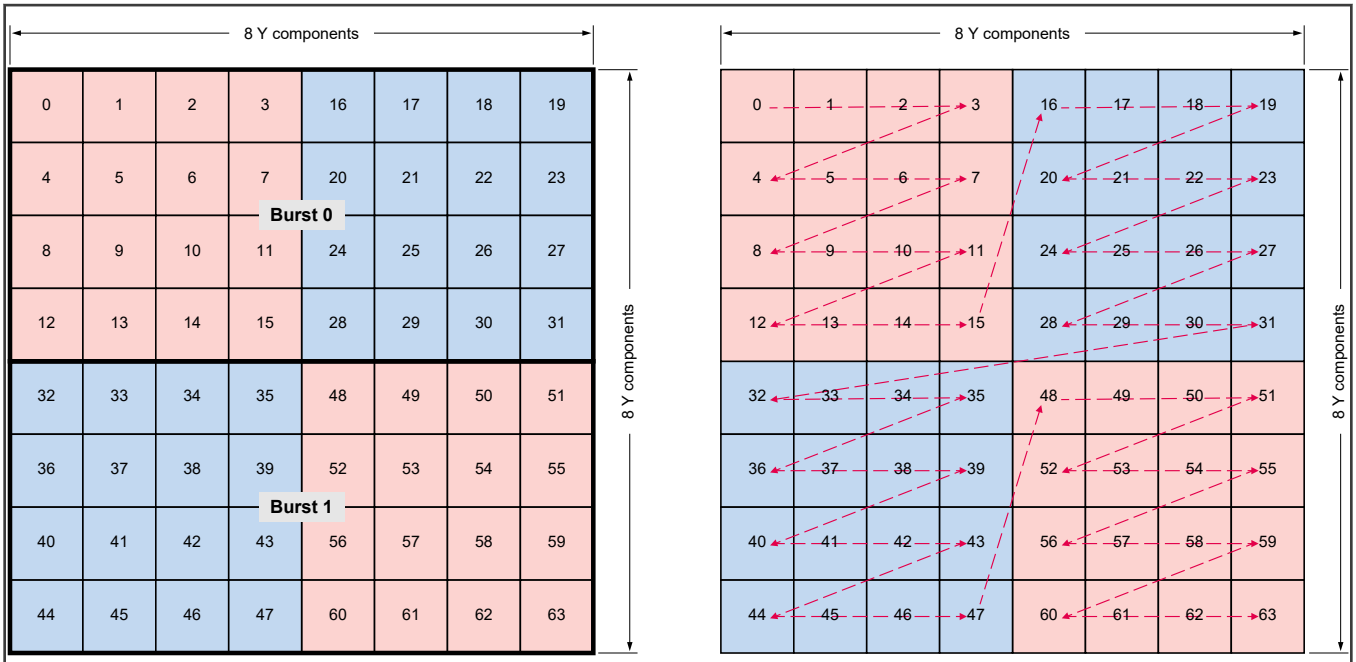


Figure 208. Request burst division and pixel data read inside one 8×8 tile of tiled NV12 Y plane

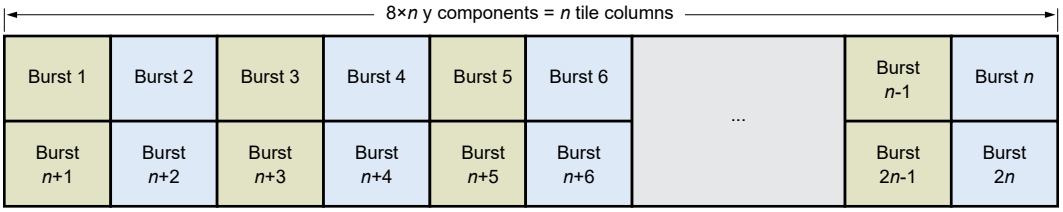


Figure 209. Request burst in one tile line for tiled NV12 Y plane

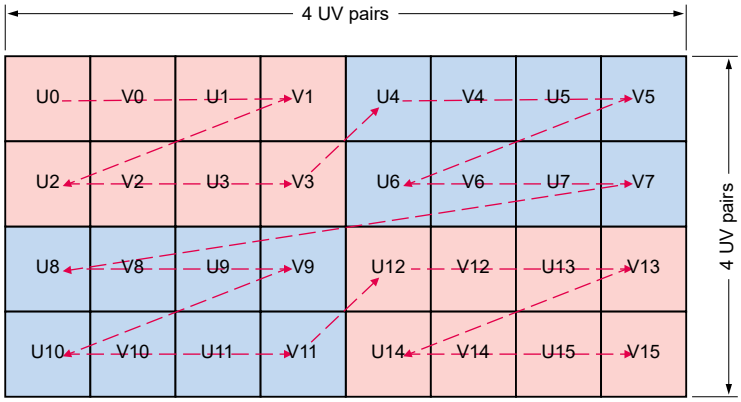


Figure 210. Pixel data read inside one 4×4 tile for tiled NV12 UV plane

For the UV plane, one burst fetches one UV tile, as shown in Figure 212 and Figure 211.

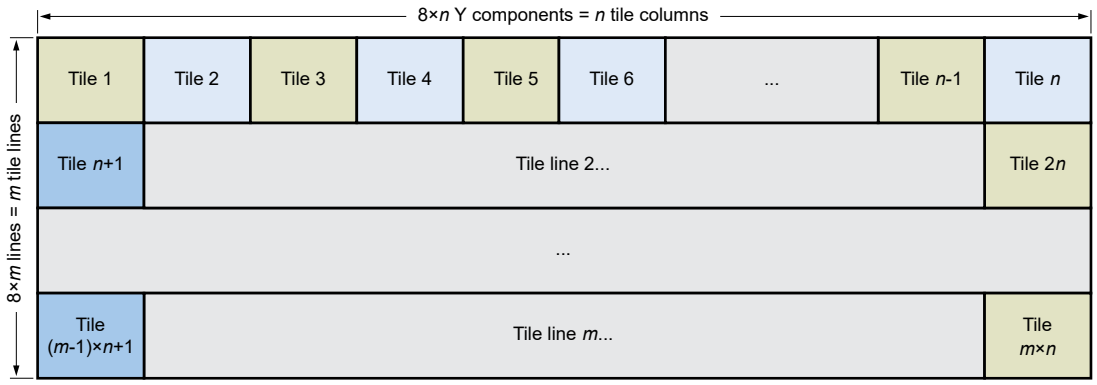


Figure 211. Screen space view: Tile arrangement for tiled NV12 Y plane

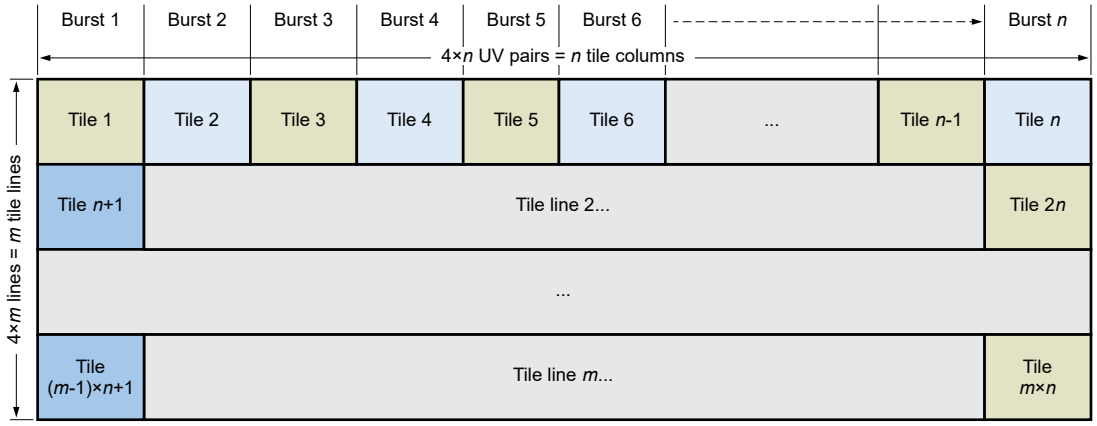


Figure 212. Screen space view: Request burst of tiled NV12 UV plane

55.3.1.1.4 Stride alignment

Stride alignment adds bytes to the end of each pixel line or tile line. This ensures that the data pattern meets the hardware requirement.

For data in linear format, the stride is used to align each pixel line. If one linear-format frame is not aligned to 64 bytes in width, you must pad the frame with dummy data after each pixel line to make it 64-byte aligned.

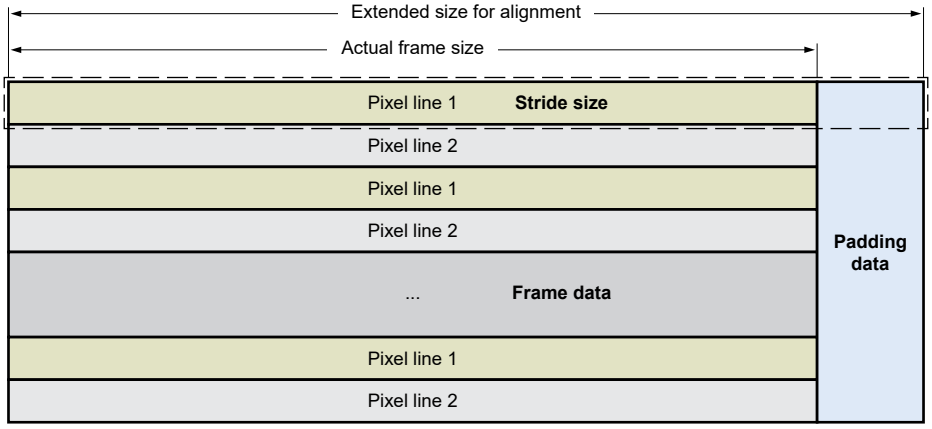


Figure 213. Stride alignment example for a frame in Linear mode

For data in tiled format, the stride is used to align each tile line. If one tiled-format frame is not aligned to the requested stride in the tile line, software pads dummy data after each tile line to make it aligned to the requested value.

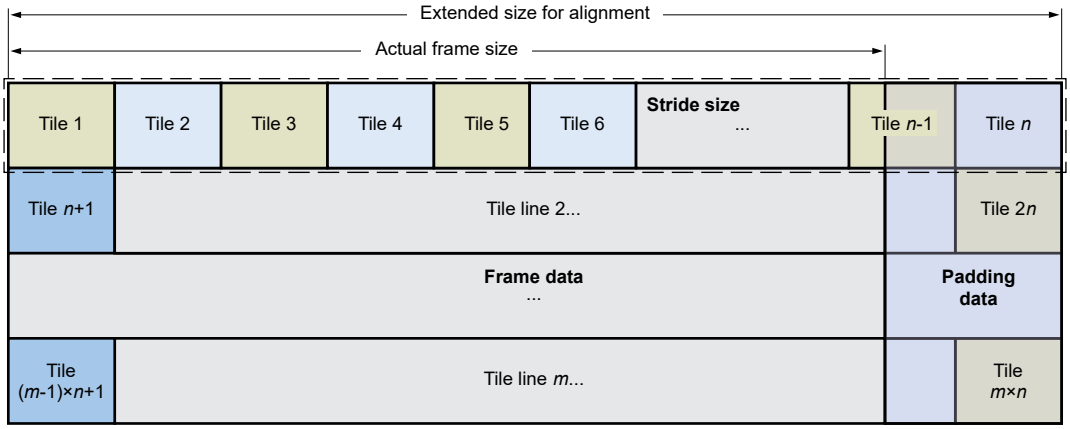


Figure 214. Stride alignment example for a frame in Tiled mode

When the input format is RGB888 or A8RGB565, you must set the stride to be the same as when the input format is ARGB8888. LCDIF then automatically calculates each tile address.

55.3.1.2 Output formats

The DPI interface supports RGB formats in 24-bit data signals. The DBI interface supports RGB formats in 8-bit, 9-bit, and 16-bit data signals.

The following sections describe the supported output formats with data bit maps.

55.3.1.2.1 DPI output formats

Figure 215 shows multiple RGB formats that the DPI interface supports.

RGB format	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D24*	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
D18CFG1							R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
D18CFG2			R5	R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0			B5	B4	B3	B2	B1	B0
D16CFG1									R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
D16CFG2				R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0				B4	B3	B2	B1	B0
D16CFG3			R4	R3	R2	R1	R0				G5	G4	G3	G2	G1	G0			B4	B3	B2	B1	B0	

*Default format

Figure 215. RGB formats for DPI output

DpiConfig0[DPI_DATA_FORMAT] specifies the DPI output format.

55.3.1.2.2 DBI output timing

The method of programming DBI timing varies with DBI types. Table 538 lists the timing registers used to set AC information (such as when to assert or de-assert a signal) for each DBI type. The display controller uses t_{unit} to drive control and data signals and calculate the write period time, which defines:

- Time unit of the ACLK cycle number for DBI Type A and Type B

t_{unit} = DbiConfig0[DBI_AC_TIME_UNIT] + 1 and the minimum value of the register field is 0.

MIPI DBI has defined different AC characteristics for each DBI type, and the timing register configuration must follow corresponding AC characteristics. The following sections describe the output timing configuration and constraints for each DBI type.

Table 538. DBI timing configuration register fields

Timing Register Field	DBI Type A Fixed E Mode	DBI Type A Clocked E Mode	DBI Type B
DbiConfig0[DBI_AC_TIME_UNIT]	Time unit factor of the ACLK cycle number	Time unit factor of the ACLK cycle number	Time unit factor of the ACLK cycle number
DbiWrChar10[DBI_WR_CS_ASSERT]	CSX assert timing	NA	NA
DbiWrChar10[DBI_WR_EOR_WR_ASSERT]	NA	E and D[15:0] assert timing	WRX and D[15:0] assert timing
DbiWrChar10[DBI_WR_PERIOD]	Total ACLK cycle number in a single write period = Register field value x t_{unit}		
DbiWrChar20[DBI_WR_EOR_WR_DE_ASRT]	NA	E de-assert timing	WRX de-assert timing
DbiWrChar20[DBI_WR_CS_DE_ASRT]	CSX de-assert timing	NA	NA

55.3.1.2.2.1 DBI Type A output timing configuration

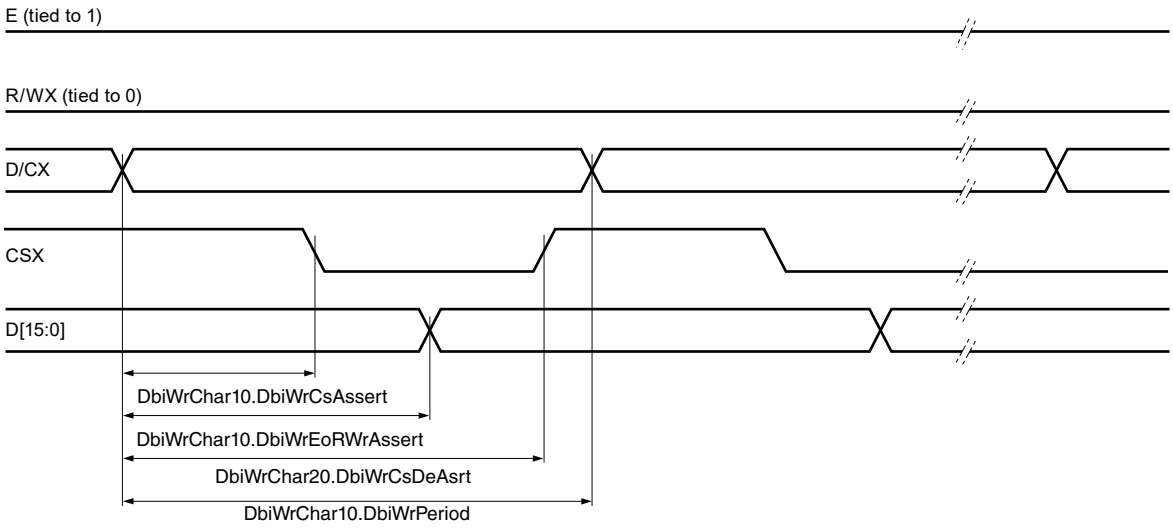


Figure 216. Output timing definition for DBI Type A fixed E mode

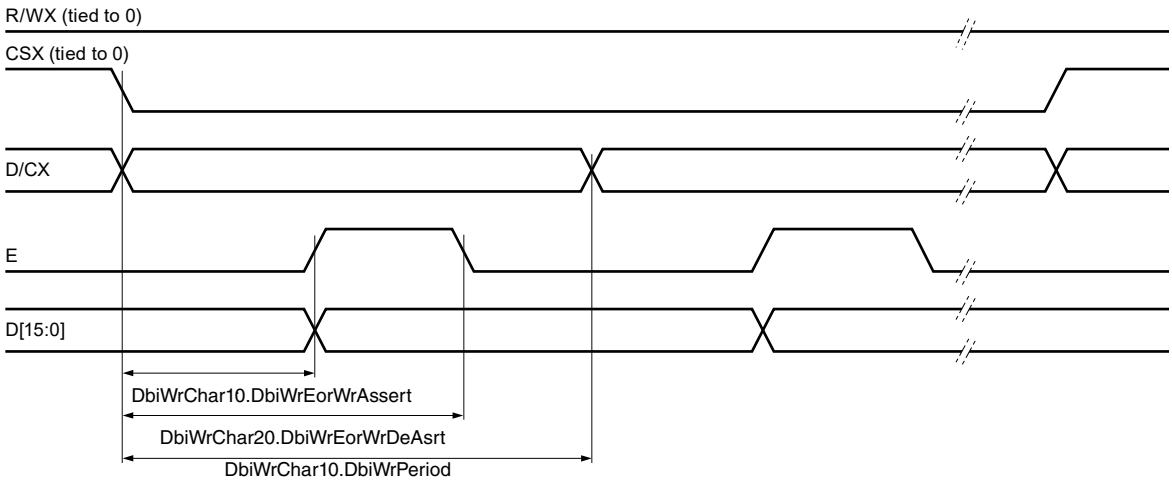


Figure 217. Output timing definition for DBI Type A clocked E mode

Table 539 shows the DBI signals used in output timing.

Table 539. DBI signals

signal	DBI Signal	Width	I/O	Description
CSX	dbiTypeab_csx	1	OUT	Chip Select for DBI Type A or B Display module is selected when low. It is set to 0 for DBI Type A in Clocked E mode.
D[15:0]	dbiTypeab_data	16	OUT	Information signal for DBI Type A or B
OE	dbiTypeab_oe	1	OUT	Data output enable for DBI Type A or B
D/CX	dbiTypeab_dcx	1	OUT	Data/Command indicator for DBI Type A or B Command is active when low.
R/WX for Type A RDX for Type B	dbiTypeab_rwx_or_rdx	1	OUT	For DBI Type A, specifies whether the current cycle is for read (1) or write (0) and set to 0. For DBI Type B, specifies whether the current cycle is for read (0) or write (1) and set to 1. For DBI Type A and Type B, read is not supported.
WRX	dbiTypeab_wrx	1	OUT	Write signal for DBI Type B Host processor writes data at falling edge.
E	dbiTypeaE	1	OUT	E clock for DBI Type A It is set to 1 for DBI Type A in Fixed E mode.

Table 540 describe the timing registers used for DBI Type A.

Table 540. Type A timing configuration register fields

DBI Type A timing register field	Description for fixed E mode	Description for clocked E mode
DbiWrChar10[DBI_WR_CS_ASSERT]	CSX assert timing	NA
DbiWrChar10[DBI_WR_EOR_WR_ASSERT]	D[15:0] assert timing	E and D[15:0] assert timing
DbiWrChar20[DBI_WR_EOR_WR_DE_ASRT]	NA	E de-assert timing
DbiWrChar20[DBI_WR_CS_DE_ASRT]	CSX de-assert timing	NA
DbiWrChar10[DBI_WR_PERIOD]	Single write period duration	Single write period duration

- The assert and de-assert register fields together with t_{unit} decide the ACLK cycle duration after which the corresponding signals are asserted or de-asserted.
  - For Fixed E mode: If DbiWrChar10[DBI_WR_CS_ASSERT] = 4, ACLK cycle number = 4 x t_{unit}. This means the CSX signal is asserted after (4 x t_{unit}) ACLK cycles.
  - For Clocked E mode: If DbiWrChar10[DBI_WR_EOR_WR_ASSERT] = 4, ACLK cycle number = 4 x t_{unit}. This means the E or D[15:0] signal is asserted after (4 x t_{unit}) ACLK cycles.
- DbiWrChar10[DBI_WR_PERIOD] decides the total ACLK cycle number in a single write period. The display controller includes a write period counter to manage signal driving. This register field configures the maximum value of this counter.

  Total cycle number in a single write period = DbiWrChar10[DBI_WR_PERIOD] x t_{unit}

  The write period counter starts from 0 for each signal. When it increments to the assert or de-assert timing configured by the assert or de-assert timing register fields, the corresponding signal is asserted or de-asserted.

55.3.1.2.2.2 DBI Type A output timing constraints

Figure 218, Figure 219 and Table 541 show the AC characteristics of DBI Type A defined in MIPI DBI.

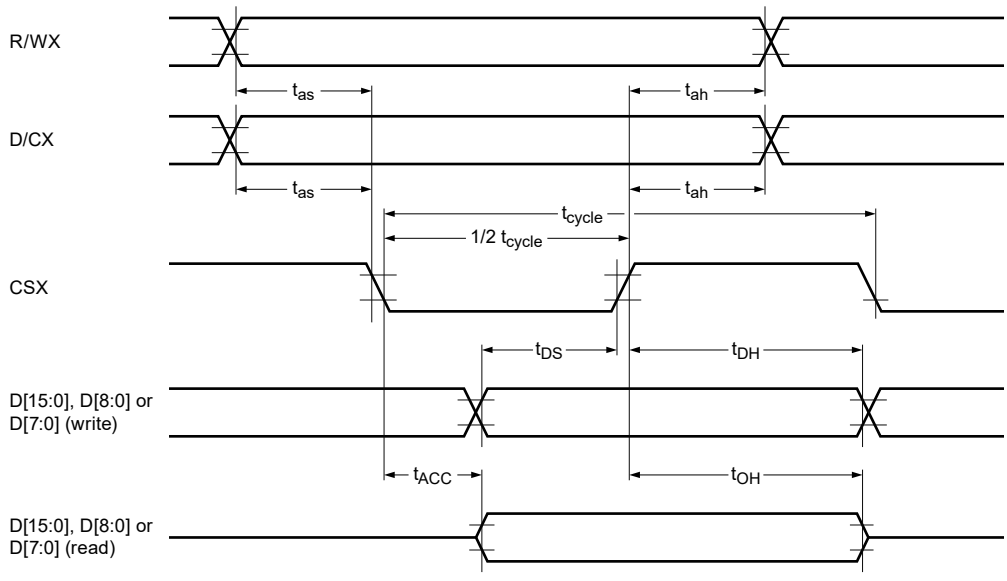


Figure 218. AC characteristics of DBI Type A fixed E mode

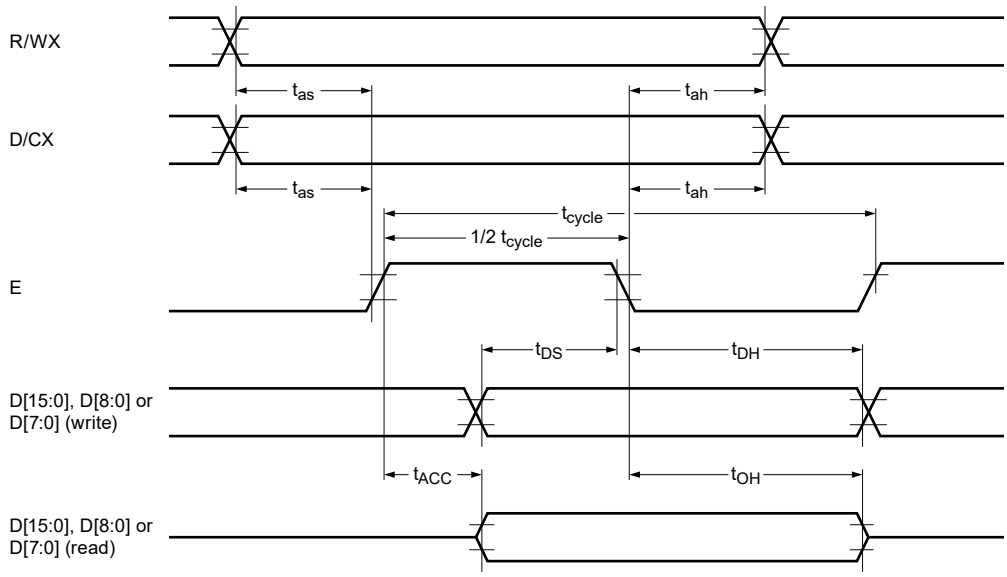


Figure 219. AC characteristics of DBI Type A clocked E mode

Table 541. AC characteristics description of DBI Type A

Signal	Symbol	Parameter	Min	Max	Unit	Description
R/WX or D/CX	t_{as}	Address setup time	T	NA	ns	
	t_{ah}	Address hold time(Write/ Read)	T	NA	ns	
CSX or E	t_{cycle}	System clock cycle time	5xT	79xT	ns	
D[15:0], D[8:0], or D[7:0]	t_{PS}	Data setup time	15	NA	ns	For maximum C_L = 30pF For minimum C_L = 8pF
	t_{PH}	Data hold time	25	NA	ns	
	t_{ACC}	Data access time	10	NA	ns	
	t_{OH}	Output hold time	10	NA	ns	

NOTE

- Ta = -30 to 70 °C, V_{DDI} range: according to Logic High level input voltage classification. GND = 0V, T = 10 ± 0.5 ns
- Does not include signal rise and fall times.

The timing register configuration for DBI Type A must meet the following requirements to ensure output timing follows the defined AC characteristics:

- For Fixed E mode:
  - DbiWrChar10[DBI_WR_CS_ASSERT] ≥ 1

    The time when the CSX signal is asserted is calculated using the following formula: DbiWrChar10[DBI_WR_CS_ASSERT] x t_{unit} x Single ACLK cycle duration.

As required by AC characteristics for DBI Type A Fixed E mode in Figure 218 and Table 541, CSX should be asserted after one t_{as}. One t_{as} lasts one T with 10 ns as the time unit for T. Therefore, the calculated CSX assert timing must be greater than or equal to 10 ns.

— DbiWrChar10[DBI_WR_CS_ASSERT] < DbiWrChar20[DBI_WR_CS_DE_ASRT] < DbiWrChar10[DBI_WR_PERIOD]

— DbiWrChar10[DBI_WR_PERIOD] ≥ 3

If the value of DbiWrChar10[DBI_WR_PERIOD] is less than 3, the display controller always changes it to 3.

- For Clocked E mode:

— DbiWrChar10[DBI_WR_EOR_WR_ASSERT] ≥ 1

The time when the E signal is asserted is calculated using the following formula:
DbiWrChar10[DBI_WR_EOR_WR_ASSERT] x t_{unit} x Single ACLK cycle duration.

As required by AC characteristics for DBI Type A Clocked E mode in Figure 219 and Table 541, E should be asserted after one t_{as}. One t_{as} lasts one T with 10 ns as the time unit for T. Therefore, the calculated E assert timing must be greater than or equal to 10 ns.

— DbiWrChar10[DBI_WR_EOR_WR_ASSERT] < DbiWrChar20[DBI_WR_EOR_WR_DE_ASRT] < DbiWrChar10[DBI_WR_PERIOD]

— DbiWrChar10[DBI_WR_PERIOD] ≥ 3

If the value of DbiWrChar10[DBI_WR_PERIOD] is less than 3, the display controller always changes it to 3.

55.3.1.2.2.3 DBI Type B output timing configuration

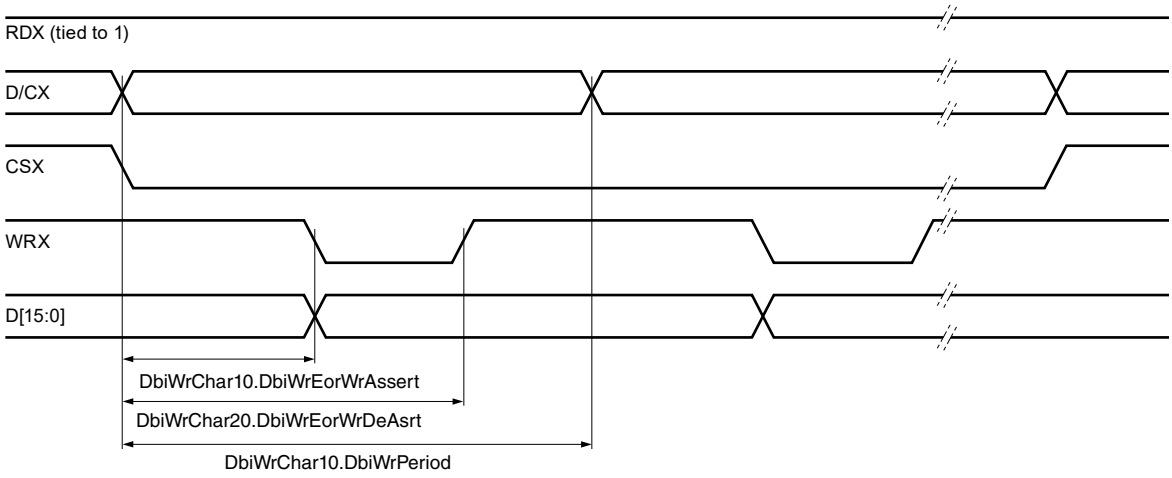


Figure 220. Output timing definition for DBI Type B

For details of DBI signals used for output timing, see Table 539.

Table 542 describe the timing registers used for DBI Type B.

Table 542. Type B timing configuration register fields

DBI Type B timing register field	Description for DBI Type B
DbiWrChar10[DBI_WR_EOR_WR_ASSERT]	WRX and D[15:0] assert timing
DbiWrChar20[DBI_WR_EOR_WR_DE_ASRT]	WRX de-assert timing
DbiWrChar10[DBI_WR_PERIOD]	Single write period duration

- The assert and de-assert register fields together with t_{unit} decide the ACLK cycle duration after which the corresponding signals are asserted or de-asserted.

  If DbiWrChar10[DBI_WR_EOR_WR_ASSERT] = 4, the WRX or D[15:0] signal is asserted after (4 × t_{unit}) ACLK cycles.

- DbiWrChar10[DBI_WR_PERIOD] decides the ACLK cycle number in a single write period.

  Total cycle number in a single write period = DbiWrChar10[DBI_WR_PERIOD] x t_{unit}

  The display controller includes a write period counter to manage signal driving. This register field configures the maximum value of this counter.

  The write period counter starts from 0 for each signal. When it increments to the assert or de-assert timing configured by the assert or de-assert timing register fields, the corresponding signal is asserted or de-asserted.

55.3.1.2.2.4 DBI Type B output timing constraints

Figure 221 and Table 543 show the AC characteristics of DBI Type B defined in MIPI DBI.

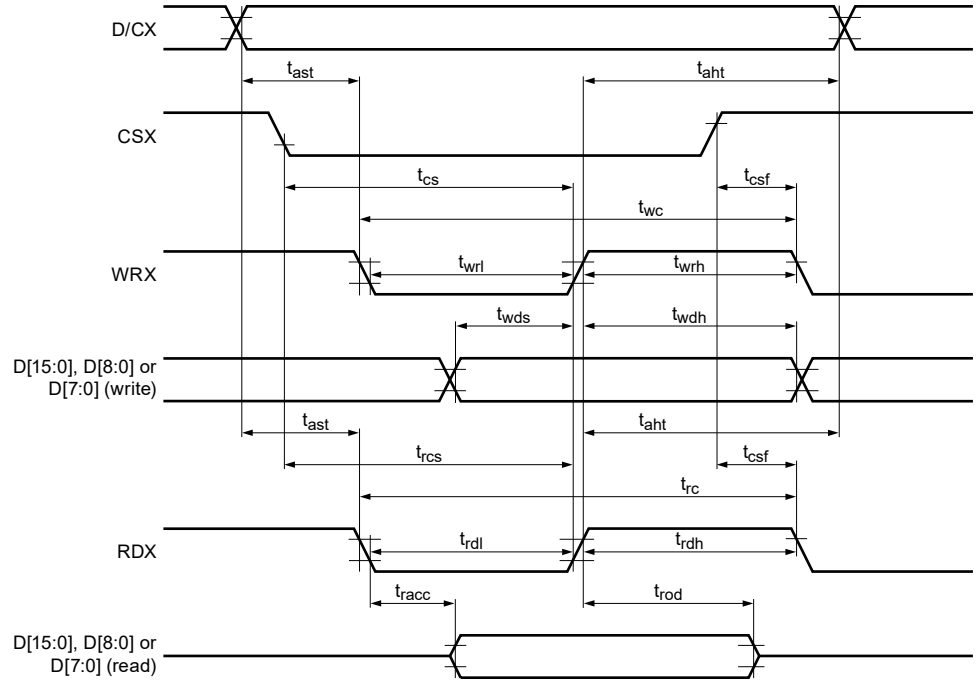


Figure 221. AC characteristics of DBI Type B

Table 543. AC characteristics description of DBI Type B

Signal	Symbol	Parameter	Min	Max	Unit	Description
D/CX	t_{ast}	Address setup time	T	NA	ns	
	t_{aht}	Address hold time(Write/ Read)	T	NA	ns	
CSX or E	t_{cs}	Chip Select setup time (Write)	2xT	NA	ns	(2xT, 3xT, …, 16xT)

Table continues on the next page...

Table 543. AC characteristics description of DBI Type B (continued)

Signal	Symbol	Parameter	Min	Max	Unit	Description
	t_{rcs}	Chip Select setup time (Read)	2xT	NA	ns	(2xT, 3xT, …, 16xT)
	t_{csf}	Chip Select wait time (Write/ read)	20	NA	ns	NA
WRX	t_{wc}	Write cycle	5xT	79xT	ns	(5xT, 6xT, …, 79xT)
	t_{wrh}	Write Control pulse H duration	3xT	63xT	ns	(5xT, 6xT, …, 79xT)
	t_{wrl}	Write Control pulse L duration	2xT	16xT	ns	(2xT, 3xT, …, 16xT)
RDX	t_{rc}	Read cycle	5xT	79xT	ns	(5xT, 6xT, …, 79xT)
	t_{rdh}	Read Control pulse H duration	3xT	63xT	ns	(5xT, 6xT, …, 79xT)
	t_{rdl}	Read Control pulse L duration	2xT	16xT	ns	(2xT, 3xT, …, 16xT)
D[15:0], D[8:0], or D[7:0]	t_{wds}	Write data setup time	15	NA	ns	For maximum C_L = 30pF For minimum C_L = 8pF
	t_{wdh}	Write data hold time	25	NA	ns	
	t_{racc}	Read access time	10	NA	ns	
	t_{rod}	Read output disable time	10	NA	ns	

NOTE

- Ta = -30 to 70 °C, V_{DDI} range: according to Logic High level input voltage classification. GND = 0V, T = 10 ± 0.5 ns
- Does not include signal rise and fall times.

The timing register configuration for DBI Type B must meet the following requirements to ensure output timing follows the defined AC characteristics:

- DbiWrChar10[DBI_WR_EOR_WR_ASSERT] ≥ 1

  The time when the WRX signal is asserted is calculated using the following formula: DbiWrChar10[DBI_WR_EOR_WR_ASSERT] x t_{unit} x Single ACLK cycle duration.

  As required by AC characteristics for DBI Type B in Figure 221 and Table 543, WRX should be asserted after one t_{ast}. One t_{ast} lasts one T with 10 ns as the time unit for T. Therefore, the calculated WrX assert timing must be greater than or equal to 10 ns.
- DbiWrChar10[DBI_WR_EOR_WR_ASSERT] < DbiWrChar20[DBI_WR_EOR_WR_DE_ASRT] < DbiWrChar10[DBI_WR_PERIOD]
- DbiWrChar10[DBI_WR_PERIOD] ≥ 3

If the value of DbiWrChar10[DBI_WR_PERIOD] is less than 3, the display controller always changes it to 3.

55.3.1.2.3 DBI Type A and Type B output formats

DBI Type A and Type B support swizzle for RGB formats.

Figure 222 shows the RGB formats for DBI type-A and type-B output without swizzle.

RGB format	Cycle	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R3G3B2*										R			G			B	
D8R4G4B4	0									R0				G0			
	1									B0				R1			
	2									G1				B1			
D8R5G6B5	0									R					G[5-3]		
	1									G[2-0]			B				
D8R6G6B6	0									R							
	1									G							
	2									B							
D8R8G8B8	0									R							
	1									G							
	2									B							
D9R6G6B6	0									R					G[5-3]		
	1									G[2-0]			B				
D16R3G3B2		R0			G0			B0		R1			G1			B1	
D16R4G4B4						R				G				B			
D16R6G6B6 option1	0	R0								G0							
	1	B0								R1							
	2	G1								B1							
D16R6G6B6 option2	0									R							
	1	G								B							
D16R8G8B8 option1	0	R0								G0							
	1	B0								R1							
	2	G1								B1							
D16R8G8B8 option2	0									R							
	1	G								B							

*Default format

Figure 222. RGB formats for DBI type-A and type-B output without swizzle

You enable swizzle using DbiConfig[DBI_OUTPUT_SWIZZLE]. When swizzle is enabled, the configured RGB formats are reshaped in BGR order, as shown in Figure 223.

RGB format	Cycle	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R3G3B2*										B		G			R		
D8R4G4B4	0									B0					G0		
	1									R0					B1		
	2									G1					R1		
D8R5G6B5	0									B					G[5-3]		
	1									G[2-0]					R		
D8R6G6B6	0									B							
	1									G							
	2									R							
D8R8G8B8	0									B							
	1									G							
	2									R							
D9R6G6B6	0									B					G[5-3]		
	1									G[2-0]					R		
D16R3G3B2		B0			G0			R0		B1		G1			R1		
D16R4G4B4						B				G					R		
D16R6G6B6 option1	0	B0								G0							
	1	R0								B1							
	2	G1								R1							
D16R6G6B6 option2	0									B							
	1	G								R							
D16R8G8B8 option1	0	B0								G0							
	1	R0								B1							
	2	G1								R1							
D16R8G8B8 option2	0									B							
	1	G								R							

*Default format

Figure 223. RGB formats for DBI type-A and type-B output with swizzle

DbiConfig[DBI_DATA_FORMAT] specifies an output format for DBI types A and B. Valid values of DBI_DATA_FORMAT for these types are 0h to Ch.

55.3.2 Layer features

55.3.2.1 Hardware cursor

LCDIF provides one cursor layer. A cursor is tracked through the top-left point of the cursor layer and the hotspot of the cursor.

55.3.2.1.1 Top-left point

The address of this point is used as the base address of the cursor, which you can configure using Cursor Base Address (CursorAddress).

This point varies with the cursor layer location:

- If a cursor layer is fully displayed in a display image, this point is the top-left point of the cursor image.

- If a cursor layer has an entire line outside the vertical boundary of a display image, the top-left point is the top point of the leftmost covered cursor layer line. Pixels in the uncovered cursor layer are not fetched from DMA.
- If a cursor layer has an entire line outside the horizontal boundary of a display image, the top-left point is still the top-left of the cursor layer. Pixels in the uncovered cursor layer are dropped after alpha blending.
- If a cursor layer has entire lines outside both the vertical and horizontal boundaries of a display image, the top-left point is the top point of the covered leftmost cursor layer line. Pixels in the uncovered cursor layer are dropped after alpha blending.

55.3.2.1.2 Hotspot

This is the point used in a search for a screen coordinate when someone makes a selection—for example, by pressing the "Enter" key on a keyboard or clicking the left button on a mouse.

The hotspot location determines the cursor position in a display image.

55.3.2.1.3 Cursor layer fully displayed

Figure 224 shows a cursor layer fully displayed in a display image. Table 544 shows the coordinates used in the figure and this section.

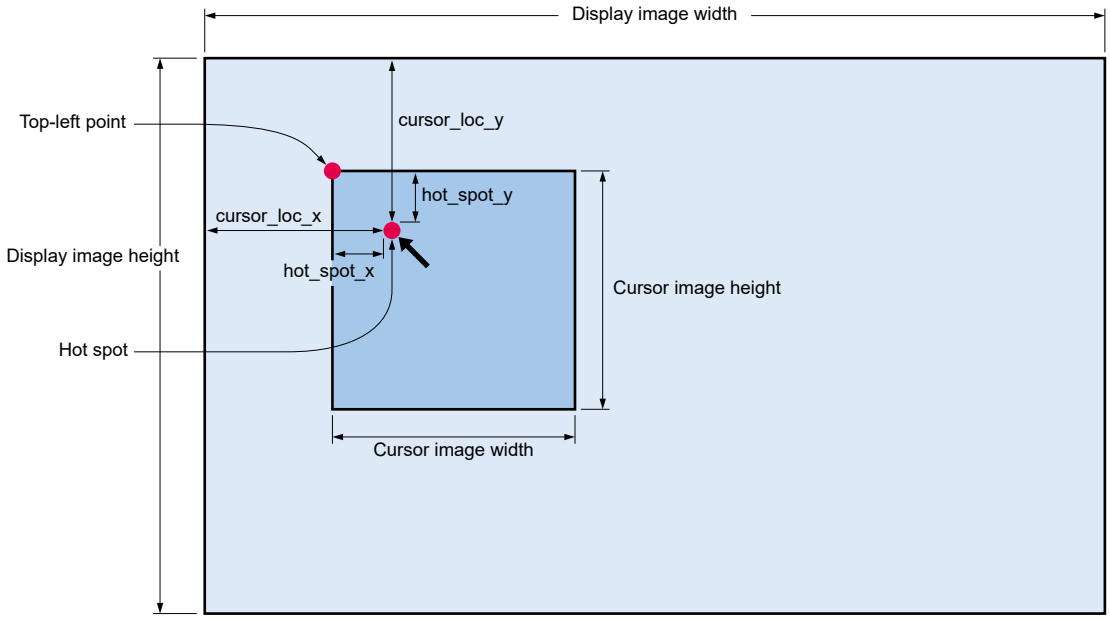


Figure 224. Cursor layer fully displayed

Table 544. Coordinates

Coordinate	Description	Field that specifies the coordinate
cursor_loc_x	Horizontal cursor location	CursorLocation[X]
cursor_loc_y	Vertical cursor location	CursorLocation[Y]
hot_spot_x	Horizontal offset to the cursor hotspot	CursorConfig[HOT_SPOT_X]
hot_spot_y	Vertical offset to the cursor hotspot	CursorConfig[HOT_SPOT_Y]

The hotspot and cursor locations must meet the following requirements to ensure a cursor layer is fully displayed:

- 0 ≤ cursor_loc_x ≤ Display image width

- 0 ≤ cursor_loc_y ≤ Display image height
- 0 ≤ hot_spot_x ≤ Cursor image width
- 0 ≤ hot_spot_y ≤ Cursor image height

55.3.2.1.4 Cursor layer partially displayed

The following figure shows some special cases where a cursor layer is partially displayed in a display image.

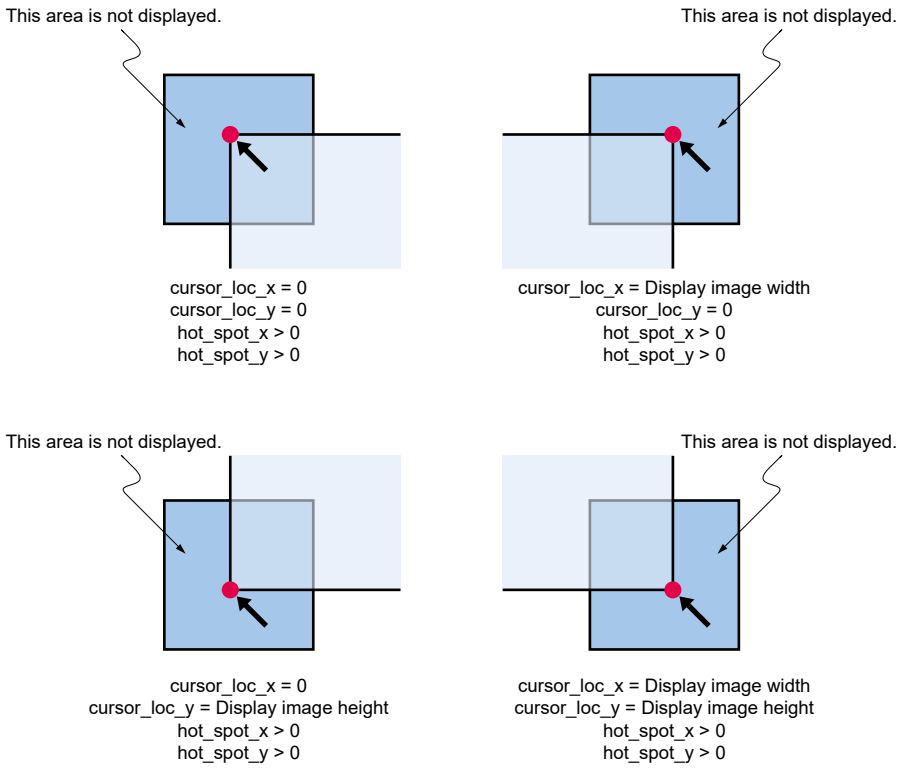


Figure 225. Cursor layer partially displayed

55.3.2.1.5 Cursor formats

The cursor image can be:

- Up to 32×32 pixels in size.
- In either ARGB8888 or masked format.

In the ARGB8888 cursor format, the ARGB values that are fetched from memory determine the color of the cursor.

In the masked cursor format, the mask values of the AND and XOR bits determine the color of one cursor pixel. The following table lists the color values for different bit configurations. Cursor Foreground Color (CursorForeground) and Cursor Background Color (CursorBackground) specify the foreground and background colors, respectively.

Table 545. Color values for different bit configurations

Bit AND	Bit XOR	Color value
0	0	Background color
0	1	Foreground color

Table continues on the next page...

Table 545. Color values for different bit configurations (continued)

Bit AND	Bit XOR	Color value
1	0	Not used
1	1	Bit inversed foreground color

55.3.2.2 Rotation and flip

LCDIF supports 180-degree rotation and flipping (along the X axis, Y axis, or both X and Y axes) of the VG and overlay layers. A 180-degree rotation is equivalent to a flip along the X and Y axes.

You cannot perform a rotation and flip simultaneously.

Use the following fields to control rotation or flip for each layer:

- OverlayConfig[ROT_ANGLE] for overlay layer 0
- OverlayConfig1[ROT_ANGLE] for overlay layer 1

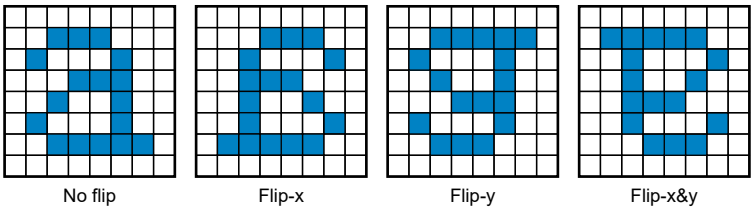


Figure 226. Flip

55.3.2.3 Decompression

Each layer except for the cursor layer provides these types of decompression to decompress input data:

- DECNano decompression supporting linear or tiled ARGB8888 and RGB888 formats
- ETC2 decompression supporting only tiled ARGB8888 format

The following table lists the supported modes of the decompression types. Use FrameBufferConfig0[DEC_MODE] to select the decompression mode.

Table 546. Decompression modes

Decompression type	Decompression mode	Compression ratio	Supported input format	AXI burst length (bytes)
DECNano	Nonsample	8:5	Linear ARGB8888	8, 16, 24, 32, or 40
DECNano	Nonsample	6:4	Linear RGB8888	32
DECNano	H sample	8:4	Linear ARGB8888	32
DECNano	H sample	6:3	Linear RGB8888	8, 16, or 24
DECNano	HV sample	8:3	Tiled ARGB8888	8, 16, or 24
DECNano	HV sample	6:2	Tiled RGB8888	16
ETC2		8:2	Tiled ARGB8888	16

NOTE

HV sample is illegal when the input data is in linear format.

The AXI burst length for compressed formats varies with the compression ratio, but the AXI outstanding request (OT) number is not affected.

You must change the stride and base address alignments for compressed formats according to the decompression modes as follows:

- If you enabled DECNano decompression for the ARGB8888 or RGB888 format, use the same stride as when decompression is disabled. In this case, LCDIF calculates each tile address based on the compression ratio of the decompression mode.
- If you enabled ETC2 decompression, set the stride to 25% of the original image's stride.

55.3.2.4 Swizzle

The components of one color format may be stored in the memory in different orders. To accommodate this possibility, LCDIF can swizzle an input pixel at the VG and overlay layers to align the component order. LCDIF supports swizzle for both RGB and YUV formats.

55.3.2.4.1 RGB swizzle

For RGB formats, the input pixel may be in any of the following format orders:

- ARGB
- RGBA
- ABGR
- BGRA

LCDIF assumes the initial input of ARGB data is in ARGB order, which means A is in the upper bits while B is in the lower bits. If the input is not in ARGB order, LCDIF uses swizzle to reshape the data. For example, if the input format is in BGRA order, you must use the following fields to configure the swizzle to handle the BGRA format:

- FrameBufferConfig0[SWIZZLE]
- OverlayConfig[SWIZZLE]
- OverlayConfig1[SWIZZLE]

The following figure shows the expected input order for the RGB format. If the input RGB data is not in this order, swizzle is required.

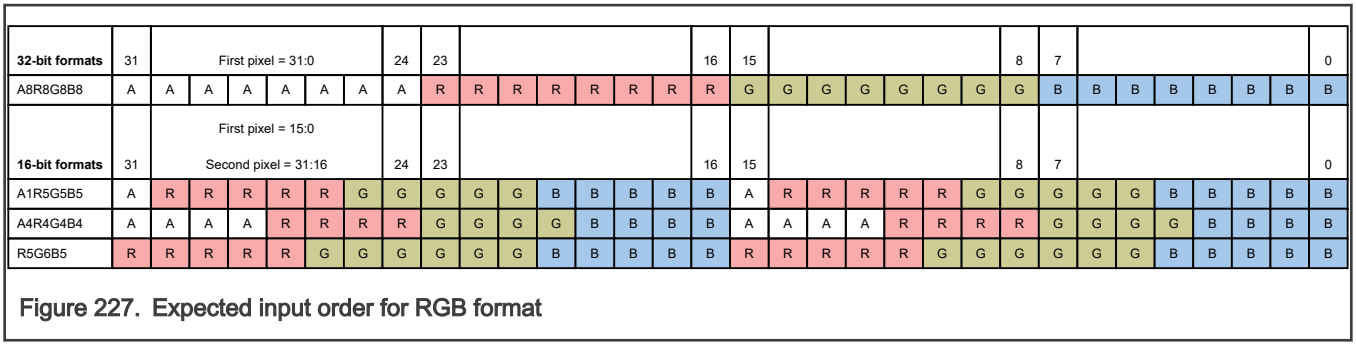


Figure 227. Expected input order for RGB format

55.3.2.4.2 Semi-planar YUV swizzle

For YUV formats, especially for UV mixed formats, the U and V pixels may be placed in different positions. LCDIF assumes that for each UV pair, U is in the lower bits while V is in the upper bits. If the input is not in this order, LCDIF uses UV-swizzle to reshape the input data. Use the following fields to configure the swizzle:

- FrameBufferConfig0[UV_SWIZZLE]
- OverlayConfig[UV_SWIZZLE]

55.3.2.5 YUV-to-RGB conversion

LCDIF supports YUV-to-RGB color space conversion for the VG and overlay 0 layers.

The conversion standard can be either BT.601 or BT.709. Use DcTileInCfg[YUV_STANDARD] and DcOverlayTileInCfg[YUV_STANDARD] to specify the standard.

In the YUV-to-RGB conversion, LCDIF uses an 8-bit pipeline to convert the YUV data into an RGB format with 8 bits for each of the R, G, and B components.

55.3.2.6 Color keying

LCDIF provides overlay and transparency effects by using color keys at the VG and overlay layers. With color keying, LCDIF filters out a special color in a source image and replaces the color with another designated color of the destination image.

Color keying is usually performed before alpha blending and is supported only in ARGB format.

Typical application scenarios include:

- Displaying an interesting scene—such as a weather map—behind people
- Creating special effects—such as floating a car on the ocean

55.3.2.6.1 Color-keying process

LCDIF performs color keying on every two of the VG and overlay layers, together with one background layer in an ascending layer order. If any of the VG or overlay layers are disabled, LCDIF does not apply color keying to the disabled layer.

The following figure shows the default layer arrangement—which is, in descending order:

4) Overlay layer 1

3) Overlay layer 0

2) VG layer

1) Background layer

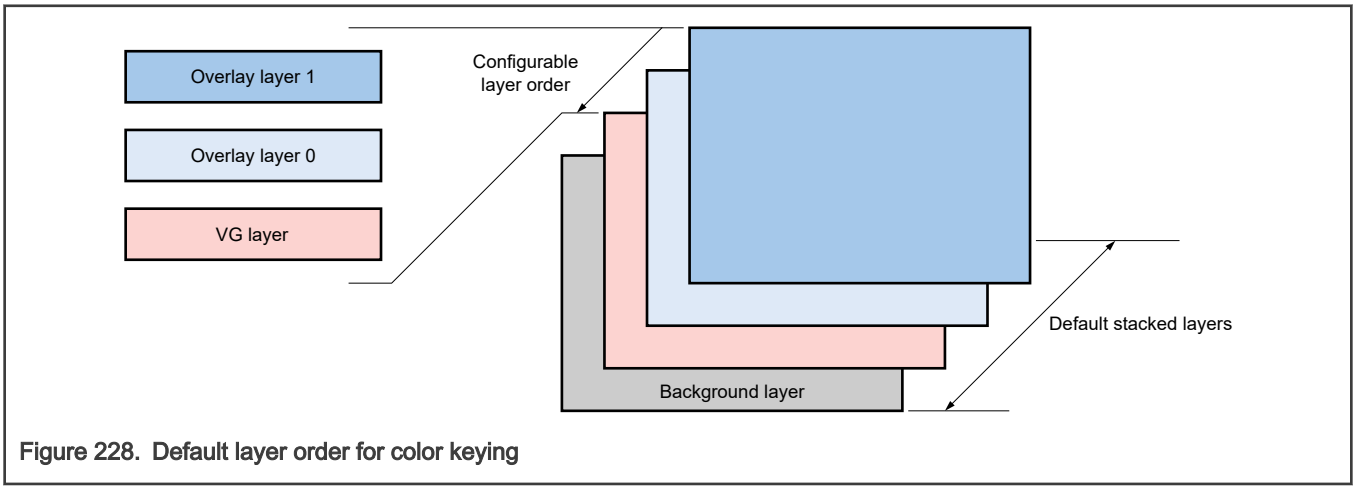


Figure 228. Default layer order for color keying

You can reconfigure the layer ordering using Blend Stack Order (BlendStackOrder). You can also specify the color of the background layer using Frame Buffer Background (FrameBufferBackground).

When color keying is enabled and the default layer order is used, LCDIF performs color keying on the paired layers in the following sequence:

1. Color keying for background and VG layers

   The VG layer is the source layer and the background layer is the destination layer.

   LCDIF replaces the selected special color of the VG layer with the background color.

LCDIF then treats the color-keyed VG layer as the destination layer for the color keying of the overlay 0 layer and the VG layer.

2. Color keying for the overlay 0 layer and the VG layer

   Overlay layer 0 is the source layer and the color-keyed VG layer is the destination layer.

   LCDIF replaces the selected special color of the overlay 0 layer with the pixel color of the color-keyed VG layer.

   LCDIF then treats the color-keyed overlay layer 0 as the destination layer for the color keying of the overlay 1 and overlay 0 layers.

3. Color keying for overlay 0 and overlay 1 layers

   Overlay layer 1 is the source layer and the color-keyed overlay layer 0 is the destination layer.

   LCDIF replaces the selected special color of overlay layer 1 with the pixel color of the color-keyed overlay 0 layer.

55.3.2.6.2 Color-keying implementation

In a typical case, you have at least one source layer (foreground) and one destination layer (background), with the source layer displayed in front of the destination layer. The source layer contains a solid color (typically, blue or green).

LCDIF performs color keying as follows:

1. Compare the component value of each pixel color of the source layer (R, G, B, or A of the color) with the color key range defined in the following registers, which are 32 bits of the ARGB8888 format. See also Figure 229.
   - Frame Buffer Color Keying Start Address (FrameBufferColorKey)
   - Frame Buffer Color Key End Address (FrameBufferColorKeyHigh)
   - Overlay Color Keying Start Address (OverlayColorKey)
   - Overlay Color Keying End Address (OverlayColorKeyHigh)
   - Overlay Layer 1 Color Keying Start Address (OverlayColorKey1)
   - Overlay Layer 1 Color Keying End Address (OverlayColorKeyHigh1)
2. When all of the components of the pixel color are in the range, LCDIF treats it as a special color and filters it out.
3. When the two layers are blended, LCDIF uses the color of the destination layer to replace the special solid color in the source layer.

Color format	31							24	23						16	15							8	7							0
A8R8G8B8	A	A	A	A	A	A	A	A	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B

Figure 229. Configuration for the color-key registers

Figure 230 shows an example of the blending result after color keying.

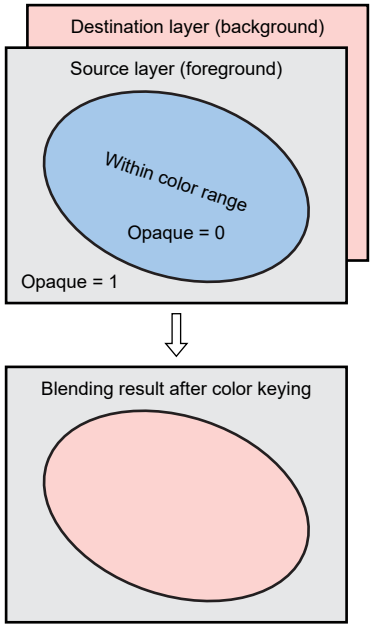


Figure 230. Blending result after color keying

55.3.3 Post-processing features

55.3.3.1 Alpha blending

LCDIF supports alpha blending for the cursor, VG, and overlay layers, together with one background layer.

Figure 231 shows the default order for alpha blending. The cursor layer is always topmost and the background layer with a pure color is bottommost. You can change the order of the VG and overlay layers using Blend Stack Order (BlendStackOrder).

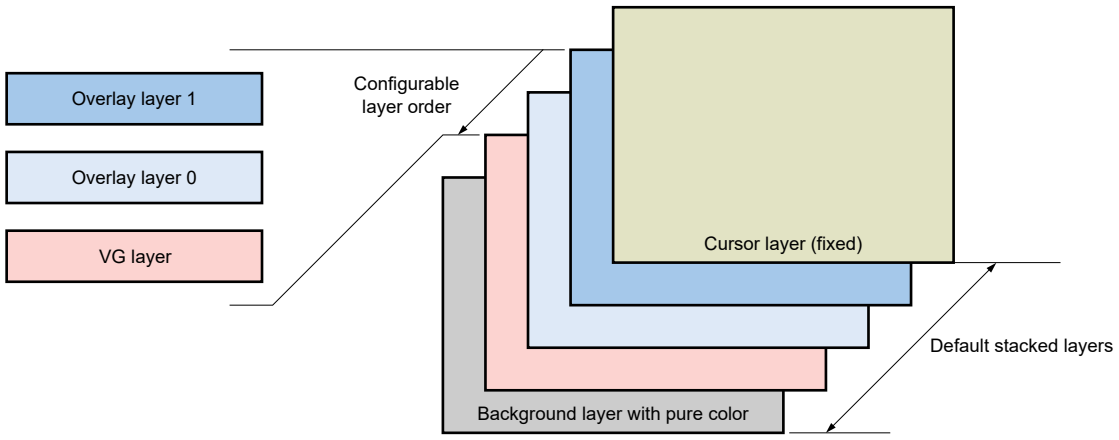


Figure 231. Default layer order for alpha blending

LCDIF performs blending from bottom to top, similar to the color-keying process. If any of the cursor, VG, or overlay layers are disabled, you must not apply alpha blending to the disabled layer. For details about the color-keying process, see Color-keying process.

Because the background layer with a pure color is used, the VG and overlay layers do not need to fill the whole screen. The VG and overlay layers can cover the screen only partially. To configure the necessary screen coordinates:

Table 547. Configuring screen coordinates

Step	Purpose	Programming
1	Specify the top-left coordinates.	Write the coordinates to: • Video Top Left Coordinate (VideoTL) • Overlay Top Left (OverlayTL) • Overlay Layer 1 Top Left (OverlayTL1)
2	Specify the layer size.	Write the size to: • Frame Buffer Size (FrameBufferSize) • Overlay Size (OverlaySize) • Overlay Layer 1 Size (OverlaySize1)
3	Verify that the bottom-right coordinates do not exceed the screen size.	Calculate the bottom-right coordinates from the values in steps 1 and 2.

55.3.3.1.1 Alpha blending algorithm

Figure 232 and Table 548 describe the stages in which LCDIF performs alpha blending.

Table 548. Alpha-blending stages

#	Stage	Description
1	Transparent or opaque conversion	The incoming alpha (independent of source or destination) can be inverted if the internal alpha rule must be followed. Internally, alpha = 0 means transparent and alpha = FFh means opaque. External content may follow the opposite rule. The output of the block is either: • `As` or `1-As` for source • `Ad` or `1-Ad` for destination
2	Global value substitution	You can use a global alpha value from a register to substitute or scale the incoming alpha. This allows any of the following operations for the incoming alpha `As`: • Pass through • `Ags` (global alpha) directly substitutes for `As` • The global alpha value (`As`×`Ags`) scales `As` The source and destination have distinct global alpha values.
3	Blending factor generation	LCDIF generates the blending factors for source and destination alpha. Depending on the blending mode, each alpha can be: • 0 • 1 • A • 1–A
4	Final blending	LCDIF implements the operations described in Alpha blending equations.

NOTE

See VideoAlphaBlendConfig, OverlayAlphaBlendConfig, and OverlayAlphaBlendConfig1 for more details on alpha blending settings.

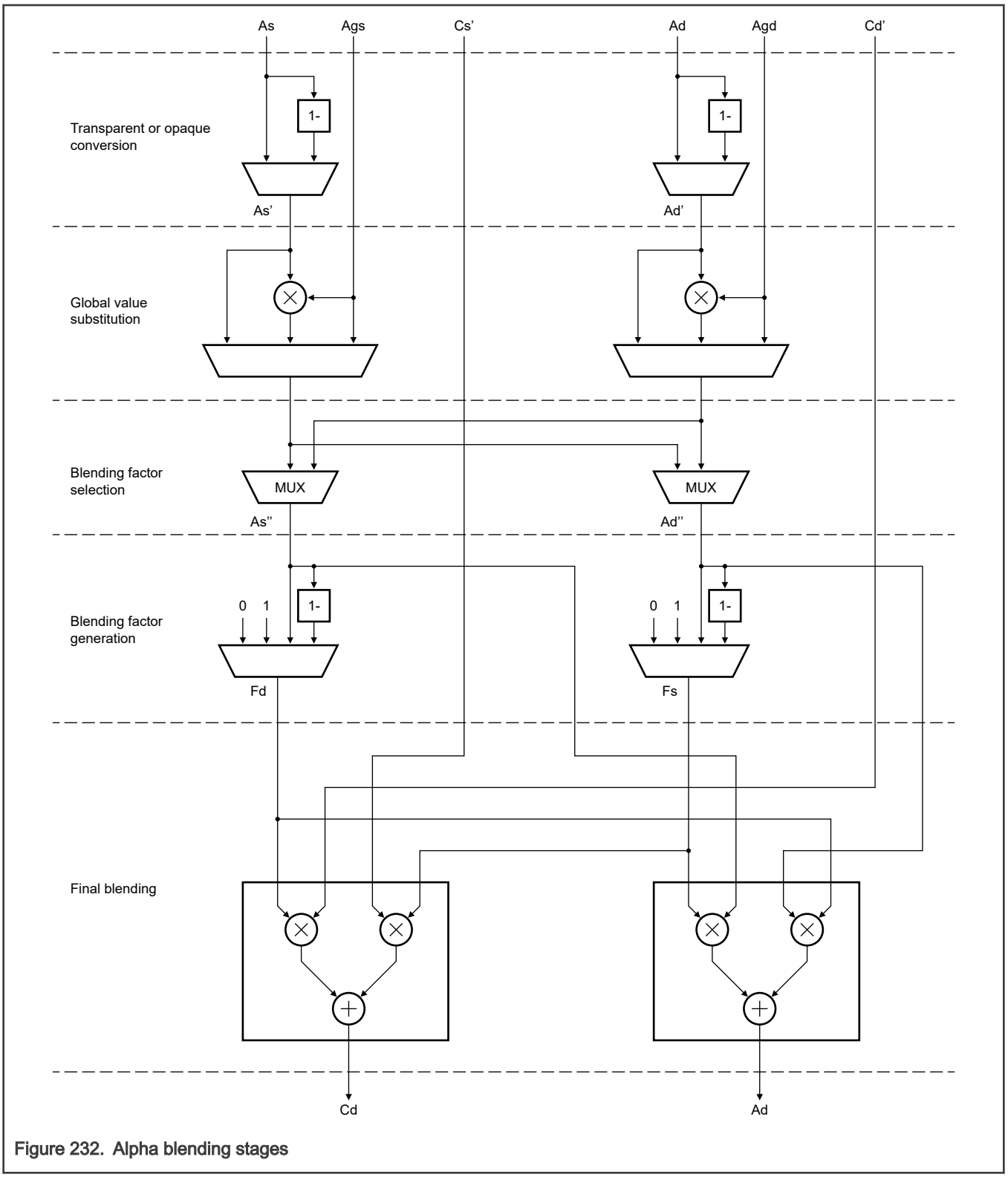


Figure 232. Alpha blending stages

55.3.3.1.2 Alpha blending equations

$$Cd = Fs \times Cs' + Fd \times Cd'$$

$$Ad = Fs \times As'' + Fd \times Ad''$$

Equation 6. Equations

Table 549. Terms

Term	Description
Cd	Destination color after blending
Fs	Source blending factor
Cs'	Source color component
Fd	Destination blending factor
Cd'	Destination color component
Ad	Destination alpha after blending
As''	Modified source alpha component
Ad''	Modified destination alpha component

55.3.3.2 Gamma correction

LCDIF provides gamma correction in the output panel. The gamma correction lookup tables (LUTs) have three separate sets—red, green, and blue. You configure these LUTs using the gamma-related registers.

If you describe gamma correction as LUT index or LUT entry data, you can set a gamma LUT sequentially based on color values, starting from the smallest value and ending at the largest value.

Use Gamma Index (GammaIndex0) and Gamma Data (GammaData0) to configure the LUT index and the LUT entry data, respectively.

If you want to perform gamma correction using the following sequence:

(0, 0)

(1, 2)

(2, 5)

(3, 6)

...

perform the following procedure:

Table 550. Gamma correction for example sequence

Step	Purpose	Programming
1	Specify that gamma correction starts from 0.	Write 0 to GammaIndex0[INDEX].
2	Fill the entire LUT.	Write 0, 2, 5, 6, ... for 256 times to Gamma Data (GammaData0).

55.3.3.3 Dither

In an image where color intensities change slowly, noticeable jumps in intensity levels may occur between pixels. You can use LCDIF's dither feature to diffuse the intensity across neighboring pixels. With this feature, LCDIF scans pixels in order and

distributes (that is, diffuses) errors in calculating a pixel's intensity to neighboring pixels. This keeps the overall intensity of the image closer to the input intensity.

55.3.3.3.1 Dither lookup table

The display controller requires a lookup table of 4 x 4 entries of 4 bits to implement dithering. This 64-bit table is located by the 32-bit registers DisplayDitherTableLow and DisplayDitherTableHigh. The dither lookup table needs to be indexed by the two least significant bits X[1:0] and Y[1:0] of the display controller.

55.3.3.3.2 Dither configuration

The following describes the dither configuration process with RedColor[3:0] as an example:

1. Create a lookup table by setting registers DisplayDitherTableLow and DisplayDitherTableHigh.
   - The lookup table includes 16 entries, 4 bits for each.
   - The lookup table provides a value U[3:0] through the indexes X[1:0] and Y[1:0].
2. Compare the color value RedColor[3:0] with this U[3:0].
   - If RedColor[3:0] > U[3:0] and RedColor[7:4] is not 4’b1111, then the final color value is NewRedColor = RedColor[7:4] + 1’b1.
   - If RedColor[3:0] is less than or equal to U[3:0], then NewRedColor = RedColor[7:4].

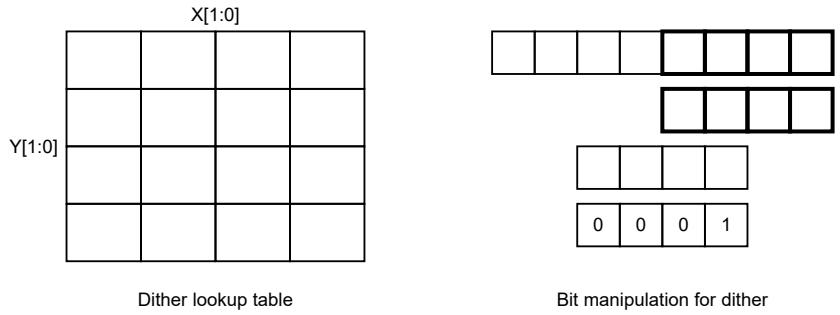


Figure 233. Dither diagram

55.3.4 Miscellaneous features

55.3.4.1 Video timing generation

Video timing generation (VTG) includes a timer corresponding to the pixel clock. Through the timer, VTG generates digital RGB output, including HSYNC, VSYNC, and Data Enable (DE). DE is asserted with Active display zone, and enables the Active display zone to be shown, as illustrated in the following figure.

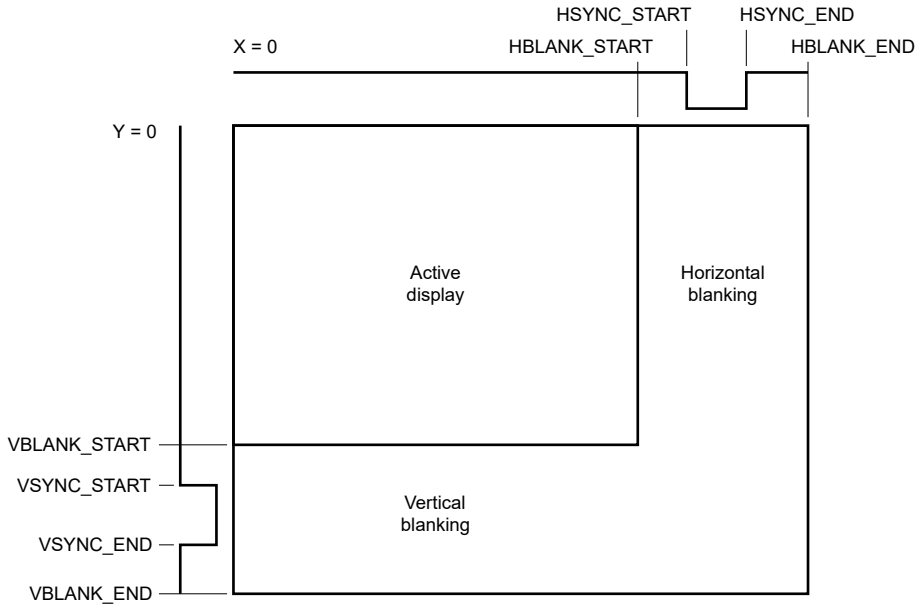


Figure 234. Timing diagram

55.3.4.2 Power management

LCDIF supports clock gating for:

- RAMs when no read/write operations are in progress.
- The VG and overlay layers when in Idle state.

Use DcControl[RAM_CLOCK_GATING] and the fields in Layer Clock Gating (LayerClockGate) to disable clock gating.

55.3.4.3 Double-buffered registers

Most LCDIF registers are double-buffered. Figure 235 shows the double-buffering mechanism:

- The AHB interface accesses shift registers and fields pertaining to set 0.
- The double-buffered registers and fields pertaining to set 1 output data for display use.
- The shift and double-buffered registers and fields share the same addresses.
- LCDIF transmits data from shift registers and fields to the double-buffered registers and fields at frame pulses.
- The shift registers and fields pertaining to set 0 are the only ones that you can read back.

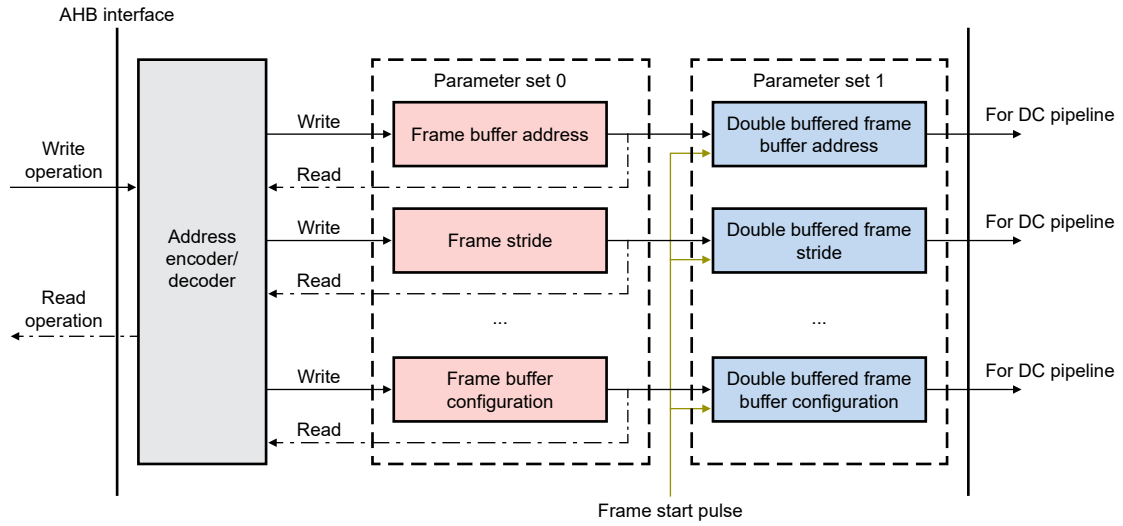


Figure 235. Double-buffered data flow

55.3.5 Clocking

LCDIF uses the following clock signals.

Table 551. Clock signals

Signal	Description	Input or output
ACLK	AXI bus clock or DC core clock	Input for DBI types A and B
HCLK	AHB interface clock	Input
pixelClk	Pixel clock from the display device	Usually input; also used as an output signal to the display

55.3.6 Reset

LCDIF uses the following reset signals to reset the AXI interface, AHB interface, and the display core either simultaneously or individually.

- ARESETn for AXI reset and display core: ensure no AXI transactions are in progress and all the display controller blocks are idle.
- HRESETn for AHB reset: ensure no AHB transactions are in progress.

Reset signals are synchronized to their corresponding clock domains—ACLK and HCLK. Their reset requirements include:

- During a reset, keep the corresponding reset signal asserted for a minimum of 32 slowest clock cycles.
- After deasserting the reset signal, wait for 128 slowest clock cycles before starting new activities.
- When the reset signals are not deasserted together, use the clock of the last reset (preferably the ACLK) for the 128-cycle wait. All deassertions except the last must comply with the 32-cycle assertion time.

55.3.7 Interrupts

The LCDIF interrupt signal is DC_INTR. It indicates LCDIF's interrupt events. When the system receives the DC_INTR signal, read Display Interrupt (DisplayIntr) to identify which of the following interrupt events occurred:

Table 552. Interrupts

Interrupt	Associated DisplayIntr field	LCDIF asserts this interrupt when:
Frame end	DISP0	One frame finishes displaying (see also Frame end interrupt)
DBI configuration error	DISP0_DBI_CFG_ERROR	The DBI command is illegal
Panel underflow	PANEL_UNDERFLOW	The data supply is insufficient
Soft reset done	SOFT_RESET_DONE	A soft reset is completed
AXI bus error	BUS_ERROR	An AXI bus transfer error occurs

55.3.7.1 Frame end interrupt

The following sequence illustrates the handling of a frame-end interrupt:

1. DC_INTR asserts at the last pixel of the visible area in a frame and remains asserted until you read DisplayIntr[DISP0].
2. Register configurations take effect.
3. After you have read DisplayIntr, DC_INTR deasserts automatically and the CPU configures the LCDIF registers—such as Frame Buffer Base Address (FrameBufferAddress0) and Cursor Base Address (CursorAddress)—for the next frame.
4. From the next frame, the LCDIF DMA starts fetching data from the system.

55.4 External signals

The following table lists the signals of LCDIF.

Table 553. External signals

Signal	Description	Direction
LCD_D[23:0]	Display color	OUT
LCD_VSYNC	Display vertical synchronization	OUT
LCD_HSYNC	Display vertical synchronization	OUT
LCD_DOTCLK	Display pixel clock	OUT
LCD_ENABLE	Display enable	OUT
LCD_DBI-D[15:0]	Information signal for DBI Type A or B	OUT
LCD_DBI_CSX_AB	Chip select for DBI Type A or B Display module is selected when low. It is set to 0 for DBI Type A in Clocked E mode.	OUT
LCD_DBI_DCX-AB	Data/Command indicator for DBI Type A or B Command is active when low.	OUT
LCD_DBI_WRX	Write signal for DBI Type B Host processor writes data at falling edge.	OUT
LCD_DBI_RWDX	For DBI Type A, specifies whether the current cycle is for read (1) or write (0) and set to 0.	OUT

Table continues on the next page...

Table 553. External signals (continued)

Signal	Description	Direction
	For DBI Type B, specifies whether the current cycle is for read (0) or write (1) and set to 1. For DBI Type A and Type B, read is not supported.	
LCD_DBI_E	E clock for DBI Type A It is set to 1 for DBI Type A in Fixed E mode.	OUT
LCD_DBI_DATA_OEN	Data output enable for DBI Type A or B	OUT

55.5 Initialization

55.5.1 DPI frame transfer action flow

The following provides the basic steps for setting the display output and frame data for DPI mode transfer.

1. Set the timing parameters according to the specifications for the target panel.
   - HDisplay0
   - HSync0
   - VDisplay0
   - VSync0
2. Set registers for the cursor, gamma, or dither function (if required).
   - Cursor*
   - Gamma*
   - DisplayDither*
   - PanelFunction[GAMMA]
   - PanelFunction[DITHER]
3. Set the frame buffer address, overlay address, stride, layer size, and location.
   - FrameBufferAddress0
   - OverlayAddress
   - OverlayAddress1
   - FrameBufferStride0
   - OverlayStride
   - OverlayStride1
   - DcTileUvFrameBufferAdr0
   - DcTileUvFrameBufferStr0
   - DcTileUvOverlayAdr
   - DcTileUvOverlayStr
   - VideoTL
   - FrameBufferSize

- OverlayTL
- OverlaySize
- OverlayTL1
- OverlaySize1

For tiled formats, any of the stride register values = One-line stride x tile height.

- FrameBufferStride0
- OverlayStride
- OverlayStride1
- DcTileUvFrameBufferStr0
- DcTileUvOverlayStr

4. Set the output bus mode to DPI.

   DbiConfig0[BUS_OUTPUT_SEL] = 0

5. Configure the panel and panel timing requirements.

   PanelConfig

6. Configure the data format of the frame buffer and enable the layer.
   - PanelFunction[OUTPUT] = 1 indicates output panel is enabled.
   - FrameBufferConfig0[ENABLE] = 1 indicates the video/graphic layer is enabled.
   - OverlayConfig[ENABLE] = 1 indicates overlay layer 0 is enabled.
   - OverlayConfig1[ENABLE] = 1 indicates overlay layer 1 is enabled.
   - Registers used to configure the tiled ARGB8888, YUY2, and NV12 formats:
     — DcTileIncfg
     — DcOverlayTileIncfg

     When the linear ARGB8888, ARGB1555, RGB565, ARGB4444 format, RGB888, or A8RGB565 is expected, the tiled format field of these registers must be set to NONE.
   - Register fields used to configure the linear ARGB8888, ARGB1555, RGB565, ARGB4444, RGB888, and A8RGB565 formats:
     — FrameBufferConfig0[FORMAT]
     — OverlayConfig[FORMAT]
     — OverlayConfig1[FORMAT]
7. Start the interface mode transfer.

   PanelWorking[WORKING] = 1

8. Update parameters for the next frame.

   PanelControl[VALID] =1 indicates the next frame configuration is completed.

9. Wait for the interrupt signal when the transfer is completed, or check the interrupt register status.
   - DisplayIntr
   - DisplayIntrEnable
10. (Optional) To write a single frame in DPI command mode, set DpiConfig0[DPI_COMMAND_MODE] = 1.
11. Repeat the preceding steps for each succeeding frame as needed.

55.5.2 DBI frame transfer action flow

The following sections provide the basic steps of setting the display output and frame data for DBI mode transfer in the action flows of initialization, refresh of panel frame buffer, and partial refresh of panel frame buffer.

Initialization

DBI initialization should be done after the panel is powered on. Figure 236 shows the DBI initialization timing:

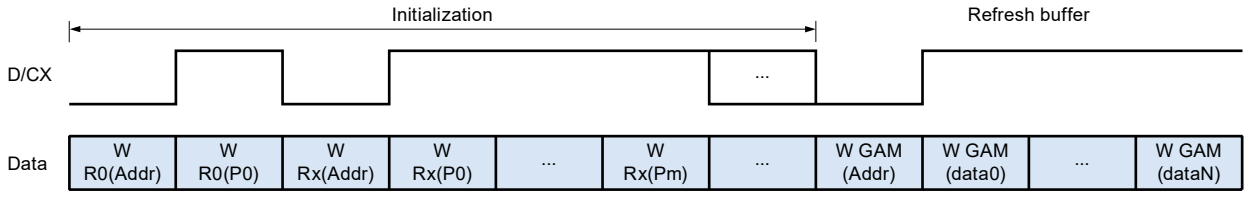


Figure 236. DBI initialization timing

1. Set PanelConfig[CLOCK] = 0. This setting is mandatory for the DBI mode.

   In DBI mode, the display controller ignores the values of any other bits in the PanelConfig register.

2. Set the output bus mode to DBI.

   DbiConfig0[BUS_OUTPUT_SEL] = 1

3. Set the DBI type for output.

   DbiConfig0[DBI_TYPE] = Type A Fixed E, Type A Clocked E, Type B.

4. Set the DBI bus in idle state.

   DbiIfReset[DBI_IF_LEVEL_RESET] = 1

5. Set the DBI output timing parameters and the AC timing specifications of the panel.

   Registers for DBI Type A or Type B:

   - DbiWrChar10
   - DbiWrChar20

6. Configure the interface output or input speed through the DbiConfig0[DBI_AC_TIME_UNIT] register field.
7. Write the DbiCmd register to send configuration data. Commands must be entered one by one.
   - DbiCmd0[DBI_COMMANDFLAG] = 0 indicates writing an address to the panel. The address to be sent to the DBI port is stored in DbiCmd0[DBI_CCOMMAND_WORD].
   - DbiCmd0[DBI_COMMANDFLAG] = 2 indicates writing one parameter to the panel. The parameter is stored in DbiCmd0[DBI_COMMAND_WORD].

Refresh of panel frame buffer (single frame example)

1. Set timing parameters according to the actual panel specifications. Parameters are set before the first frame refresh, and then are ignored after that.
   - HDisplay.TOTAL = Panel width
   - HDisplay.DISPLAY_END = Panel width
   - HSync.START = 0
   - HSync.END = Panel width
   - VDisplay.TOTAL = Panel height + 1

     The extra value 1 is used for waiting for the data FIFO to be filled. This extra value should be 2 for the tile data format.
   - VDisplay.DISPLAY_END = Panel height

- VSync.START = 0
- VSync.END = Panel height

2. Set the registers for the cursor, gamma, or dither function (if required).
   - Cursor*
   - Gamma*
   - DisplayDither*
   - PanelFunction.GAMMA
   - PanelFunction.DITHER
3. Set the frame buffer address or overlay address, stride, layer size, and location.
   - FrameBufferAddress
   - OverlayAddress
   - OverlayAddress1
   - FrameBufferStride
   - OverlayStride
   - OverlayStride1
   - DcTileUvFrameBufferAdr
   - DcTileUvFrameBufferStr
   - DcTileUvOverlayAdr
   - DcTileUvOverlayStr
   - VideoTL
   - FrameBufferSize
   - OverlayTL
   - OverlaySize
   - OverlayTL1
   - OverlaySize1

   For tiled formats, any of the following stride register values = One-line stride x Tile height:
   - FrameBufferStride
   - OverlayStride
   - OverlayStride1
   - DcTileUvFrameBufferStr
   - DcTileUvOverlayStr
4. Configure the data format of the frame buffer and enable the layer.
   - PanelFunction.OUTPUT = 1 indicates the output panel is enabled.
   - FrameBufferConfig.Enable = 1 indicates the video/graphic layer is enabled.
   - OverlayConfig.Enable = 1 indicates overlay layer 0 is enabled.
   - OverlayConfig1.Enable = 1 indicates overlay layer 1 is enabled.
   - Registers used to configure the tiled ARGB8888, YUY2, and NV12 formats: DcTileIncfg and DcOverlayTileIncfg

If the linear ARGB8888, ARGB1555, RGB565, ARGB4444, RGB888, or A8RGB565 format is expected, the tiled format fields of these registers must be set to NONE.

- Registers used to configure the linear ARGB8888, ARGB1555, RGB565, ARGB4444, RGB888, and A8RGB565 formats:
  - FrameBufferConfig
  - OverlayConfig
  - OverlayConfig1

5. Write the DbiCmd register to send configuration data. Commands are entered one by one. DbiCmd.DBI_COMMANDFLAG=1 indicates writing to the DBI bus. For this flag setting, the value of DbiCmd.DBI_COMMAND_WORD is ignored.
6. Wait for the interrupt signal when the write is completed, or check the interrupt register status.
   - DisplayIntr
   - DisplayIntrEnable
7. Repeat the preceding steps for each succeeding frame as needed.

Partial refresh of panel frame buffer (single frame example)

1. Set timing parameters according to the actual panel specifications. Parameters are set before the first frame refresh, and then ignored after that.
   - HDisplay.TOTAL = Partial window width
   - HDisplay.DISPLAY_END = Partial window width
   - HSync.START = Actual clipped window x start
   - HSync.END = Actual clipped window x end
   - VDisplay.TOTAL = Partial window height + 1

     The extra value 1 is used for waiting for the data FIFO to be filled. This extra value should be 2 for the tile data.
   - VDisplay.DISPLAY_END = Partial window height
   - VSync does not require setting for the linear format. For tiled data:
     - VSync.START = Actual clipped window y start
     - VSync.END = Actual clipped window y end
2. Set the registers for the cursor, gamma, or dither function (if required).
   - Cursor*
   - Gamma*
   - DisplayDither*
   - PanelFunction.GAMMA
   - PanelFunction.DITHER
3. Set the frame buffer address, overlay address, stride, layer size, and location.
   - FrameBufferAddress
   - OverlayAddress
   - OverlayAddress1
   - FrameBufferStride
   - OverlayStride

- OverlayStride1
- DcTileUvFrameBufferAdr
- DcTileUvFrameBufferStr
- DcTileUvOverlayAdr
- DcTileUvOverlayStr
- VideoTL
- FrameBufferSize
- OverlayTL
- OverlaySize
- OverlayTL1
- OverlaySize1

For tiled formats, any of the following stride register values = One-line stride x Tile height:

- FrameBufferStride
- OverlayStride
- OverlayStride1
- DcTileUvFrameBufferStr
- DcTileUvOverlayStr

4. Configure the data format of the frame buffer and enable the layer.
   - PanelFunction.OUTPUT = 1 indicates the output panel is enabled.
   - FrameBufferConfig.Enable = 1 indicates the video/graphic layer is enabled.
   - OverlayConfig.Enable = 1 indicates overlay layer 0 is enabled.
   - OverlayConfig1.Enable = 1 indicates overlay layer 1 is enabled.
   - Registers used to configure the tiled ARGB8888, YUY2, and NV12 formats: DcTileIncfg and DcOverlayTileIncfg

     If the linear ARGB8888, ARGB1555, RGB565, ARGB4444, RGB888, or A8RGB565 format is expected, the tiled format fields of these registers must be set to NONE.
   - Registers used to configure the linear ARGB8888, ARGB1555, RGB565, ARGB4444, RGB888, and A8RGB565 formats:
     - FrameBufferConfig
     - OverlayConfig
     - OverlayConfig1
5. Write the DbiCmd register to send configuration data for the partial window. Commands are entered one by one.

   DbiCmd.DBI_COMMANDFLAG=1 indicates writing to the DBI bus.

   For this flag setting, the value of DbiCmd.DBI_COMMAND_WORD is ignored.
6. Wait for the interrupt signal when the write is completed, or check the interrupt register status.
   - DisplayIntr
   - DisplayIntrEnable
7. Repeat the preceding steps for each succeeding frame as needed.

55.6 LCDIF register descriptions

55.6.1 LCDIF memory map

LCDIF base address: 4048_0000h

Offset	Register	Width (In bits)	Access	Reset value
2024h	Frame Buffer Configuration (FrameBufferConfig0)	32	RW	0000_0000h
2028h	Frame Buffer Base Address (FrameBufferAddress0)	32	RW	0000_0000h
202Ch	Frame Buffer Stride (FrameBufferStride0)	32	RW	0000_0000h
2034h	Tile Input Configuration (DcTileInCfg0)	32	RW	0000_0000h
2038h	Frame Buffer Tiled UV Base Address (DcTileUvFrameBufferAdr0)	32	RW	0000_0000h
203Ch	Frame Buffer Tiled UV Stride (DcTileUvFrameBufferStr0)	32	RW	0000_0000h
2058h	Frame Buffer Background (FrameBufferBackground)	32	RW	0000_0000h
205Ch	Frame Buffer Color Keying Start Address (FrameBufferColorKey)	32	RW	0000_0000h
2060h	Frame Buffer Color Key End Address (FrameBufferColorKeyHigh)	32	RW	0000_0000h
2064h	Frame Buffer Clear Value (FrameBufferClearValue)	32	RW	0000_0000h
2068h	Video Top Left Coordinate (VideoTL)	32	RW	0000_0000h
206Ch	Frame Buffer Size (FrameBufferSize)	32	RW	0000_0000h
2070h	Video Global Alpha (VideoGlobalAlpha)	32	RW	0000_0000h
2074h	Blend Stack Order (BlendStackOrder)	32	RW	0000_0000h
2078h	Alpha Blending Configuration (VideoAlphaBlendConfig)	32	RW	0000_0000h
207Ch	Overlay Configuration (OverlayConfig)	32	RW	0000_0000h
2080h	Overlay Address (OverlayAddress)	32	RW	0000_0000h
2084h	Overlay Stride (OverlayStride)	32	RW	0000_0000h
2088h	Tile Input Configuration (DcOverlayTileInCfg)	32	RW	0000_0000h
208Ch	Tile Uv Overlay Address (DcTileUvOverlayAdr)	32	RW	0000_0000h
2090h	Tile Uv Overlay Stride (DcTileUvOverlayStr)	32	RW	0000_0000h
2094h	Overlay Top Left (OverlayTL)	32	RW	0000_0000h
2098h	Overlay Size (OverlaySize)	32	RW	0000_0000h
209Ch	Overlay Color Keying Start Address (OverlayColorKey)	32	RW	0000_0000h
20A0h	Overlay Color Keying End Address (OverlayColorKeyHigh)	32	RW	0000_0000h
20A4h	Overlay Alpha Blending Configuration (OverlayAlphaBlendConfig)	32	RW	0000_0000h
20A8h	Overlay Global Alpha (OverlayGlobalAlpha)	32	RW	0000_0000h
20ACh	Overlay Clear Value (OverlayClearValue)	32	RW	0000_0000h
20B0h	Overlay Layer 1 Configuration (OverlayConfig1)	32	RW	0000_0000h
20B4h	Overlay Layer 1 Address (OverlayAddress1)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
20B8h	Overlay Layer 1 Stride (OverlayStride1)	32	RW	0000_0000h
20BCh	Overlay Layer 1 Top Left (OverlayTL1)	32	RW	0000_0000h
20C0h	Overlay Layer 1 Size (OverlaySize1)	32	RW	0000_0000h
20C4h	Overlay Layer 1 Color Keying Start Address (OverlayColorKey1)	32	RW	0000_0000h
20C8h	Overlay Layer 1 Color Keying End Address (OverlayColorKeyHigh1)	32	RW	0000_0000h
20CCh	Overlay Layer 1 Alpha Blending Configuration (OverlayAlphaBlendConfig1)	32	RW	0000_0000h
20D0h	Overlay Layer 1 Global Alpha (OverlayGlobalAlpha1)	32	RW	0000_0000h
20D4h	Overlay Layer 1 Clear Value (OverlayClearValue1)	32	RW	0000_0000h
20E0h	Dither Lookup Table Low Threshold (DisplayDitherTableLow0)	32	RW	0000_0000h
20E4h	Dither Lookup Table High Threshold (DisplayDitherTableHigh0)	32	RW	0000_0000h
20E8h	Panel Configuration (PanelConfig0)	32	RW	0000_0000h
20ECh	Panel Control (PanelControl)	32	RW	0000_0000h
20F0h	Panel Function (PanelFunction)	32	RW	0000_0000h
20F4h	Panel Working (PanelWorking)	32	W	0000_0000h
20F8h	Panel State (PanelState)	32	R	0000_0000h
2100h	Horizontal Display Total and Visible Pixel Count (HDisplay0)	32	RW	0000_0000h
2104h	Horizontal Sync Counter (HSync0)	32	RW	0000_0000h
2110h	Vertical Total and Visible Pixel Count (VDisplay0)	32	RW	0000_0000h
2114h	Vertical Sync Counter (VSync0)	32	RW	0000_0000h
2118h	Display Current Location (DisplayCurrentLocation0)	32	R	0000_0000h
211Ch	Gamma Index (GammaIndex0)	32	W	0000_0000h
2120h	Gamma Data (GammaData0)	32	W	0000_0000h
2124h	Cursor Configuration (CursorConfig)	32	RW	0000_0000h
2128h	Cursor Base Address (CursorAddress)	32	RW	0000_0000h
212Ch	Cursor Location (CursorLocation)	32	RW	0000_0000h
2130h	Cursor Background Color (CursorBackground)	32	RW	0000_0000h
2134h	Cursor Foreground Color (CursorForeground)	32	RW	0000_0000h
2138h	Display Interrupt (DisplayIntr)	32	R	0000_0000h
213Ch	Display Interrupt Enable (DisplayIntrEnable)	32	RW	4000_0000h
2140h	DBI Configuration (DbiConfig0)	32	RW	0000_0040h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
2144h	DBI Interface Reset (DbiIfReset0)	32	W	0000_0000h
2148h	DBI Write AC Characteristics 1 (DbiWrChar10)	32	RW	0000_0000h
214Ch	DBI Write AC Characteristics 2 (DbiWrChar20)	32	RW	0000_0000h
2150h	DBI Command Control (DbiCmd0)	32	W	0000_0000h
2154h	DPI Configuration (DpiConfig0)	32	RW	0000_0005h
2160h	Source Endian Configuration (SrcConfigEndian)	32	RW	0000_0000h
2164h	Soft Reset (SoftReset)	32	W	0000_0000h
2168h	Display Controller Control (DcControl)	32	RW	0000_0000h
21A0h	Layer Clock Gating (LayerClockGate)	32	RW	0000_0000h

55.6.2 Frame Buffer Configuration (FrameBufferConfig0)

Offset

Register	Offset
FrameBufferConfig0	2024h

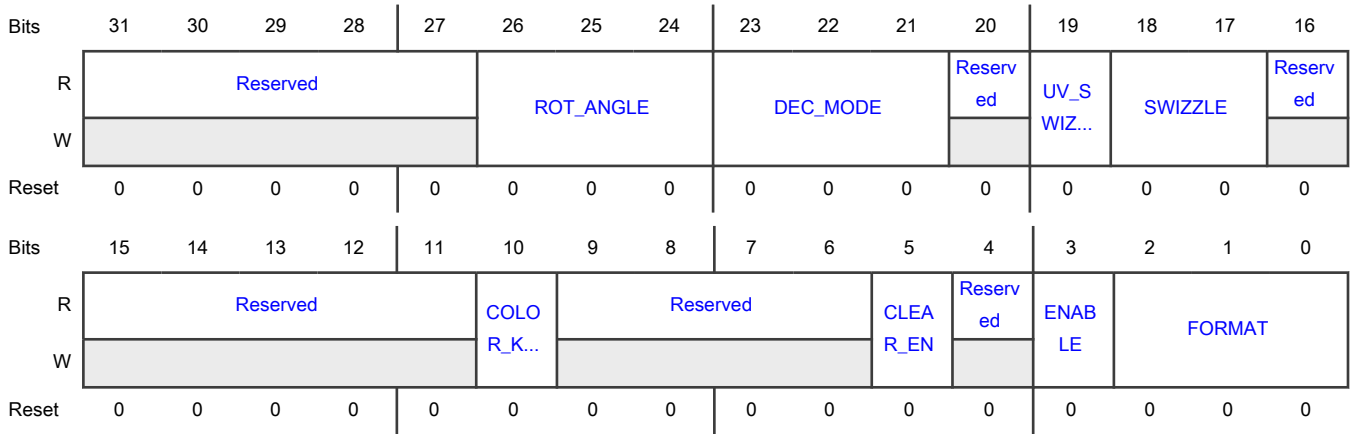
Function

Controls the frame buffer and timing.

NOTE

This register is double-buffered.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-24 ROT_ANGLE	Rotation Angle Specifies the rotation angle for the source surface. FrameBufferSize[HEIGHT] specifies the height of the surface in pixels. 000b - Sets the rotation angle to 0 degree. 010b - Sets the rotation angle to 180 degree. 100b - Flips horizontal 101b - Flips vertical All other values are reserved.
23-21 DEC_MODE	Decompression Mode Specifies the video decompression mode. 000b - Decompression disabled 001b - Non-Subsample 010b - Horizontal Subsample 011b - Horizontal and Vertical Subsample 100b - ETC2 All other values are reserved.
20 —	Reserved
19 UV_SWIZZLE	Swizzle for UV Reshapes the input data if U is not in the lower bits and V is not in the upper bits.
18-17 SWIZZLE	Swizzle for RGB Specifies the input format of a pixel for swizzling at the video, graphic, and overlay layers to switch the component position in the pixel data. 00b - ARGB 01b - RGBA 10b - ABGR 11b - BGRA
16-11 —	Reserved
10	Enable Color Keying

Table continues on the next page...

Table continued from the previous page...

Field	Function
COLOR_KEY_EN	0b - Disable 1b - Enable
9-6 —	Reserved
5 CLEAR_EN	Enable Clear 0b - Disable 1b - Enable
4 —	Reserved
3 ENABLE	Enable Layer Enables this layer. 0b - Disable 1b - Enable
2-0 FORMAT	Frame Buffer Format Specifies the format of the frame buffer. 000b - None; no frame buffer exists and LCDIF does not produce any output 001b - A4R4G4B4 010b - A1R5G5B5 011b - R5G6B5 100b - A8R8G8B8 101b - R8G8B8 110b - A8R5G6B5 All other values are reserved.

55.6.3 Frame Buffer Base Address (FrameBufferAddress0)

Offset

Register	Offset
FrameBufferAddress0	2028h

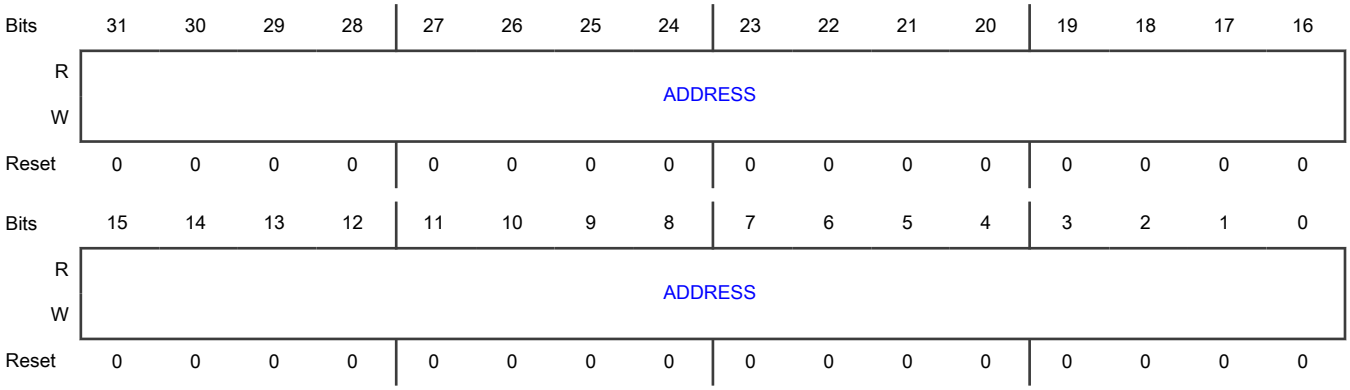
Function

Specifies the starting address of the frame buffer.

NOTE

This register is double-buffered.

Diagram



Fields

Field	Function
31-0 ADDRESS	Starting Address Specifies a 32-bit starting address.

55.6.4 Frame Buffer Stride (FrameBufferStride0)

Offset

Register	Offset
FrameBufferStride0	202Ch

Function

Specifies the stride of the frame buffer, in bytes.

NOTE

This register is double-buffered.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															STRIDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STRIDE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-17 —	Reserved
16-0 STRIDE	Frame Buffer Stride Specifies the number of bytes from the start of one line to the next line.

55.6.5 Tile Input Configuration (DcTileInCfg0)

Offset

Register	Offset
DcTileInCfg0	2034h

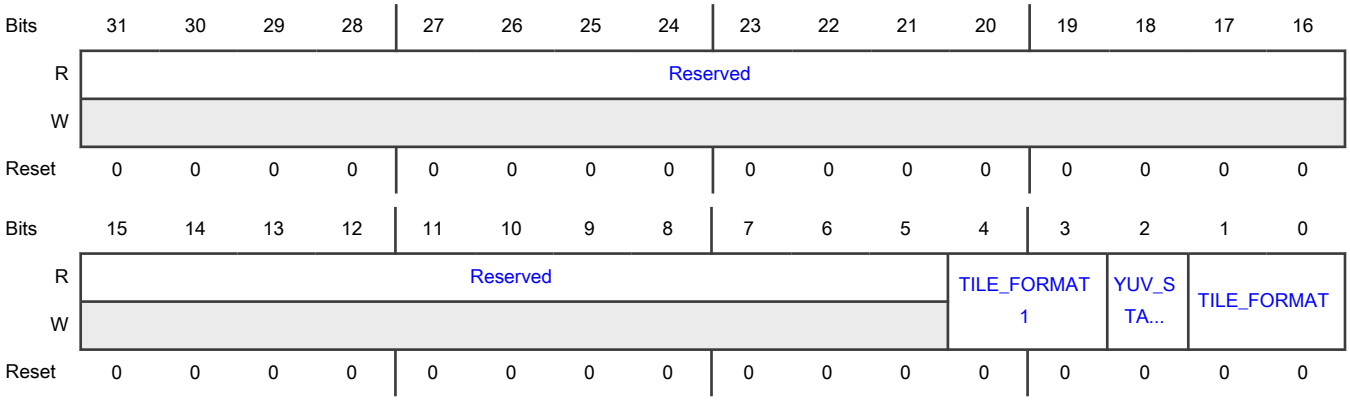
Function

Configures the tile input.

NOTE

This register is double-buffered.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-3 TILE_FORMAT 1	Tile Format 1 Selects high bit of tile format. • {Tile_Format1,Tile_Format} = 0x4, RGB888 • {Tile_Format1,Tile_Format} = 0x5 A8RGB565 **NOTE** This field is double buffered. It needs to be combined with Tile_Format in the form of {Tile_Format1, Tile_Format}
2 YUV_STANDARD	YUV Standard Specifies the YUV standard. 0b - BT601 1b - BT709
1-0 TILE_FORMAT	Tile Input Format Specifies the tile input format. 00b - No tile input. See DcTileInCfg0[TILE_FORMAT1]. 01b - ARGB8888 10b - YUY2 11b - NV12

55.6.6 Frame Buffer Tiled UV Base Address (DcTileUvFrameBufferAdr0)

Offset

Register	Offset
DcTileUvFrameBufferAdr0	2038h

Function

Specifies the UV frame buffer address when tile input is used.

NOTE

This register is double-buffered.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADDRESS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ADDRESS	Base Address

55.6.7 Frame Buffer Tiled UV Stride (DcTileUvFrameBufferStr0)

Offset

Register	Offset
DcTileUvFrameBufferStr0	203Ch

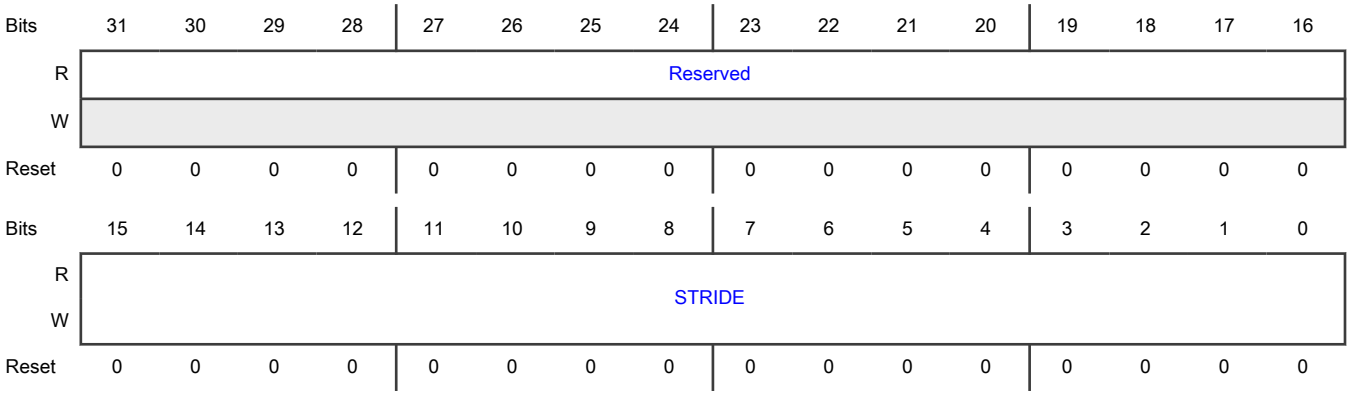
Function

Specifies the stride of the UV frame buffer when tile input is used.

NOTE

This register is double-buffered.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 STRIDE	Stride

55.6.8 Frame Buffer Background (FrameBufferBackground)

Offset

Register	Offset
FrameBufferBackground	2058h

Function

Sets the background color of frame buffer.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ALPHA								RED							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GREEN								BLUE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 ALPHA	Alpha Alpha value **NOTE** This field is double buffered.
23-16 RED	Red Red color **NOTE** This field is double buffered.
15-8 GREEN	Green Green color **NOTE** This field is double buffered.
7-0 BLUE	Blue Blue color **NOTE** This field is double buffered.

55.6.9 Frame Buffer Color Keying Start Address (FrameBufferColorKey)

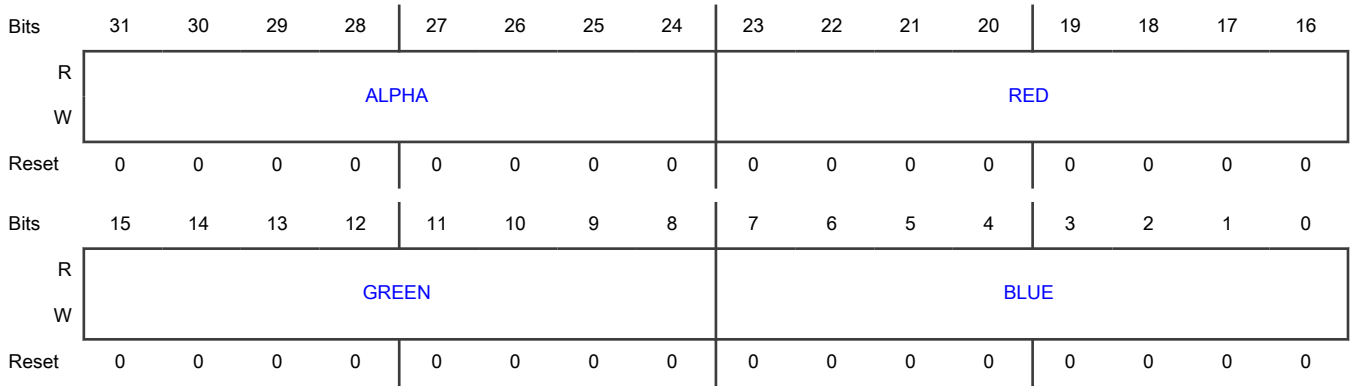
Offset

Register	Offset
FrameBufferColorKey	205Ch

Function

Sets the start address of the color keying range of frame buffer.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.10 Frame Buffer Color Key End Address (FrameBufferColorKeyHigh)

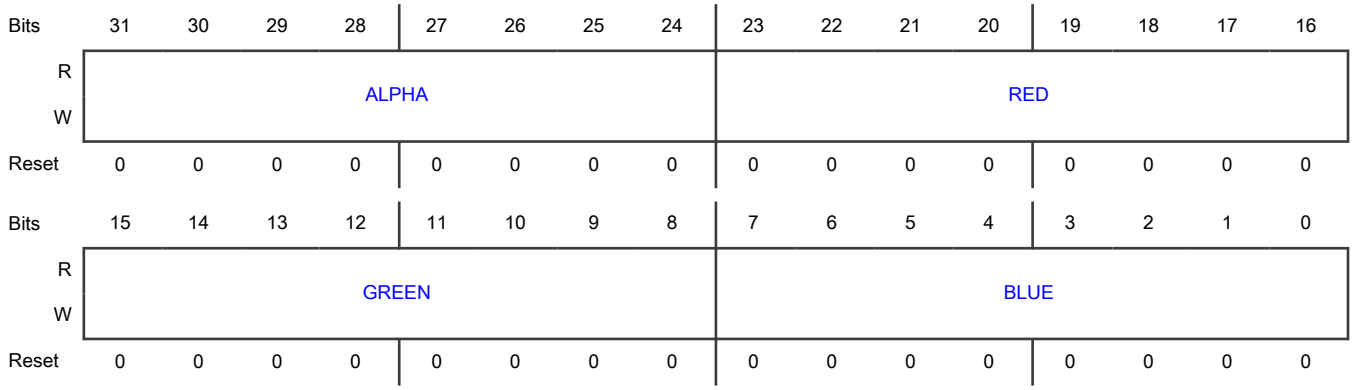
Offset

Register	Offset
FrameBufferColorKeyHigh	2060h

Function

Sets the end address of the color key range of frame buffer.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.11 Frame Buffer Clear Value (FrameBufferClearValue)

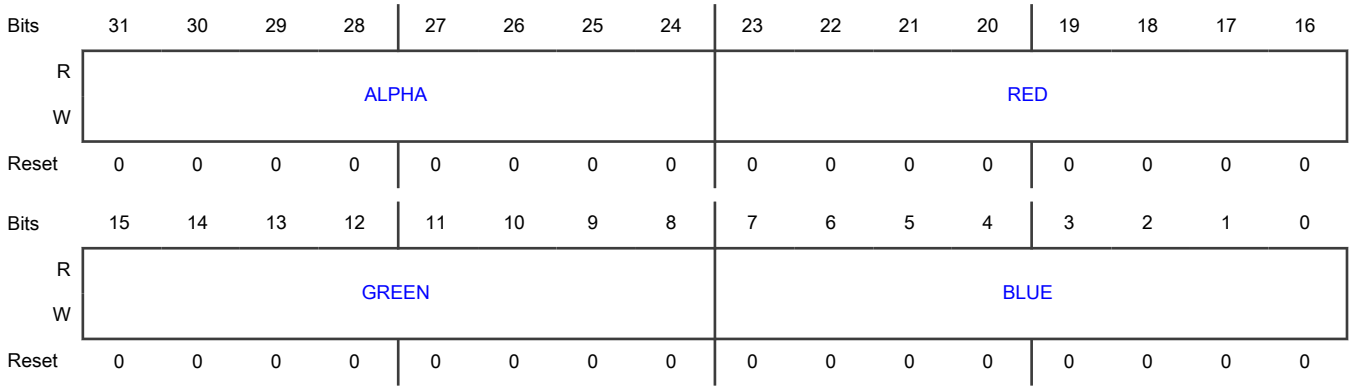
Offset

Register	Offset
FrameBufferClearValue	2064h

Function

Sets the clear value when FrameBufferConfig0[CLEAR_EN] is enabled in the format of A8R8G8B8.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue

55.6.12 Video Top Left Coordinate (VideoTL)

Offset

Register	Offset
VideoTL	2068h

Function

Sets the top left coordinate of panel pixel where the video must start. Note that there is no panning inside the video.

NOTE

This register is double buffered.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 Y	Top Boundary Top boundary of video window.
15-12 —	Reserved
11-0 X	Left Boundary Left boundary of video window

55.6.13 Frame Buffer Size (FrameBufferSize)

Offset

Register	Offset
FrameBufferSize	206Ch

Function

Sets the video size information of frame buffer.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				HEIGHT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				WIDTH											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 HEIGHT	Video Height Video height of frame buffer **NOTE** This field is double buffered.
15-12 —	Reserved
11-0 WIDTH	Video Width Video width of frame buffer **NOTE** This field is double buffered.

55.6.14 Video Global Alpha (VideoGlobalAlpha)

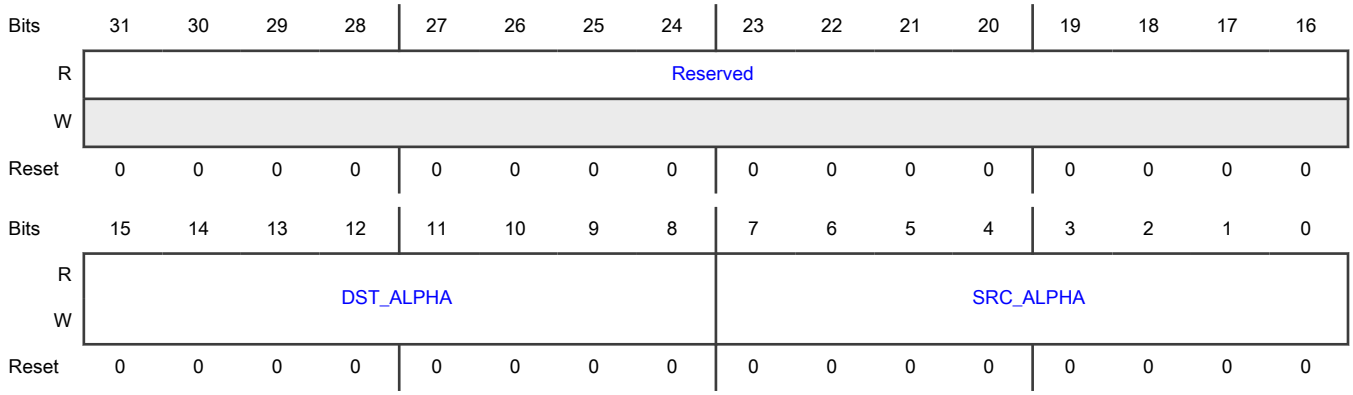
Offset

Register	Offset
VideoGlobalAlpha	2070h

Function

Sets the global alpha information for the video of frame buffer.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DST_ALPHA	Destination Alpha Sets the destination alpha for video. **NOTE** This field is double buffered.
7-0 SRC_ALPHA	Source Alpha Sets the source alpha for video. **NOTE** This field is double buffered.

55.6.15 Blend Stack Order (BlendStackOrder)

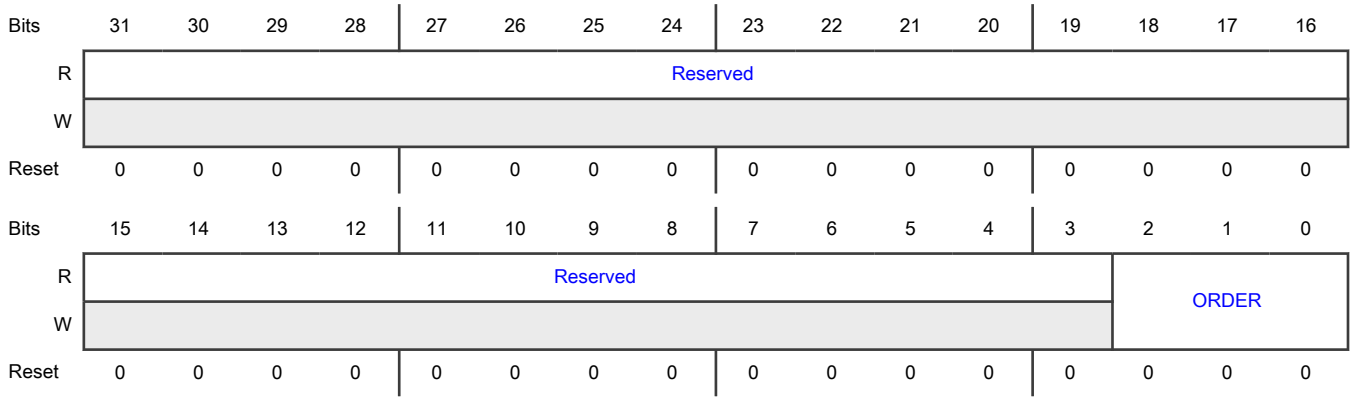
Offset

Register	Offset
BlendStackOrder	2074h

Function

Sets the order of VG, overlay 0, and overlay 1 layers for blending, in which the first layer is the lowest layer, the second layer is the middle layer, and the third layer is the highest layer.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 ORDER	Stack Order Stack order of video, overlay 0, and overlay 1 layers for blending. **NOTE** This field is double buffered. 000b - In the order of VG, Overlay 0 and Overlay 1 layers. 001b - In the order of VG, Overlay 1 and Overlay 0 layers. 010b - In the order of Overlay 0, VG and Overlay 1 layers. 011b - In the order of Overlay 0, Overlay 1, and VG layers. 100b - In the order of Overlay 1, VG and Overlay 0 layers. 101b - In the order of Overlay 1, Overlay 0, and VG layers.

55.6.16 Alpha Blending Configuration (VideoAlphaBlendConfig)

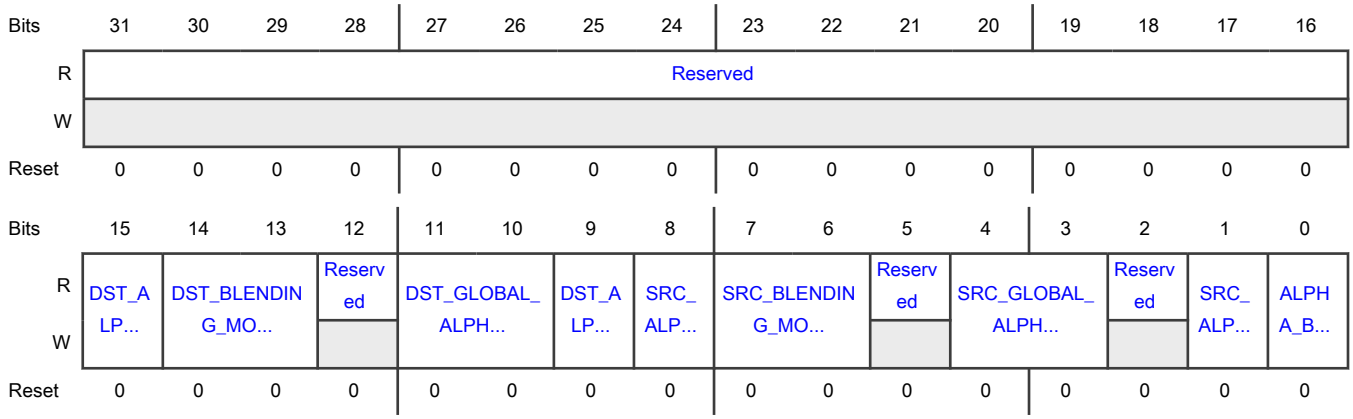
Offset

Register	Offset
VideoAlphaBlendConfig	2078h

Function

The display controller supports alpha blending for the cursor, VG, and overlay layers, together with one background layer.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 DST_ALPHA_FACTOR	Destination Alpha Factor Generates the blending factor for destination alpha. **NOTE** This field is double buffered. 0b - Do not generate the blending factor for destination alpha. 1b - Generates the blending factor for destination alpha.
14-13 DST_BLENDING_MODE	Destination Alpha Blending Mode Sets the blending mode for destination alpha. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the destination color 11b - Not affected by the color of the underlying pixel
12 —	Reserved
11-10 DST_GLOBAL_ALPHA_MODE	Destination Global Alpha Mode Sets the global destination alpha mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field is double buffered. 00b - Normal destination alpha mode 01b - Global destination alpha mode 10b - Scaled (by the global alpha value) destination alpha mode
9 DST_ALPHA_MODE	Destination Alpha Mode Sets the destination alpha mode. **NOTE** This field is double buffered. 0b - Normal destination alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
8 SRC_ALPHA_FACTOR	Source Alpha Factor Generates the blending factor for source alpha. **NOTE** This field is double buffered. 0b - Do not generate the blending factor. 1b - Generates the blending factor.
7-6 SRC_BLENDING_MODE	Source Blending Mode Sets the blending mode for source alpha. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the source color 11b - Not affected by the color of the underlying pixel
5 —	Reserved
4-3 SRC_GLOBAL_ALPHA_MODE	Source Global Alpha Mode Sets the global alpha mode to substitute or scale the source alpha. **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Normal source alpha mode 01b - Global source alpha mode 10b - Scaled (by the global alpha value) source alpha mode
2 —	Reserved
1 SRC_ALPHA_MODE	Source Alpha Mode Sets the source alpha mode for blending. **NOTE** This field is double buffered. 0b - Normal source alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
0 ALPHA_BLEND	Alpha Blending Enables alpha blending for video and graphics from the bottom layer to the top layer. If any of the cursor, VG, or overlay layers is disabled, alpha blending is not applied to the disabled layer. **NOTE** This field is double buffered. 0b - Disables alpha blending for video. 1b - Enables alpha blending for video.

55.6.17 Overlay Configuration (OverlayConfig)

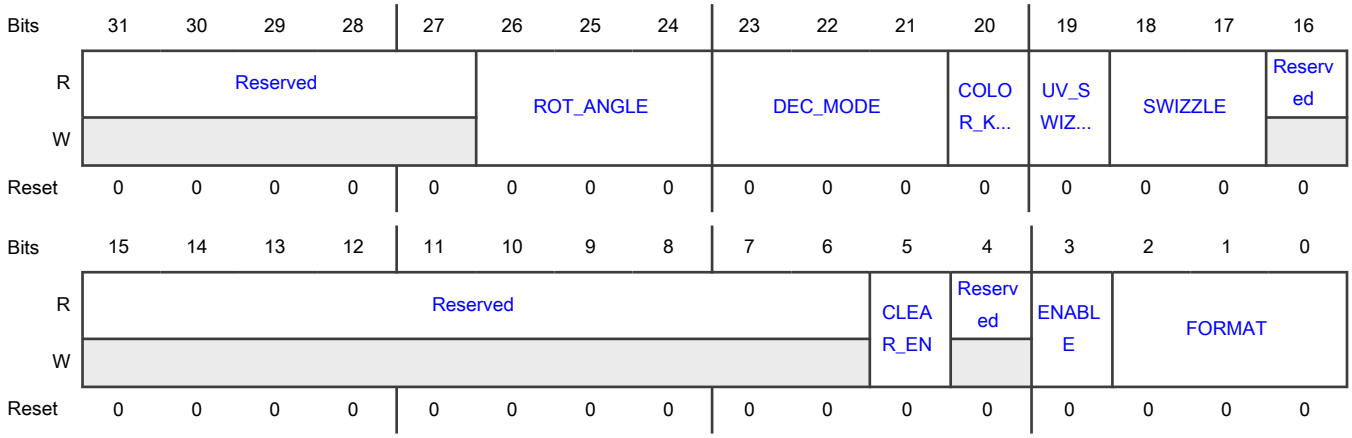
Offset

Register	Offset
OverlayConfig	207Ch

Function

Sets overlay control.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-24 ROT_ANGLE	Rotation Angle Specifies the rotation angle or flip for the source surface. Height field specifies the height of the surface in pixels. **NOTE** This field is double buffered. 000b - Sets the rotation angle to 0 degree. 010b - Sets the rotation angle to 180 degree. 100b - Flips horizontal 101b - Flips vertical All other values are reserved.
23-21 DEC_MODE	Decompression Mode Sets decompression mode on 0verlay 0. **NOTE** This field is double buffered. 000b - Disables decompression. 001b - Non Subsample. Enables non_subsample decompression mode. 010b - Horizontal Subsample. Enables horizontal subsample decompression mode. 011b - Horizontal Vertical Subsample. Enables horizontal and vertical subsample decompression mode. 100b - ETC2. Enables ETC2 decompression mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
20 COLOR_KEY_EN	Color Keying Enables or disables color keying. **NOTE** This field is double buffered. 0b - Disables color keying. 1b - Enables color keying.
19 UV_SWIZZLE	UV Swizzle Swizzles an input pixel for UV formats at the VG and overlay layers to align the component order. **NOTE** This field is double buffered.
18-17 SWIZZLE	RGB Swizzle Swizzles an input pixel for RGB formats at the VG and overlay layers to align the component order. **NOTE** This field is double buffered. 00b - Input pixel in ARGB format 01b - Input pixel in RGBA format 10b - Input pixel in ABGR format 11b - Input pixel in BGRA format
16-6 —	Reserved
5 CLEAR_EN	Clear Enables or disables clearing the overlay layer. **NOTE** This field is double buffered. 0b - Disables clearing the overlay layer. 1b - Enables clearing the overlay layer.
4 —	Reserved
3 ENABLE	Enable Enables or disables this overlay layer.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field is double buffered. 0b - Disables this overlay layer. 1b - Enables this overlay layer.
2-0 FORMAT	Overlay Format Sets the overlay format. None means there is no overlay. **NOTE** This field is double buffered. 000b - No overlay 001b - Overlay format is A4R4G4B4. 010b - Overlay format is A1R5G5B5. 011b - Overlay format is R5G6B5. 100b - Overlay format is A8R8G8B8. 101b - Overlay format is A8R5G6B5.

55.6.18 Overlay Address (OverlayAddress)

Offset

Register	Offset
OverlayAddress	2080h

Function

Sets the starting address of the overlay.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ADDRESS	Starting Address A 32-bit address.

55.6.19 Overlay Stride (OverlayStride)

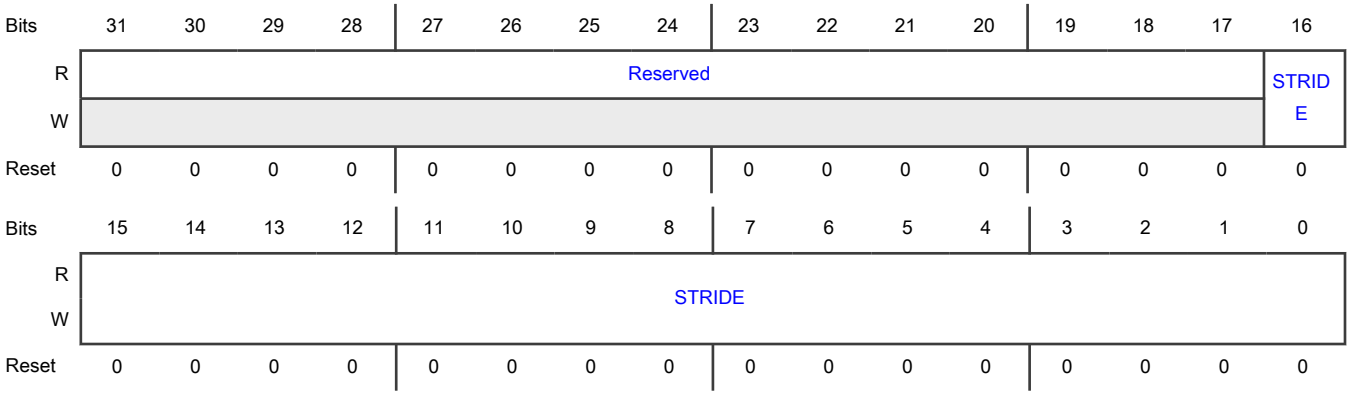
Offset

Register	Offset
OverlayStride	2084h

Function

Sets the stride of the overlay in bytes.

Diagram



Fields

Field	Function
31-17 —	Reserved
16-0 STRIDE	Stride Number of bytes from start of one line to next line.

55.6.20 Tile Input Configuration (DcOverlayTileInCfg)

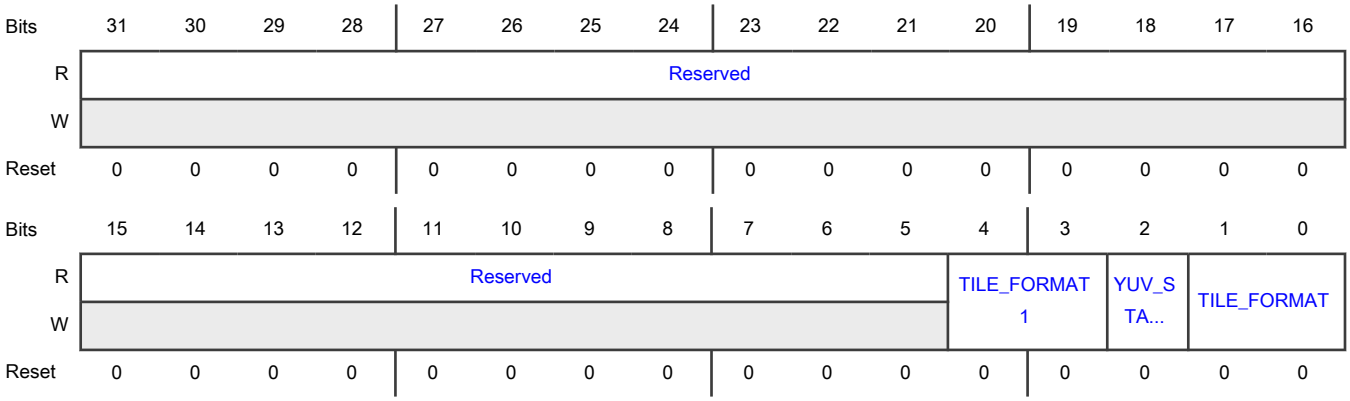
Offset

Register	Offset
DcOverlayTileInCfg	2088h

Function

Sets the tile input format for the overlay.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-3 TILE_FORMAT 1	Tile Format 1 Selects high bit of tile format. • {Tile_Format1,Tile_Format} = 0x4, RGB888 • {Tile_Format1,Tile_Format} = 0x5 A8RGB565 **NOTE** This field is double buffered. It needs to be combined with Tile_Format in the form of {Tile_Format1, Tile_Format}
2 YUV_STANDARD	YUV Standard Selects YUV standard. **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Selects the YUV standard BT601. 1b - Selects the YUV standard BT709.
1-0 TILE_FORMAT	Tile Input Format Sets the tile input format. **NOTE** This field is double buffered. 00b - No tile input. See DcOverlayTileInCfg[TILE_FORMAT1]. 01b - Sets the tile input format to ARGB8888. 10b - Sets the tile input format to YUY2. 11b - Sets the tile input format to NV12.

55.6.21 Tile Uv Overlay Address (DcTileUvOverlayAdr)

Offset

Register	Offset
DcTileUvOverlayAdr	208Ch

Function

Sets the starting overlay address of the tiled UV plane.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ADDRESS	Starting Overlay Address A 32-bit address.

55.6.22 Tile Uv Overlay Stride (DcTileUvOverlayStr)

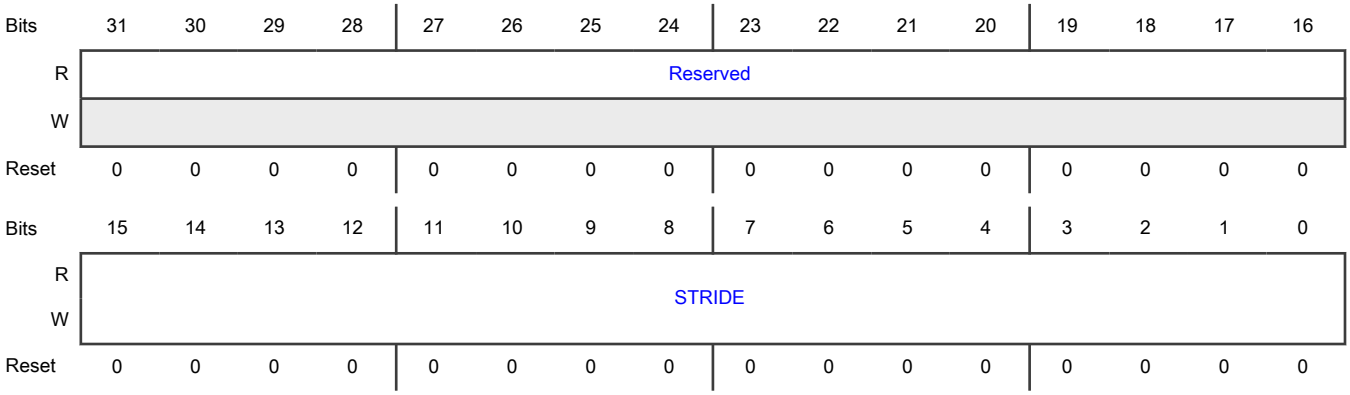
Offset

Register	Offset
DcTileUvOverlayStr	2090h

Function

Sets the stride of the overlay of the tiled UV plane in bytes.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 STRIDE	Overlay Stride Number of bytes from start of one line to next line. **NOTE** This field is double buffered.

55.6.23 Overlay Top Left (OverlayTL)

Offset

Register	Offset
OverlayTL	2094h

Function

Sets the top left coordinate of panel pixel where the overlay starts. Note that there is no panning inside the overlay.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 Y	Top Boundary Sets the top boundary of overlay window.
15-12 —	Reserved
11-0 X	Left Boundary Sets the left boundary of overlay window.

55.6.24 Overlay Size (OverlaySize)

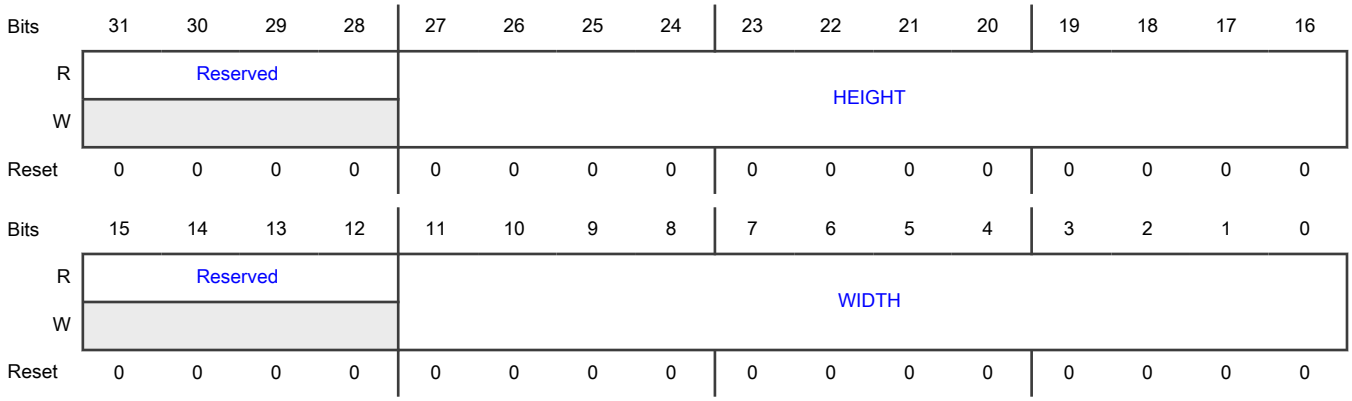
Offset

Register	Offset
OverlaySize	2098h

Function

Sets the overlay size.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-16 HEIGHT	Height Sets the overlay height.
15-12 —	Reserved
11-0 WIDTH	Width Sets the overlay width.

55.6.25 Overlay Color Keying Start Address (OverlayColorKey)

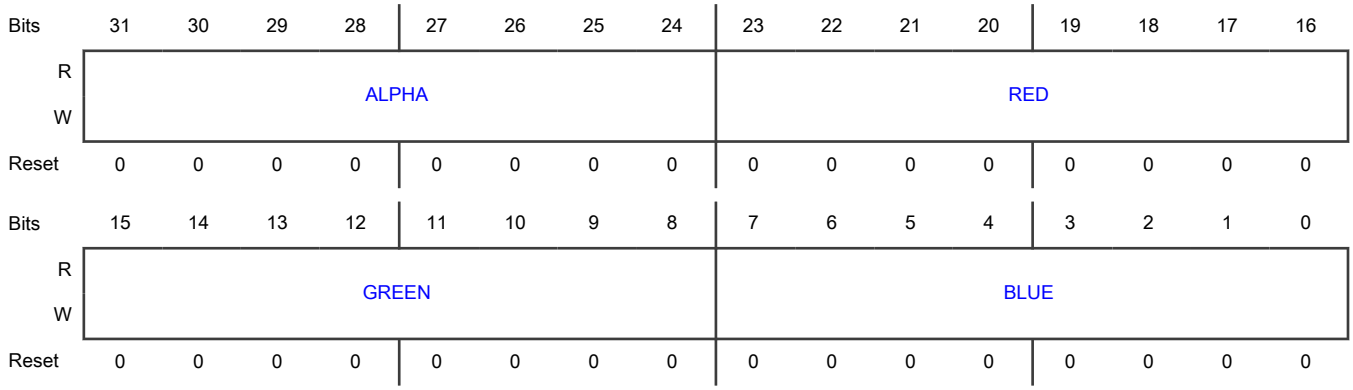
Offset

Register	Offset
OverlayColorKey	209Ch

Function

Sets the start of color keying range for overlay.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.26 Overlay Color Keying End Address (OverlayColorKeyHigh)

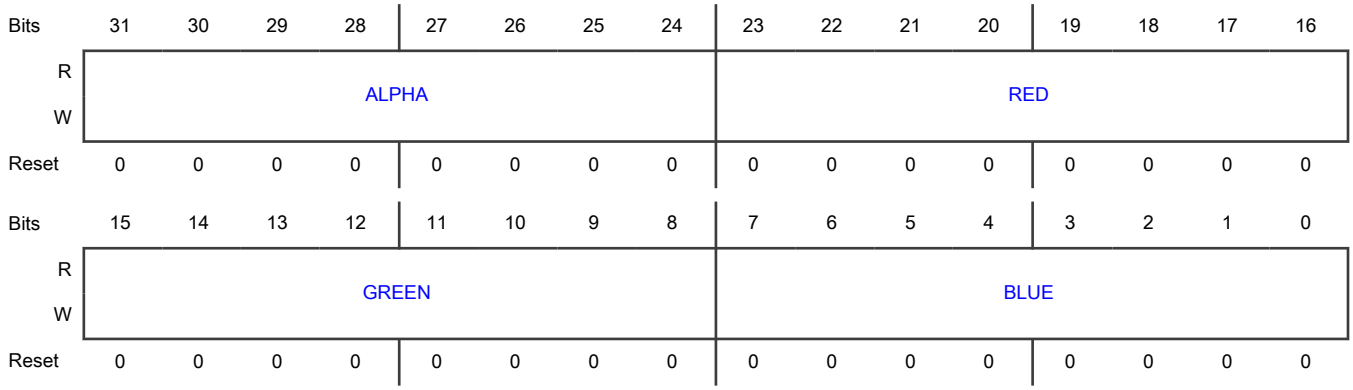
Offset

Register	Offset
OverlayColorKeyHigh	20A0h

Function

Sets the end of color keying range for overlay.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.27 Overlay Alpha Blending Configuration (OverlayAlphaBlendConfig)

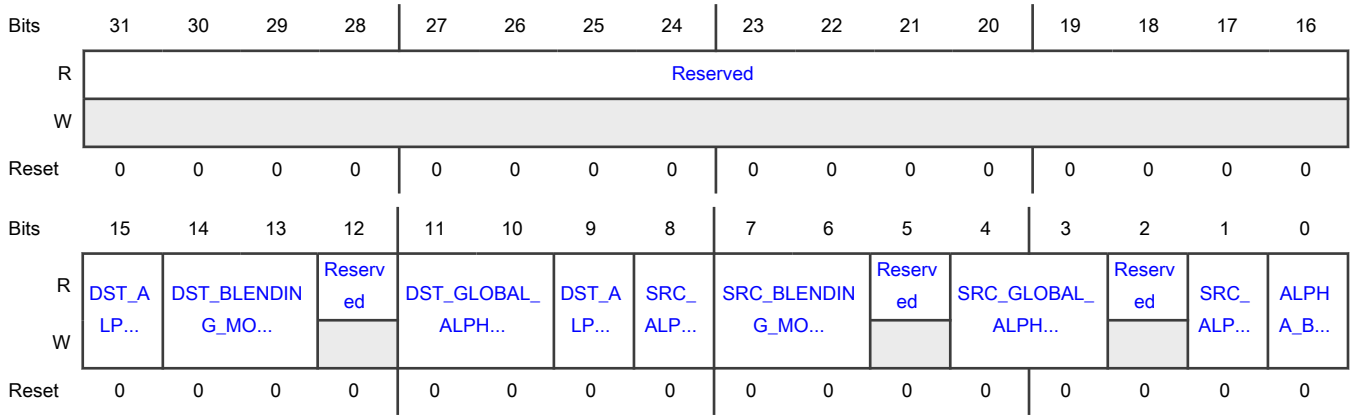
Offset

Register	Offset
OverlayAlphaBlendConfig	20A4h

Function

Sets alpha blending for overlay.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 DST_ALPHA_FACTOR	Destination Alpha Factor Generates the blending factor for destination alpha. **NOTE** This field is double buffered. 0b - Do not generate the blending factor for destination alpha. 1b - Generates the blending factor for destination alpha.
14-13 DST_BLENDING_MODE	Destination Blending Mode Sets the blending mode for destination alpha. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the destination color 11b - Not affected by the color of the underlying pixel
12 —	Reserved
11-10 DST_GLOBAL_ALPHA_MODE	Destination Global Alpha Mode Sets the global destination alpha mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field is double buffered. 00b - Normal destination alpha mode 01b - Global destination alpha mode 10b - Scaled (by the global alpha value) destination alpha mode
9 DST_ALPHA_MODE	Destination Alpha Mode Sets destination alpha mode. **NOTE** This field is double buffered. 0b - Normal destination alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
8 SRC_ALPHA_FACTOR	Source Alpha Factor Generates the blending factor for source alpha for the overlay layer. **NOTE** This field is double buffered. 0b - Do not generate the blending factor. 1b - Generates the blending factor.
7-6 SRC_BLENDING_MODE	Source Blending Mode Sets the blending mode for source alpha. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the source color 11b - Not affected by the color of the underlying pixel
5 —	Reserved
4-3 SRC_GLOBAL_ALPHA_MODE	Source Global Alpha Mode Sets the global alpha mode to substitute or scale the source alpha. 00b - Normal source alpha mode 01b - Global source alpha mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Scaled (by the global alpha value) source alpha mode
2 —	Reserved
1 SRC_ALPHA_MODE	Source Alpha Mode Sets the source alpha mode for blending. 0b - Normal alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
0 ALPHA_BLEND	Alpha Blend Enables alpha blending for overlay from the bottom layer to the top layer. **NOTE** This field is double buffered. 0b - Disables alpha blending for overlay. 1b - Enables alpha blending for overlay.

55.6.28 Overlay Global Alpha (OverlayGlobalAlpha)

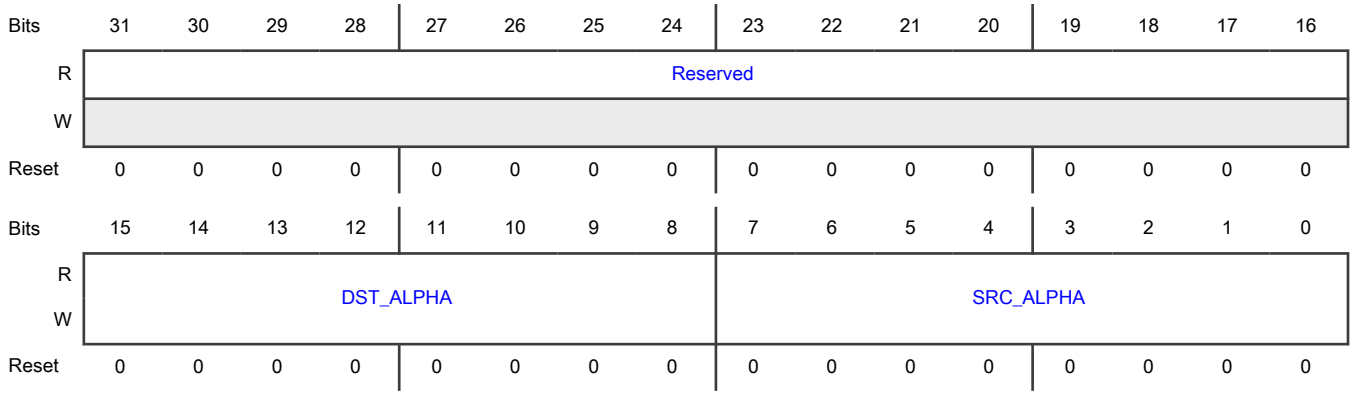
Offset

Register	Offset
OverlayGlobalAlpha	20A8h

Function

Sets the global alpha for overlay.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DST_ALPHA	Destination Alpha Sets the destination global alpha for overlay. **NOTE** This field is double buffered.
7-0 SRC_ALPHA	Source Alpha Sets the source global alpha for overlay. **NOTE** This field is double buffered.

55.6.29 Overlay Clear Value (OverlayClearValue)

Offset

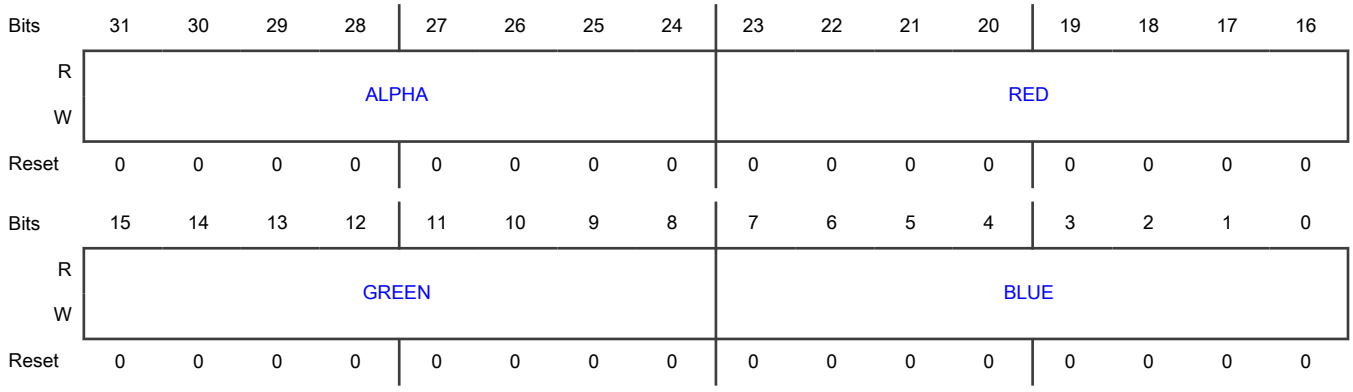
Register	Offset
OverlayClearValue	20ACh

Function

Sets the clear value for overlay, which is used when OverlayConfig[CLEAR_EN] is enabled in the format of A8R8G8B8.

This register is double buffered.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.30 Overlay Layer 1 Configuration (OverlayConfig1)

Offset

Register	Offset
OverlayConfig1	20B0h

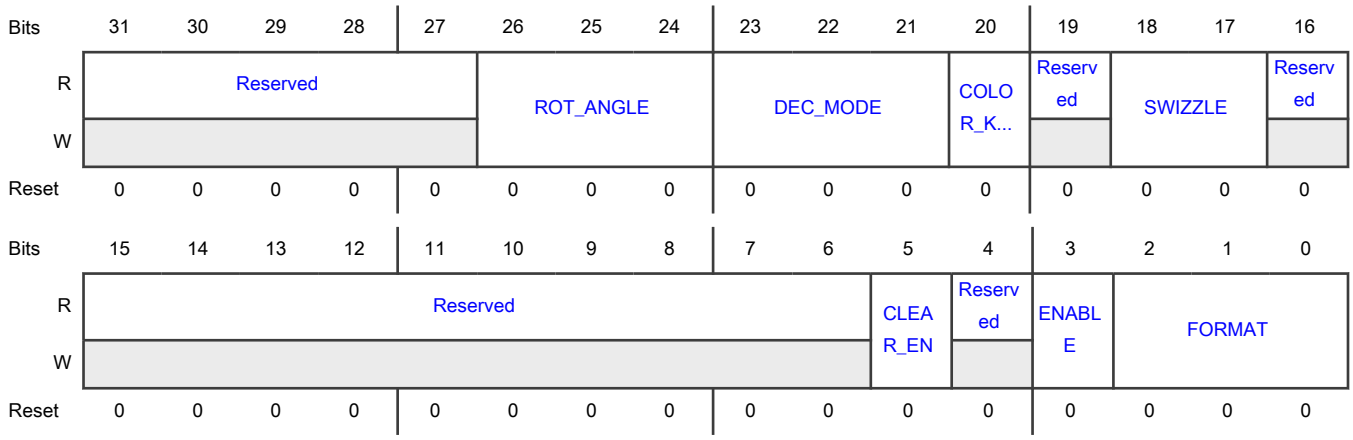
Function

Configures the overlay layer 1 for control.

NOTE

This register is double buffered.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-24 ROT_ANGLE	Rotation Angle Sets the rotation angle for source surface. Height field specifies the height of the surface in pixels. **NOTE** This field is double buffered. 000b - Sets the rotation angle to 0 degree. 010b - Sets the rotation angle to 180 degree. 100b - Flips horizontal 101b - Flips vertical
23-21 DEC_MODE	Decompression Mode Enables or disables decompression mode on the overlay layer 1. **NOTE** This field is double buffered. 000b - Disables decompression. 001b - Enables non_subsample decompression mode. 010b - Enables horizontal subsample decompression mode. 011b - Enables horizontal and vertical subsample decompression mode. 100b - Enables ETC2 decompression mode.
20 COLOR_KEY_EN	Enable Color Keying Enables or disables color keying on the overlay layer 1. **NOTE** This field is double buffered. 0b - Disables color keying. 1b - Enables color keying.
19 —	Reserved
18-17 SWIZZLE	RGB Swizzle Swizzles an input pixel for RGB formats at the overlay layer 1 to align the component order. **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Input pixel in ARGB format order 01b - Input pixel in RGBA format order 10b - Input pixel in ABGR format order 11b - Input pixel in BGRA format order
16-6 —	Reserved
5 CLEAR_EN	Enable Clear Enables or disables clearing the overlay layer 1. **NOTE** This field is double buffered. 0b - Disables clearing the overlay layer 1. 1b - Enables clearing the overlay layer 1.
4 —	Reserved
3 ENABLE	Enable Enables or disables the overlay layer 1. **NOTE** This field is double buffered. 0b - Disables the overlay layer 1. 1b - Enables the overlay layer 1.
2-0 FORMAT	Input Format Sets the input format for overlay. None means there is no overlay. **NOTE** This field is double buffered. 000b - No overlay 001b - Sets the input format to A4R4G4B4. 010b - Sets the input format to A1R5G5B5. 011b - Sets the input format to R5G6B5. 100b - Sets the input format to A8R8G8B8. 101b - Sets the input format to R8G8B8. 110b - Sets the input format to A8R5G6B5.

55.6.31 Overlay Layer 1 Address (OverlayAddress1)

Offset

Register	Offset
OverlayAddress1	20B4h

Function

Sets the starting address of the overlay layer 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ADDRESS	Start Address Sets a 32-bit address as the starting address of the overlay layer 1.

55.6.32 Overlay Layer 1 Stride (OverlayStride1)

Offset

Register	Offset
OverlayStride1	20B8h

Function

Sets the stride of the overlay layer 1 in bytes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															STRIDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STRIDE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-17 —	Reserved
16-0 STRIDE	Stride of Overlay Layer 1 Number of bytes from start of one line to next line.

55.6.33 Overlay Layer 1 Top Left (OverlayTL1)

Offset

Register	Offset
OverlayTL1	20BCh

Function

Sets the top left coordinate of panel pixel where the overlay layer 1 should start.

NOTE

There is no panning inside the overlay layer 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 Y	Top boundary Top boundary of the overlay layer 1 window.
15-12 —	Reserved
11-0 X	Left Boundary Left boundary of the overlay layer 1 window.

55.6.34 Overlay Layer 1 Size (OverlaySize1)

Offset

Register	Offset
OverlaySize1	20C0h

Function

Sets the size of the overlay layer 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				HEIGHT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				WIDTH											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 HEIGHT	Height Sets the height of the overlay layer 1.
15-12 —	Reserved
11-0 WIDTH	Width Sets the width of the overlay layer 1.

55.6.35 Overlay Layer 1 Color Keying Start Address (OverlayColorKey1)

Offset

Register	Offset
OverlayColorKey1	20C4h

Function

Sets the start address of color keying for the overlay layer 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ALPHA								RED							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GREEN								BLUE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.36 Overlay Layer 1 Color Keying End Address (OverlayColorKeyHigh1)

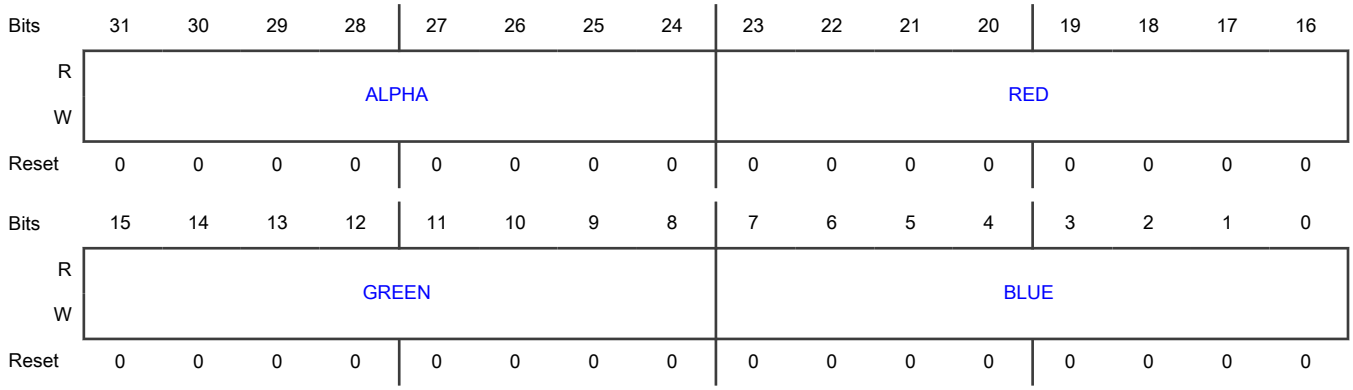
Offset

Register	Offset
OverlayColorKeyHigh1	20C8h

Function

Sets the end address of color keying for the overlay layer 1.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.37 Overlay Layer 1 Alpha Blending Configuration (OverlayAlphaBlendConfig1)

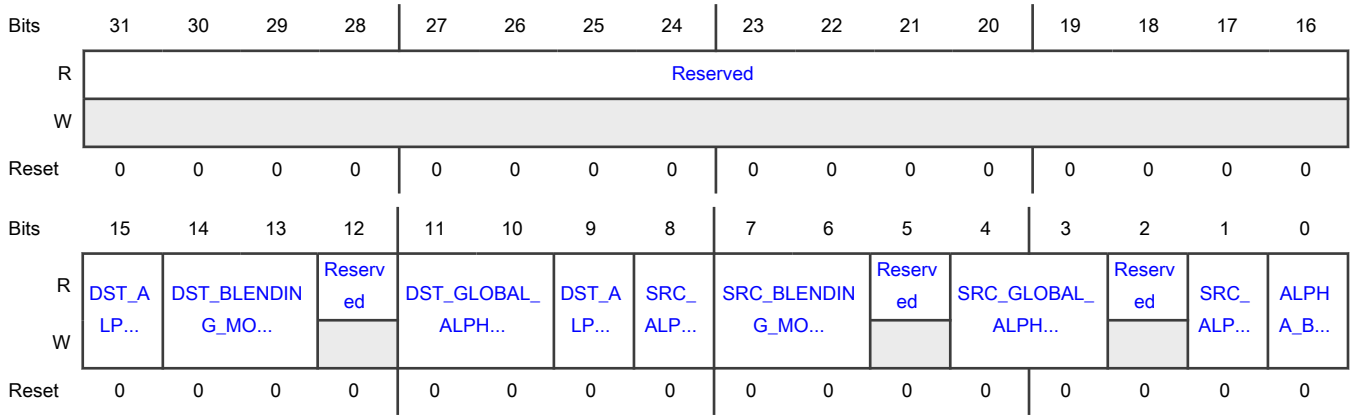
Offset

Register	Offset
OverlayAlphaBlendConfig1	20CCh

Function

Sets the alpha blending for the overlay layer 1.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 DST_ALPHA_FACTOR	DST Alpha Factor Generates the blending factor for destination alpha for the overlay layer 1. **NOTE** This field is double buffered. 0b - Do not generate the blending factor for destination alpha for the overlay layer 1. 1b - Generates the blending factor for destination alpha for the overlay layer 1.
14-13 DST_BLENDING_MODE	Destination Blending Mode Sets the destination alpha blending mode for the overlay layer 1. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the source color 11b - Not affected by the color of the underlying pixel
12 —	Reserved
11-10 DST_GLOBAL_ALPHA_MODE	Destination Global Alpha Mode Sets the destination global alpha mode for the overlay layer 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field is double buffered. 00b - Normal destination alpha mode 01b - Global destination alpha mode 10b - Scaled (by the global alpha value) destination alpha mode
9 DST_ALPHA_MODE	Destination Alpha Mode Sets the destination alpha mode for the overlay layer 1. **NOTE** This field is double buffered. 0b - Normal alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
8 SRC_ALPHA_FACTOR	Source Alpha Factor Generates the blending factor for source alpha for the overlay layer 1. **NOTE** This field is double buffered. 0b - Do not generate the blending factor for source alpha for the overlay layer 1. 1b - Generates the blending factor for source alpha for the overlay layer 1.
7-6 SRC_BLENDING_MODE	Source Blending Mode Sets the source alpha blending mode for the overlay layer 1. **NOTE** This field is double buffered. 00b - Each component of the color is multiplied by (0, 0, 0,0). 01b - Each component of the color is multiplied by (1, 1, 1,1). 10b - Normal as the result color is the source color 11b - Not affected by the color of the underlying pixel
5 —	Reserved
4-3 SRC_GLOBAL_ALPHA_MODE	Source Global Alpha Mode Sets the source global alpha mode for blending for the overlay layer 1. **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00b - Normal source alpha mode 01b - Global source alpha mode 10b - Scaled (by the global alpha value) source alpha mode
2 —	Reserved
1 SRC_ALPHA_MODE	Source Alpha Mode Sets the source alpha mode for blending for the overlay layer 1. **NOTE** This field is double buffered. 0b - Normal alpha mode 1b - Inversed alpha mode if the internal alpha rule needs to be followed
0 ALPHA_BLEND	Alpha Blend Enables alpha blending for the overlay layer 1 from the bottom layer to the top layer. **NOTE** This field is double buffered. 0b - Disables alpha blending for the overlay layer 1. 1b - Enables alpha blending for the overlay layer 1.

55.6.38 Overlay Layer 1 Global Alpha (OverlayGlobalAlpha1)

Offset

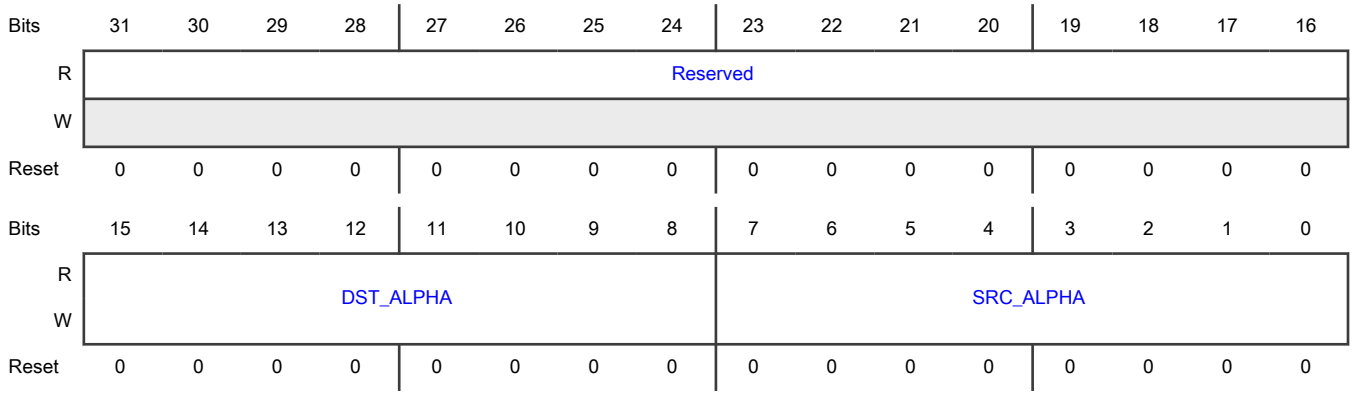
Register	Offset
OverlayGlobalAlpha1	20D0h

Function

Sets the global alpha value for the overlay layer 1.

NOTE
This field is double buffered.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DST_ALPHA	Destination Alpha Sets the destination global alpha for the overlay layer 1.
7-0 SRC_ALPHA	Source Alpha Sets the source global alpha for the overlay layer 1.

55.6.39 Overlay Layer 1 Clear Value (OverlayClearValue1)

Offset

Register	Offset
OverlayClearValue1	20D4h

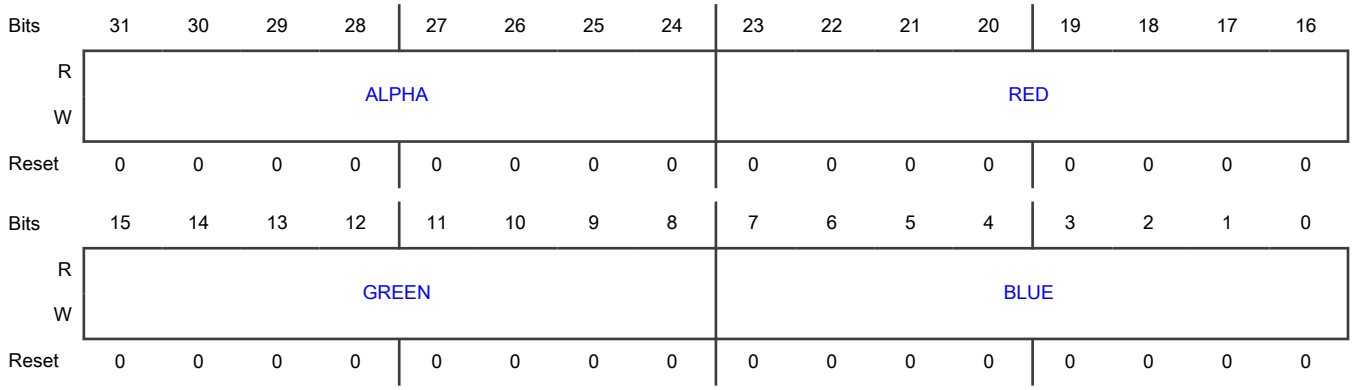
Function

Sets the clear value of the overlay layer 1 when OverlayConfig1[CLEAR_EN] is enabled in the format of A8R8G8B8.

NOTE

This field is double buffered.

Diagram



Fields

Field	Function
31-24 ALPHA	Alpha 8-bit alpha value
23-16 RED	Red 8-bit red color
15-8 GREEN	Green 8-bit green color
7-0 BLUE	Blue 8-bit blue color

55.6.40 Dither Lookup Table Low Threshold (DisplayDitherTableLow0)

Offset

Register	Offset
DisplayDitherTableLow0	20E0h

Function

Sets the lower threshold for creating a dither lookup table.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Y1_X3				Y1_X2				Y1_X1				Y1_X0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Y0_X3				Y0_X2				Y0_X1				Y0_X0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 Y1_X3	Y1_X3 Sets the low threshold value to x,y=3,1.
27-24 Y1_X2	Y1_X2 Sets the low threshold value to x,y=2,1.
23-20 Y1_X1	Y1_X1 Sets the low threshold value to x,y=1,1.
19-16 Y1_X0	Y1_X0 Sets the low threshold value to x,y=0,1.
15-12 Y0_X3	Y0_X3 Sets the low threshold value to x,y=3,0.
11-8 Y0_X2	Y0_X2 Sets the low threshold value to x,y=2,0.
7-4 Y0_X1	Y0_X1 Sets the low threshold value to x,y=1,0.
3-0 Y0_X0	Y0_X0 Sets the low threshold value to x,y=0,0.

55.6.41 Dither Lookup Table High Threshold (DisplayDitherTableHigh0)

Offset

Register	Offset
DisplayDitherTableHigh0	20E4h

Function

Sets the high threshold for creating a dither lookup table.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Y3_X3				Y3_X2				Y3_X1				Y3_X0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Y2_X3				Y2_X2				Y2_X1				Y2_X0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 Y3_X3	Y3_X3 Sets the high threshold value to x,y=3,3.
27-24 Y3_X2	Y3_X2 Sets the high threshold value to x,y=2,3.
23-20 Y3_X1	Y3_X1 Sets the high threshold value to x,y=1,3.
19-16 Y3_X0	Y3_X0 Sets the high threshold value to x,y=0,3.
15-12 Y2_X3	Y2_X3 Sets the high threshold value to x,y=3,2.
11-8 Y2_X2	Y2_X2 Sets the high threshold value to x,y=2,2.
7-4 Y2_X1	Y2_X1 Sets the high threshold value to x,y=1,2.
3-0 Y2_X0	Y2_X0 Sets the high threshold value to x,y=0,2.

55.6.42 Panel Configuration (PanelConfig0)

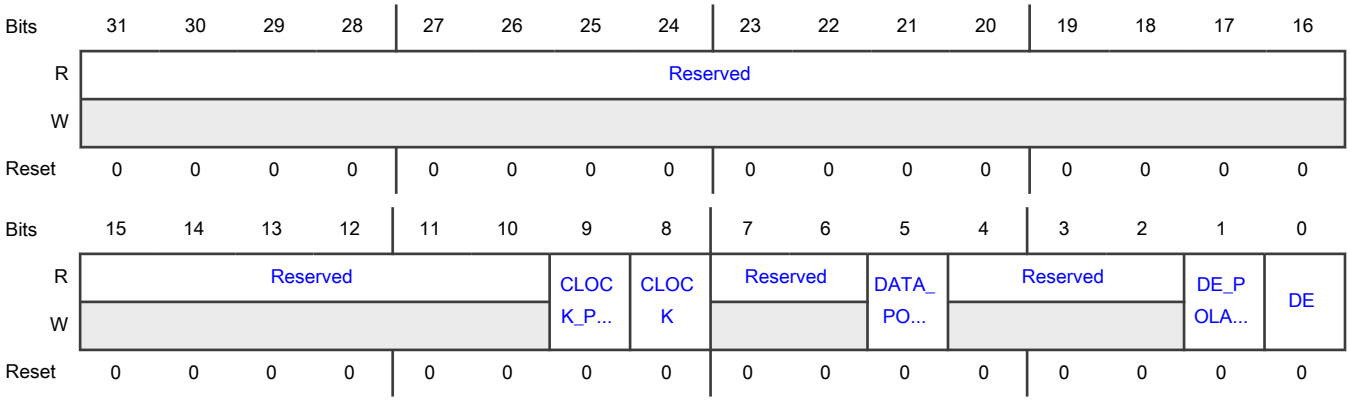
Offset

Register	Offset
PanelConfig0	20E8h

Function

Sets output panel and panel timing.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 CLOCK_POLARITY	Clock Polarity Sets the clock polarity. 0b - Sets the clock polarity to positive. 1b - Sets the clock polarity to negative.
8 CLOCK	Clock Enables or disables clock. 0b - Disables clock. 1b - Enables clock.
7-6 —	Reserved
5	Data Polarity

Table continues on the next page...

Table continued from the previous page...

Field	Function
DATA_POLARITY	Sets the data polarity. 0b - Sets the data polarity to positive. 1b - Sets the data polarity to negative.
4-2 —	Reserved
1 DE_POLARITY	Data Enable Polarity Sets the polarity of data enabled. 0b - Sets the polarity of data enabled to positive. 1b - Sets the polarity of data enabled to negative.
0 DE	Data Enable Enables or disables data. 0b - Disables data. 1b - Enables data.

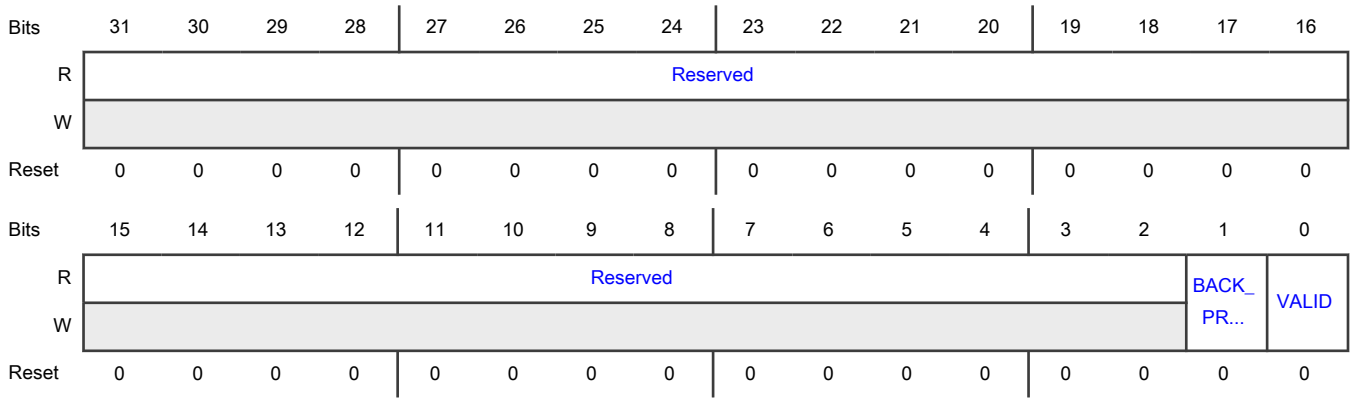
55.6.43 Panel Control (PanelControl)

Offset

Register	Offset
PanelControl	20ECh

Function
Updates parameters for the next frame.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 BACK_PRESSURE_DISABLE	Back Pressure Disable Enables or disables back pressure. 0b - Enables back pressure. 1b - Disables back pressure.
0 VALID	Valid Defines whether you can copy a new set of registers at the next VBLANK or not. This ensures a frame always starts with a valid working set if this register is programmed last, which reduces the need for software to wait for the start of a VBLANK signal in order to ensure all states are loaded before the next VBLANK. 0b - Pending 1b - Copies a set of registers at the next VBLANK.

55.6.44 Panel Function (PanelFunction)

Offset

Register	Offset
PanelFunction	20F0h

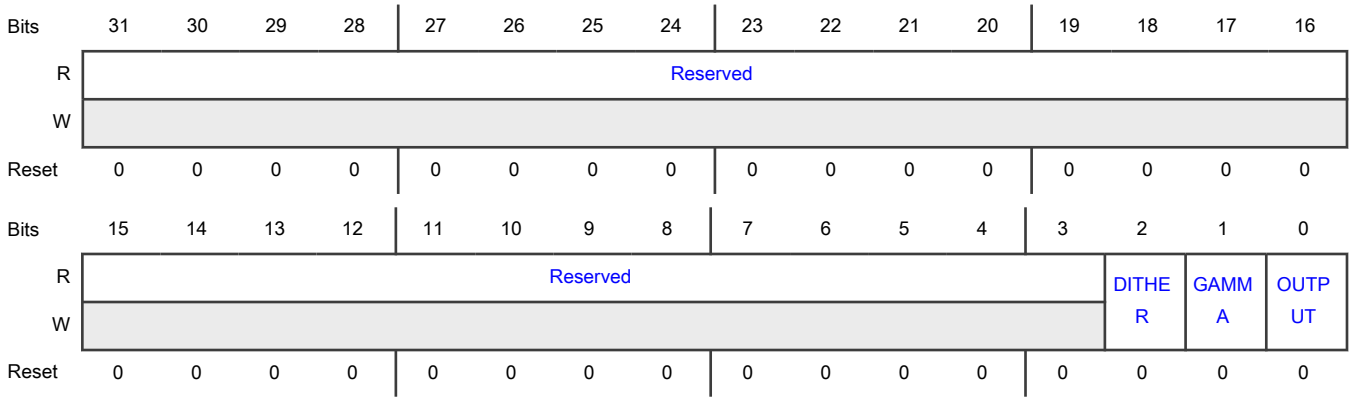
Function

Sets output panel.

NOTE

This register is double buffered.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 DITHER	Dither Enabling dithering allows R8G8B8 modes to show better on panels with less bits-per-pixel. **NOTE** This field is double buffered. 0b - Disables dither. 1b - Enables dither.
1 GAMMA	Gamma When Gamma is enabled, the R, G, and B channels are routed through the Gamma LUT to perform gamma correction. **NOTE** This field is double buffered. 0b - Disables Gamma. 1b - Enables Gamma.
0 OUTPUT	Output When Output is enabled, pixels will be displayed. When Output is disabled, all pixels will be black. This allows a panel to have correct timing but without any pixels. **NOTE** This field is double buffered. 0b - Disables output panel. 1b - Enables output panel.

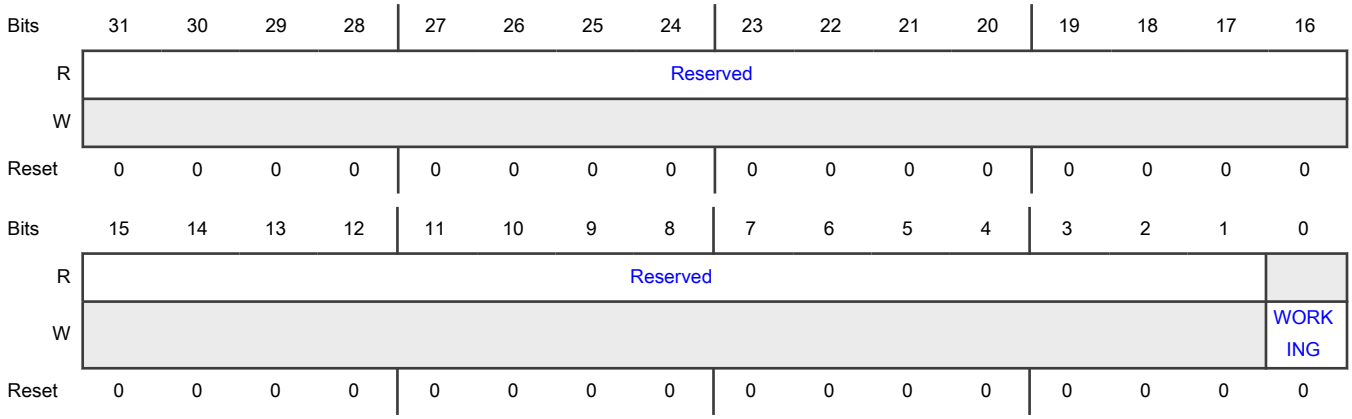
55.6.45 Panel Working (PanelWorking)

Offset

Register	Offset
PanelWorking	20F4h

Function
Triggers display controller to start the interface mode transfer.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 WORKING	Working Writing one in this register will force a reset of the display controller. All registers will be copied to the working set and counters will be reset to the end of HSYNC and VSYNC. 1b - Starts reset of the display controller.

55.6.46 Panel State (PanelState)

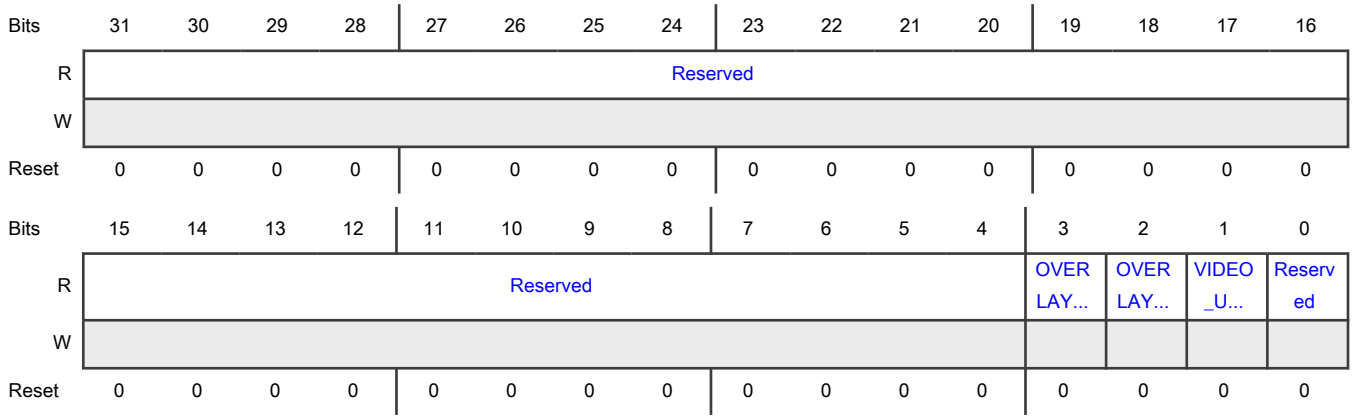
Offset

Register	Offset
PanelState	20F8h

Function

Indicates the display controller status.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 OVERLAY_UNDER_FLOW1	Overlay Layer 1 Under Flow When the overlay layer 1 FIFO underflows, this bit gets set to one. Reading this register will reset it back to zero. 0b - Do not set the overlay layer 1 FIFO to underflow. 1b - Sets the overlay layer 1 FIFO to underflow.
2 OVERLAY_UNDER_FLOW	Overlay Layer 0 Under Flow When the overlay layer 0 FIFO underflows, this bit gets set to one. Reading this register will reset it back to zero. 0b - Do not set the overlay layer 0 FIFO to underflow. 1b - Sets the overlay layer 0 FIFO to underflow.
1 VIDEO_UNDER_FLOW	Video Under Flow When the display FIFO underflows, this bit sets to one. Reading this register will reset it back to zero. 0b - Do not set the display FIFO video to underflow. 1b - Sets the display FIFO video to underflow.
0 —	Reserved

55.6.47 Horizontal Display Total and Visible Pixel Count (HDisplay0)

Offset

Register	Offset
HDisplay0	2100h

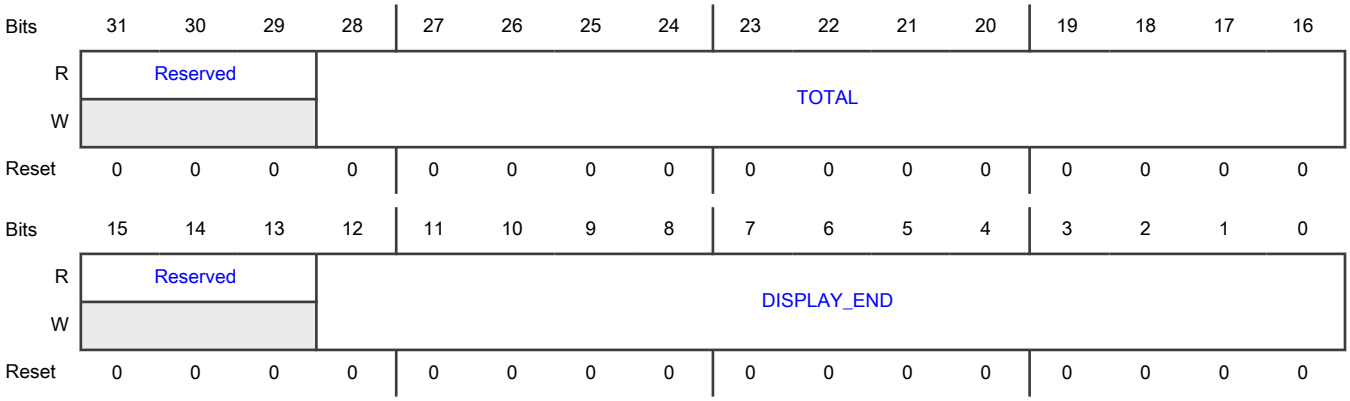
Function

Sets the horizontal total and display end counters.

NOTE

This register is double buffered.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 TOTAL	Total Total number of horizontal pixels.
15-13 —	Reserved
12-0 DISPLAY_END	Display End Number of visible horizontal pixels.

55.6.48 Horizontal Sync Counter (HSync0)

Offset

Register	Offset
HSync0	2104h

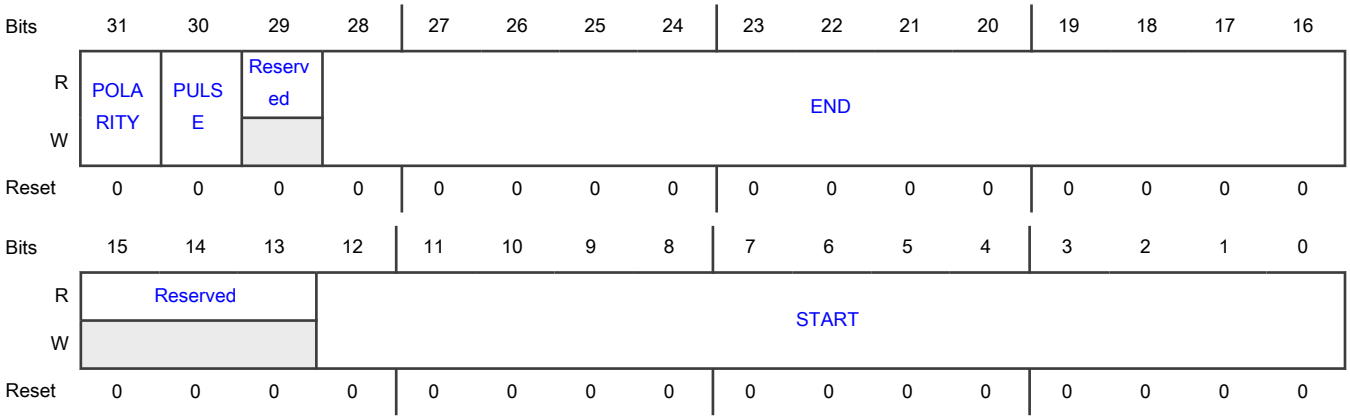
Function

Sets the horizontal sync pulse parameters for display.

NOTE

This register is double buffered.

Diagram



Fields

Field	Function
31 POLARITY	Sync Pulse Polarity Sets the polarity of the horizontal sync pulse. 0b - Sets the polarity of the horizontal sync pulse to positive. 1b - Sets the polarity of the horizontal sync pulse to negative.
30 PULSE	Sync Pulse Enables or disables horizontal sync pulse control. 0b - Disables horizontal sync pulse control. 1b - Enables horizontal sync pulse control.
29 —	Reserved
28-16	End

Table continues on the next page...

Table continued from the previous page...

Field	Function
END	Sets the end of horizontal sync pulse.
15-13 —	Reserved
12-0 START	Start Sets the start of horizontal sync pulse.

55.6.49 Vertical Total and Visible Pixel Count (VDisplay0)

Offset

Register	Offset
VDisplay0	2110h

Function

Sets the total and visible of vertical lines for display.

NOTE

This field is double buffered.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				TOTAL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				DISPLAY_END											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-16 TOTAL	Total Total number of visible vertical lines.
15-12 —	Reserved
11-0 DISPLAY_END	Display End Number of visible vertical lines.

55.6.50 Vertical Sync Counter (VSync0)

Offset

Register	Offset
VSync0	2114h

Function

Sets the vertical sync pulse parameters for display.

NOTE

This field is double buffered.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	POLARITY	PULSE	Reserved		END											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				START											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 POLARITY	Polarity Sets the polarity of the vertical sync pulse.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Sets the polarity of the vertical sync pulse to positive. 1b - Sets the polarity of the vertical sync pulse to negative.
30 PULSE	Vertical Sync Pulse Sets the vertical sync pulse control. 0b - Disables vertical sync pulse. 1b - Enables vertical sync pulse.
29-28 —	Reserved
27-16 END	End Sets the end of the vertical sync pulse.
15-12 —	Reserved
11-0 START	Start Sets the start of the vertical sync pulse.

55.6.51 Display Current Location (DisplayCurrentLocation0)

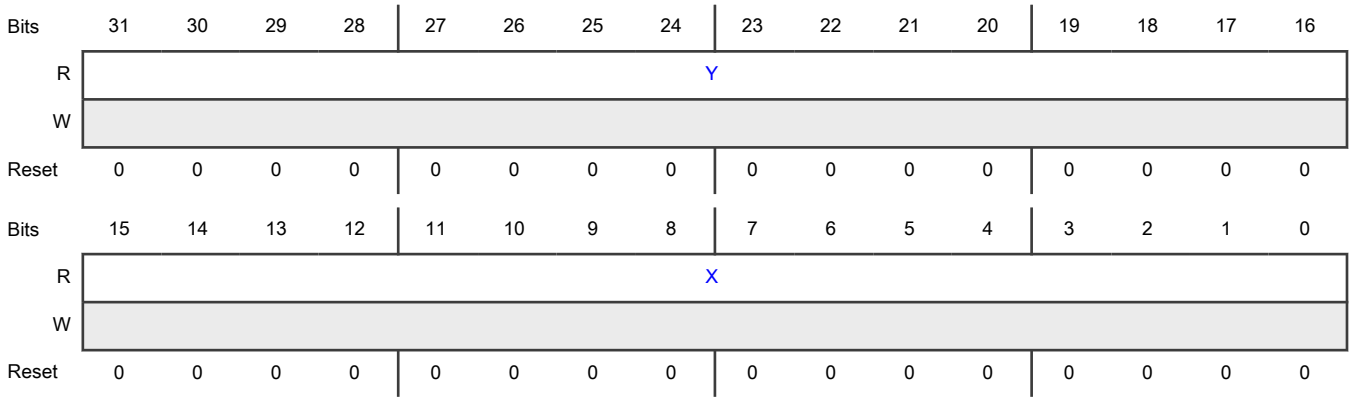
Offset

Register	Offset
DisplayCurrentLocation0	2118h

Function

Indicates the current x,y location of display controller.

Diagram



Fields

Field	Function
31-16 Y	Y Current Y location.
15-0 X	X Current X location.

55.6.52 Gamma Index (GammaIndex0)

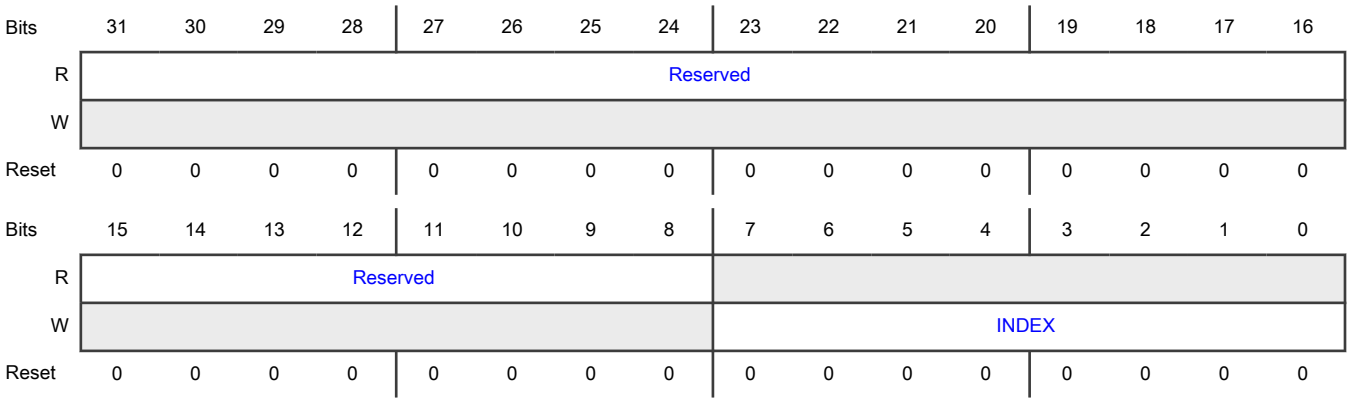
Offset

Register	Offset
GammaIndex0	211Ch

Function

Sets LUT index into the gamma lookup table.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 INDEX	Index Sets the LUT index into the gamma lookup table.

55.6.53 Gamma Data (GammaData0)

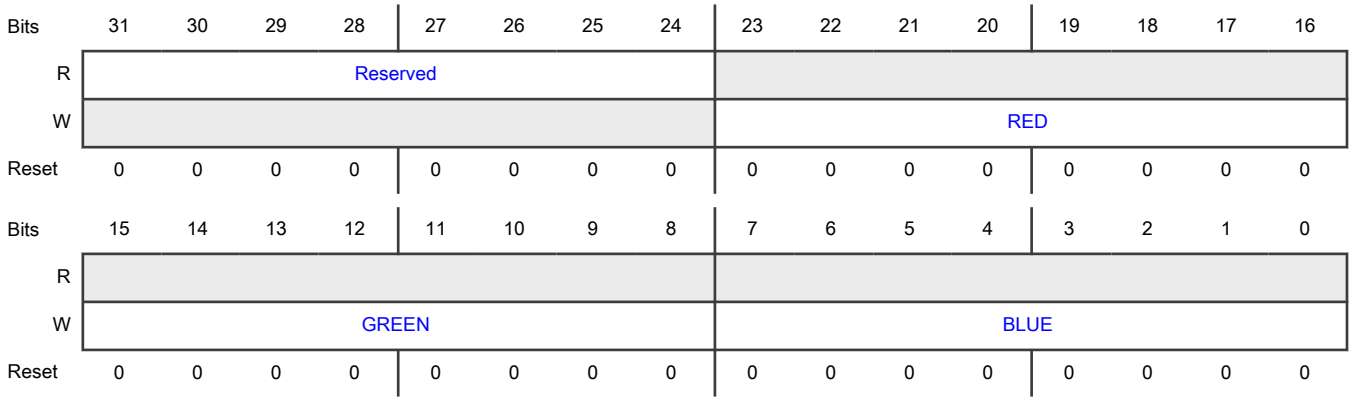
Offset

Register	Offset
GammaData0	2120h

Function

Translation values for the gamma table. When this register gets written, the data gets stored in the gamma table at the LUT index specified by the GammaIndex0 register. After the register is written, the index gets incremented.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 RED	Red Sets red translation value.
15-8 GREEN	Green Sets green translation value.
7-0 BLUE	Blue Sets blue translation value.

55.6.54 Cursor Configuration (CursorConfig)

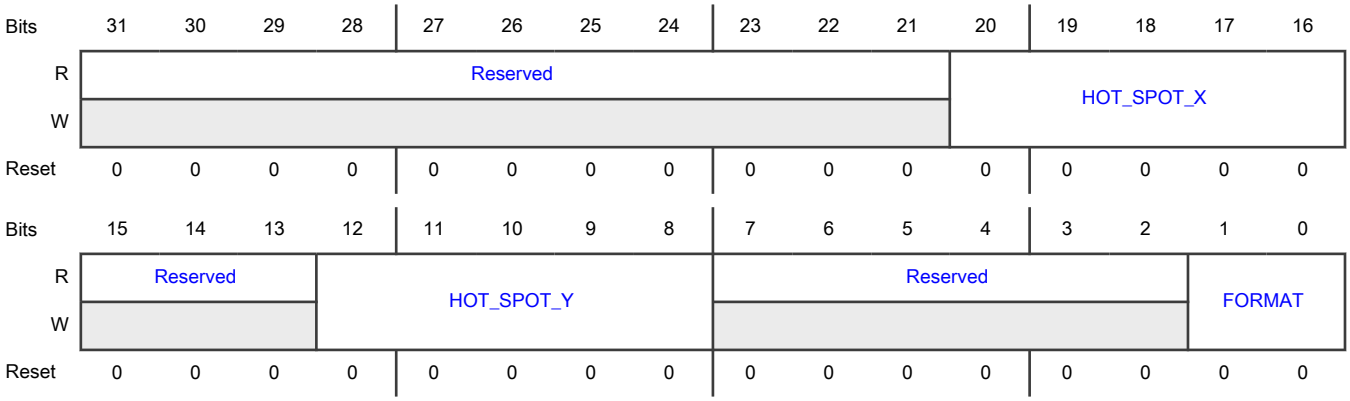
Offset

Register	Offset
CursorConfig	2124h

Function

Sets cursor parameters. A cursor is tracked through the top-left point of the cursor layer and the hotspot of the cursor. Double-buffered values in this register cannot be read while a flip is in progress.

Diagram



Fields

Field	Function
31-21 —	Reserved
20-16 HOT_SPOT_X	Horizontal Hot Spot Sets the horizontal offset to cursor hotspot. **NOTE** This field is double buffered.
15-13 —	Reserved
12-8 HOT_SPOT_Y	Vertical Hot Spot Sets the vertical offset to cursor hotspot. **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-2 —	Reserved
1-0 FORMAT	Format Sets the format of a cursor. **NOTE** This field is double buffered. 00b - Disables the format of a cursor. 01b - Sets the cursor in masked format, where the mask values of the AND and XOR bits decide the color of one cursor image pixel.. 10b - Sets the cursor in A8R8G8B8 format, where the color of the cursor image is decided by the ARGB values fetched from the memory.

55.6.55 Cursor Base Address (CursorAddress)

Offset

Register	Offset
CursorAddress	2128h

Function

Sets the base address of the cursor shape.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ADDRESS	Base address A 32-bit base address.

55.6.56 Cursor Location (CursorLocation)

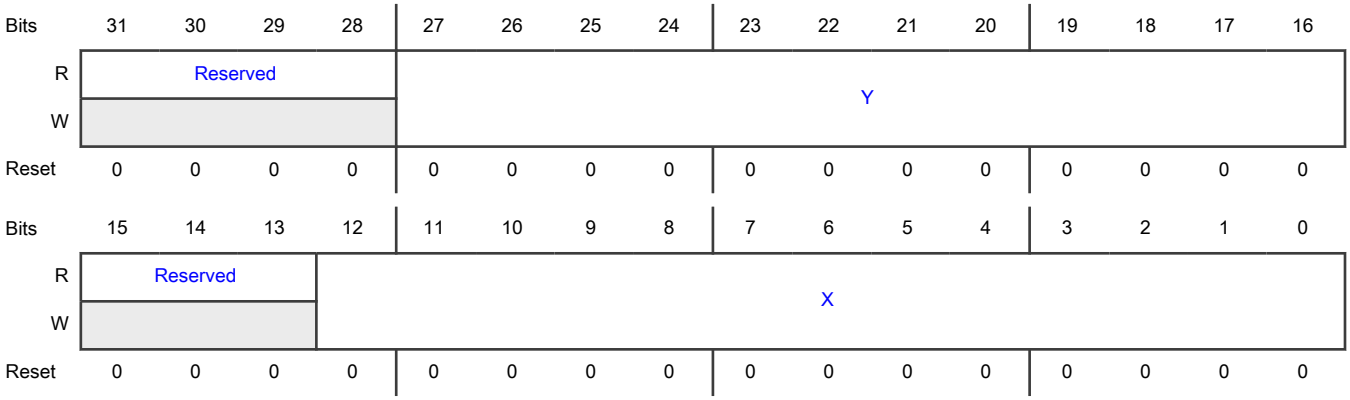
Offset

Register	Offset
CursorLocation	212Ch

Function

Sets the location of the cursor on the owning display.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-16 Y	Y Sets the Y location of cursor's hotspot. **NOTE** This field is double buffered.
15-13 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
12-0 X	X Sets X horizontal location of cursor's hotspot. **NOTE** This field is double buffered.

55.6.57 Cursor Background Color (CursorBackground)

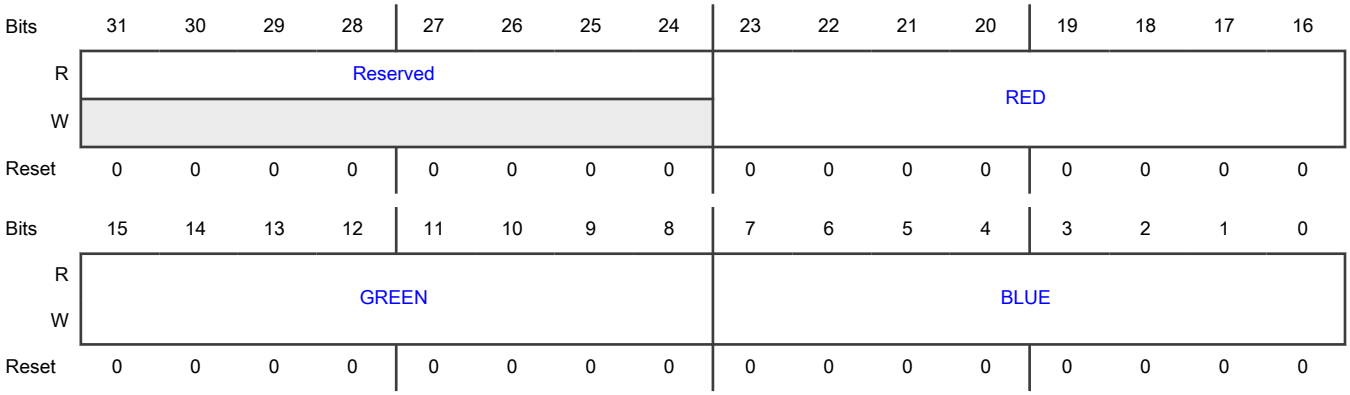
Offset

Register	Offset
CursorBackground	2130h

Function

Sets the background color of cursor in the masked format.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 RED	Red Red value **NOTE** This field is double buffered.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-8 GREEN	Green Green value **NOTE** This field is double buffered.
7-0 BLUE	Blue Blue value **NOTE** This field is double buffered.

55.6.58 Cursor Foreground Color (CursorForeground)

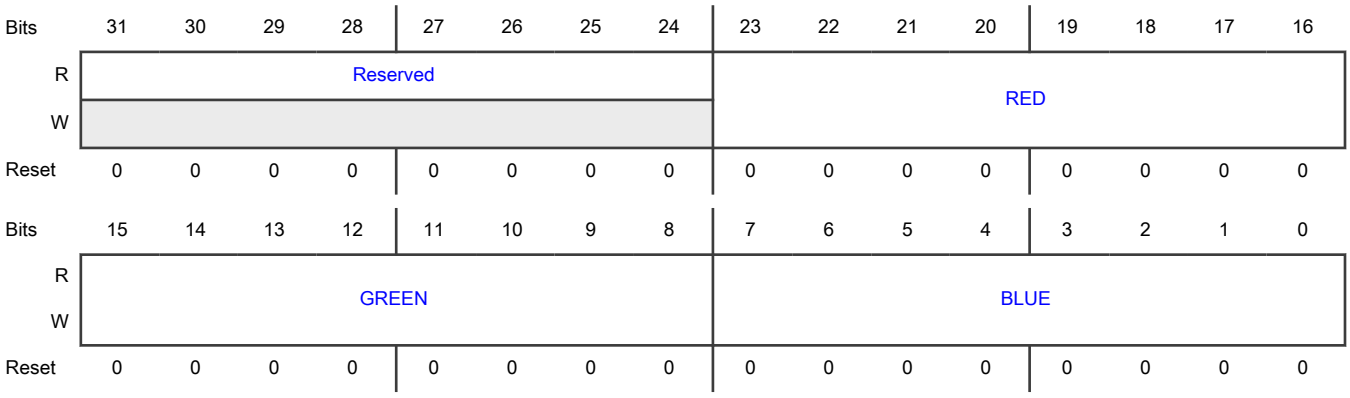
Offset

Register	Offset
CursorForeground	2134h

Function

Sets the foreground color of cursor in masked format.

Diagram



Fields

Field	Function
31-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
23-16 RED	Red Red value **NOTE** This field is double buffered.
15-8 GREEN	Green Green value **NOTE** This field is double buffered.
7-0 BLUE	Blue Blue value **NOTE** This field is double buffered.

55.6.59 Display Interrupt (DisplayIntr)

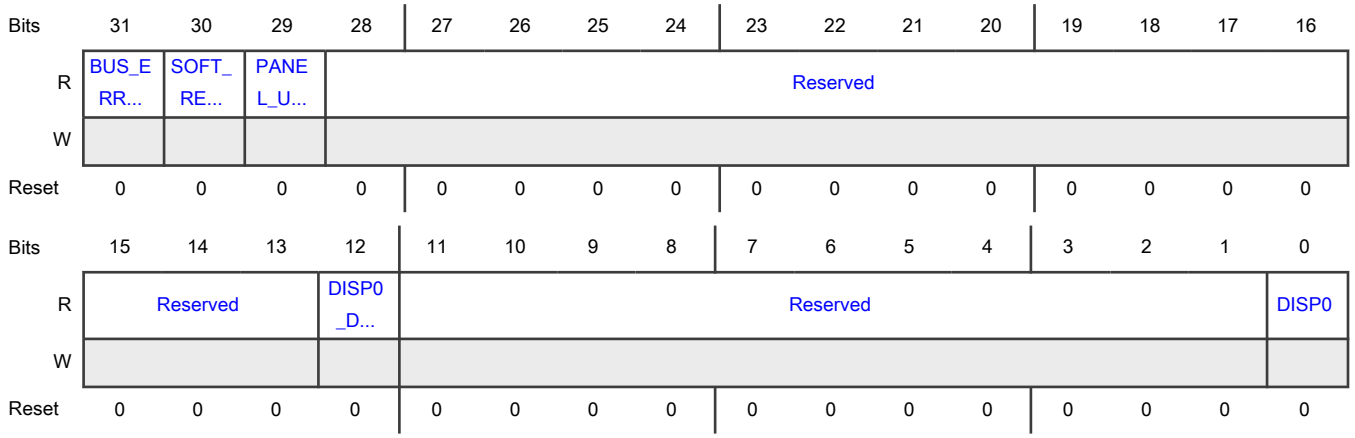
Offset

Register	Offset
DisplayIntr	2138h

Function

This is the interrupt register for the display. This register will automatically clear after a read. The interrupt bit is set when the current frame buffer is used up. The interrupt signal will be pulled up only when the interrupt enable bit in the DisplayIntrEnable register is enabled.

Diagram



Fields

Field	Function
31 BUS_ERROR	Bus Error Bus error interrupt
30 SOFT_RESET_DONE	Soft Reset Done Soft reset done interrupt
29 PANEL_UNDERFLOW	Panel Underflow Panel underflow interrupt
28-13 —	Reserved
12 DISP0_DBI_CFG_ERROR	Display_0 DBI Configure Error DBI configuration error for Display_0
11-1 —	Reserved
0 DISP0	Display_0 Frame done interrupt

55.6.60 Display Interrupt Enable (DisplayIntrEnable)

Offset

Register	Offset
DisplayIntrEnable	213Ch

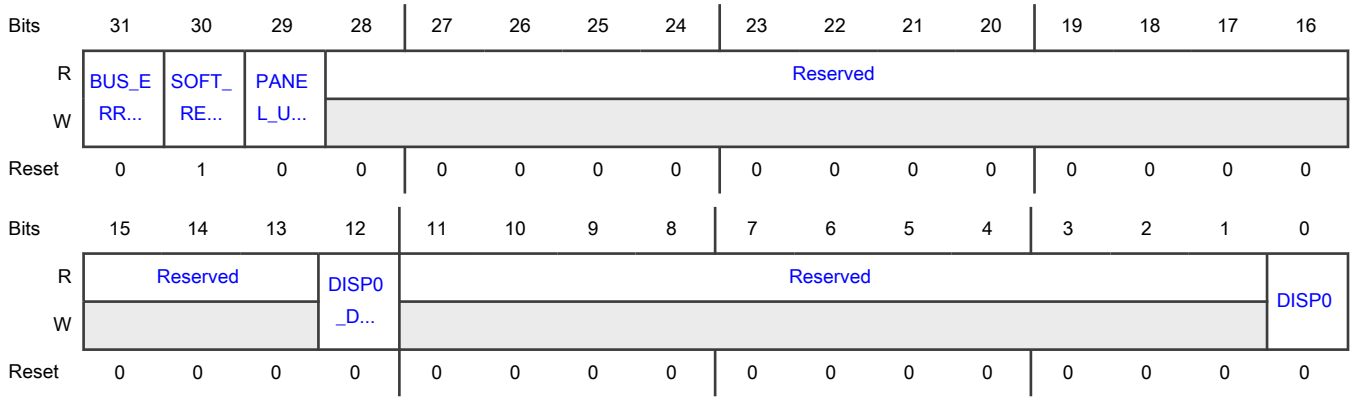
Function

Enables interrupt for display_0 (and display_1 if present).

NOTE

This field is double buffered.

Diagram



Fields

Field	Function
31 BUS_ERROR	Bus Error Enables bus error interrupt. 0b - Enables bus error interrupt. 1b - Disables bus error interrupt.
30 SOFT_RESET_DONE	Soft Reset Done Enables soft reset done interrupt. 0b - Disables soft reset done interrupt. 1b - Enables soft reset done interrupt.
29 PANEL_UNDERFLOW	Panel Underflow Enables panel underflow interrupt. 0b - Enables panel underflow interrupt. 1b - Disables panel underflow interrupt.
28-13 —	Reserved
12 DISP0_DBI_CFG_ERROR	Display_0 DBI Configuration Error Enables Display_0 DBI configuration error interrupt. **NOTE** This field is double buffered. 0b - Enables Display_0 DBI configuration error. 1b - Disables Display_0 DBI configuration error.
11-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
0 DISP0	Display_0 Enables Display_0 interrupt. 0b - Enables Display_0 interrupt. 1b - Disables Display_0 interrupt.

55.6.61 DBI Configuration (DbiConfig0)

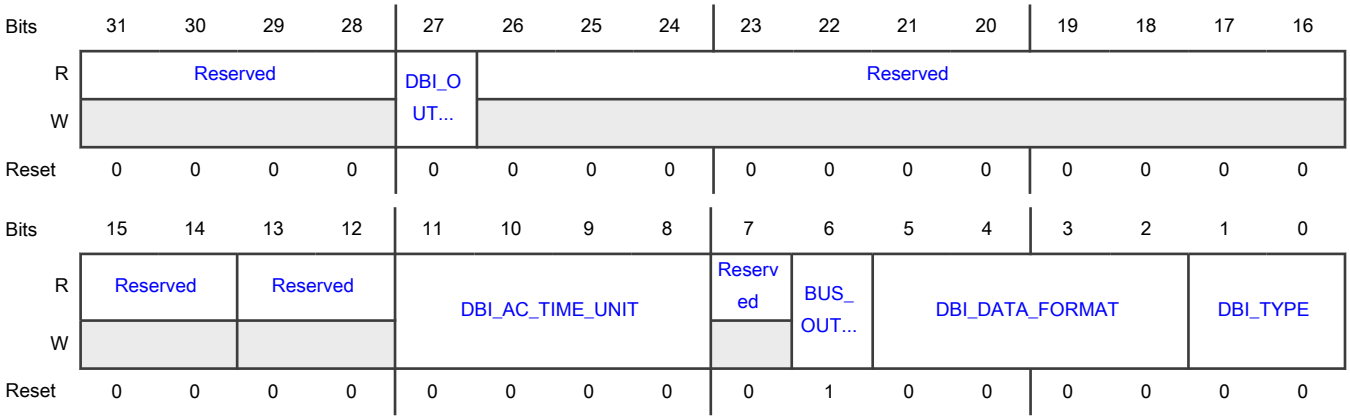
Offset

Register	Offset
DbiConfig0	2140h

Function

Sets output format for DBI types.

Diagram



Fields

Field	Function
31-28 —	Reserved
27 DBI_OUTPUT_SWIZZLE	DBIT Output Swizzle Set Sets DBI output data swizzle.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - RGB 1b - BGR
26-14 —	Reserved
13-12 —	Reserved
11-8 DBI_AC_TIME_UNIT	DBI AC Time Unit Sets the time units for AC characteristics (such as when to assert/de-assert a signal) for each DBI type.
7 —	Reserved
6 BUS_OUTPUT_SEL	Output Bus Select Selects the display output bus. 0b - Uses DPI bus. 1b - Uses DBI bus.
5-2 DBI_DATA_FORMAT	DBI Data Format Specifies the DBI output data format. 0000b - D8R3G3B2 0001b - D8R4G4B4 0010b - D8R5G6B5 0011b - D8R6G6B6 0100b - D8R8G8B8 0101b - D9R6G6B6 0110b - D16R3G3B2 0111b - D16R4G4B4 1000b - D16R5G6B5 1001b - D16_R6_G6_B6_OP1 1010b - D16_R6_G6_B6_OP2 1011b - D16_R8_G8_B8_OP1 1100b - D16_R8_G8_B8_OP2
1-0	DBI Type Sets the DBI type for output.

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBI_TYPE	00b - Selects DBI Type A Fixed E mode. 01b - Selects DBI Type A Clocked E mode. 10b - Selects DBI type B.

55.6.62 DBI Interface Reset (DbiIfReset0)

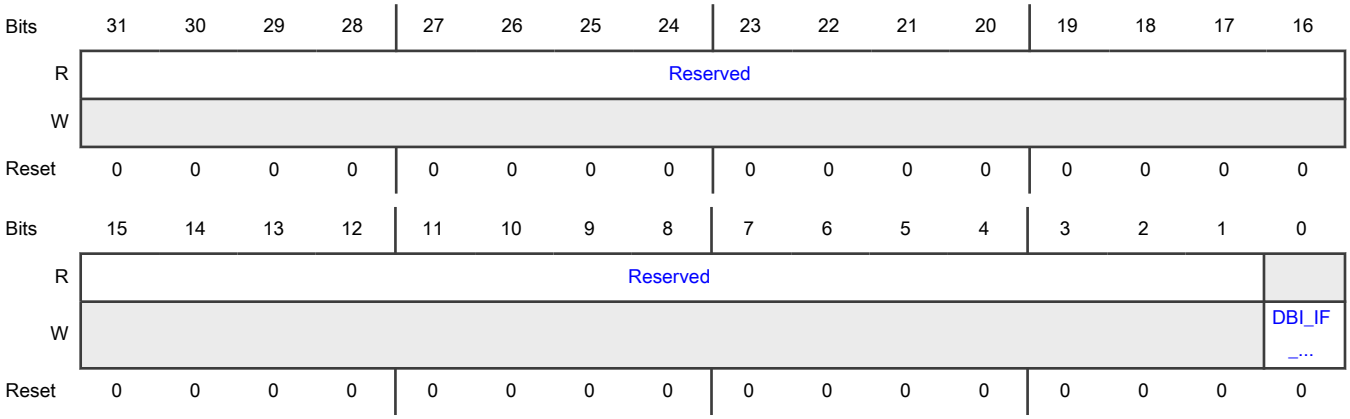
Offset

Register	Offset
DbiIfReset0	2144h

Function

Resets DBI bus or not.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 DBI_IF_LEVEL_RESET	DBI Interface Level Reset Resets DBI interface to idle state. 1b - Resets the DBI bus in idle state.

55.6.63 DBI Write AC Characteristics 1 (DbiWrChar10)

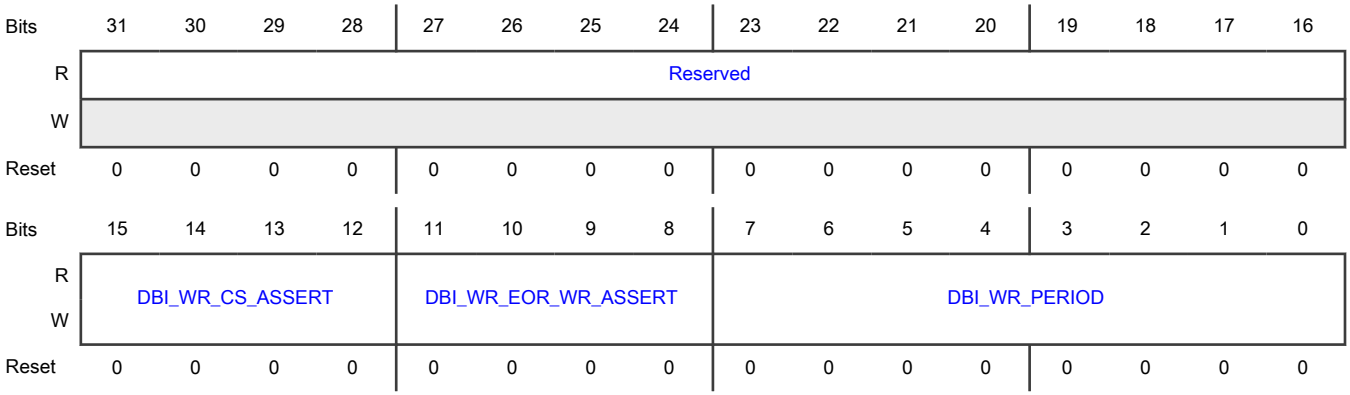
Offset

Register	Offset
DbiWrChar10	2148h

Function

Sets the DBI output timing parameters and the AC timing specifications of the panel for DBI Type A.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-12 DBI_WR_CS_ASSERT	DBI Write CS Assert Cycle number = Setting * (DbiAcTimeUnit + 1). • 0: When Type A Fixed E mode: Time to assert CSX. • 1: When Type A Clock E mode: Not used. • 2: When Type B mode: Not used.
11-8 DBI_WR_EOR_WR_ASSERT	DBI Clock E Signal or WRX Signal Assert Cycle number = Setting * (DbiAcTimeUnit + 1). • 0: When Type A Fixed E mode: Not used. • 1: When Type A Clock E mode: Time to assert E. • 2: When Type B mode: Time to assert WRX.
7-0	DBI Write Period Sets the single write period duration.

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBI_WR_PERIOD	Cycle number = Setting * (DbiAcTimeUnit + 1). This field must be no less than 3.

55.6.64 DBI Write AC Characteristics 2 (DbiWrChar20)

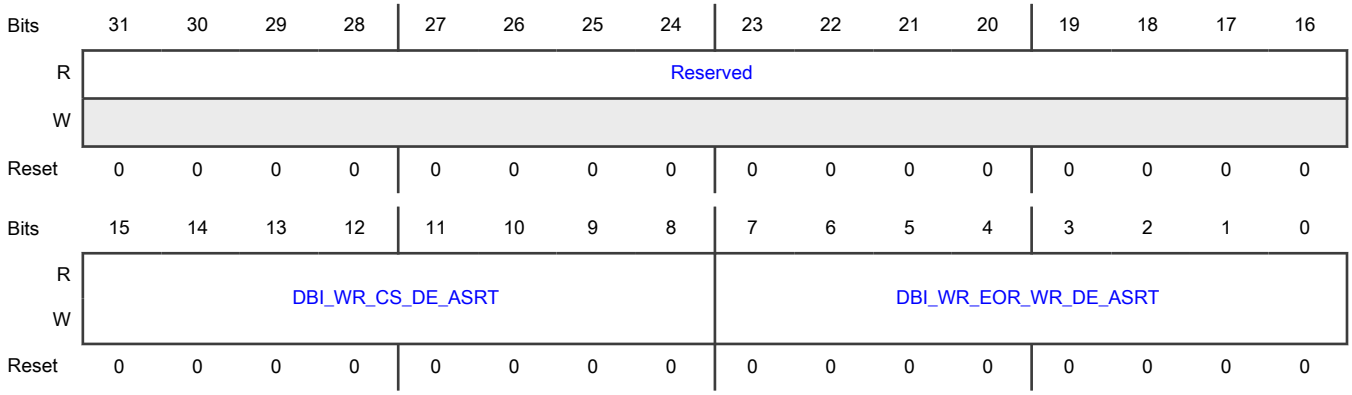
Offset

Register	Offset
DbiWrChar20	214Ch

Function

Sets the DBI output timing parameters and the AC timing specifications of the panel for DBI Type A and B.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DBI_WR_CS_DE_ASRT	DBI Write CS De-assert Cycle number = Setting * (DbiAcTimeUnit + 1). • 0: When Type A Fixed E mode: Time to de-assert CSX. • 1: When Type A Clock E mode: Not used. • 2: When Type B mode: Not used.
7-0	DBI Clock E Signal or WRX Signal Desert Assert

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBI_WR_EOR_WR_DE_ASRT	Cycle number = Setting * (DbiAcTimeUnit + 1). • 0: When Type A Fixed E mode: Not used. • 1: When Type A Clock E mode: Time to de-assert E. • 2: When Type B mode: Time to de-assert WRX.

55.6.65 DBI Command Control (DbiCmd0)

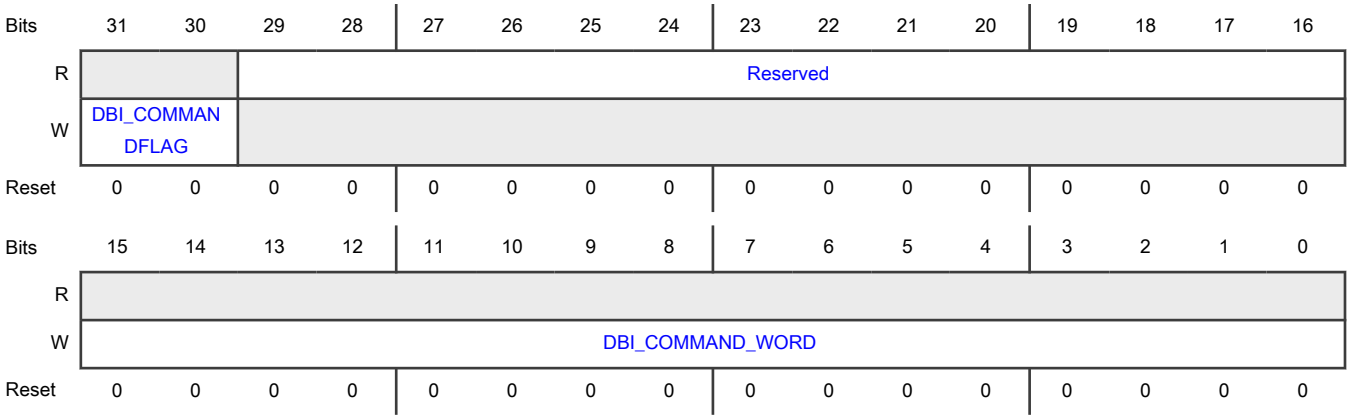
Offset

Register	Offset
DbiCmd0	2150h

Function

DBI command in/out port. Writes to this register will send command/data to the DBI port.

Diagram



Fields

Field	Function
31-30 DBI_COMMAN DFLAG	DBI Command Flag Sets the type of data to be sent to the DBI port and stored in [DBI_COMMAND_WORD]. 00b - Address. [DBI_COMMAND_WORD] will contain an address. 01b - Write Memory Start. Starts to write memory. The contents contained in [DBI_COMMAND_WORD] are ignored. 10b - Parameter or Data. [DBI_COMMAND_WORD] will contain a parameter or data.

Table continues on the next page...

Table continued from the previous page...

Field	Function
29-16 —	Reserved
15-0 DBI_COMMAND_WORD	DBI Command Word The type of data contained in this word is specified using [DBI_COMMANDFLAG].

55.6.66 DPI Configuration (DpiConfig0)

Offset

Register	Offset
DpiConfig0	2154h

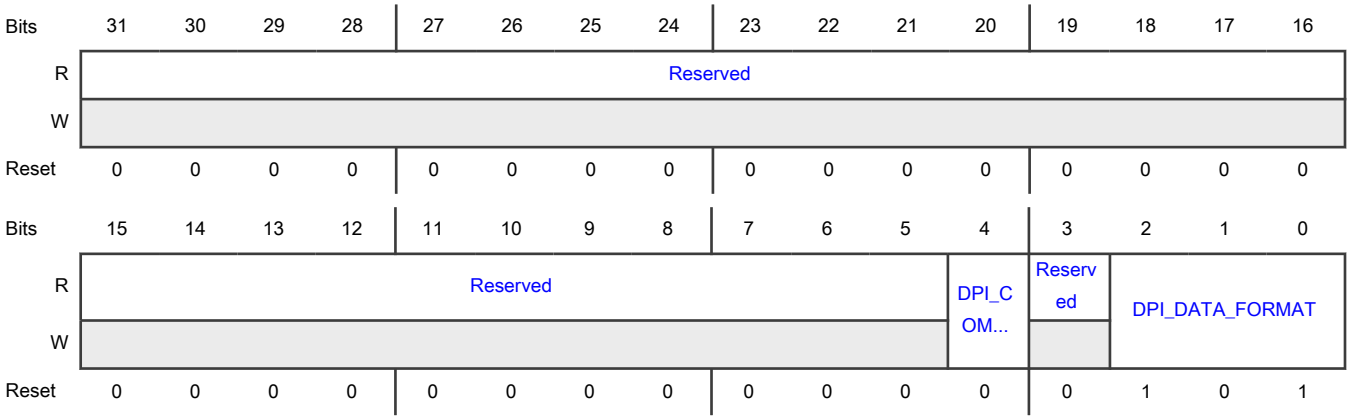
Function

Sets output format for DPI.

NOTE

This register is double buffered.

Diagram



Fields

Field	Function
31-5 —	Reserved
4	DPI Command Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
DPI_COMMAND_MODE	Sets the DPI command mode. 0b - Disables the DPI command mode. 1b - Enables the DPI command mode.
3 —	Reserved
2-0 DPI_DATA_FORMAT	DPI Data Format Specifies the DPI interface data format. 000b - D16CFG1 001b - D16CFG2 010b - D16CFG3 011b - D18CFG1 100b - D18CFG2 101b - D24

55.6.67 Source Endian Configuration (SrcConfigEndian)

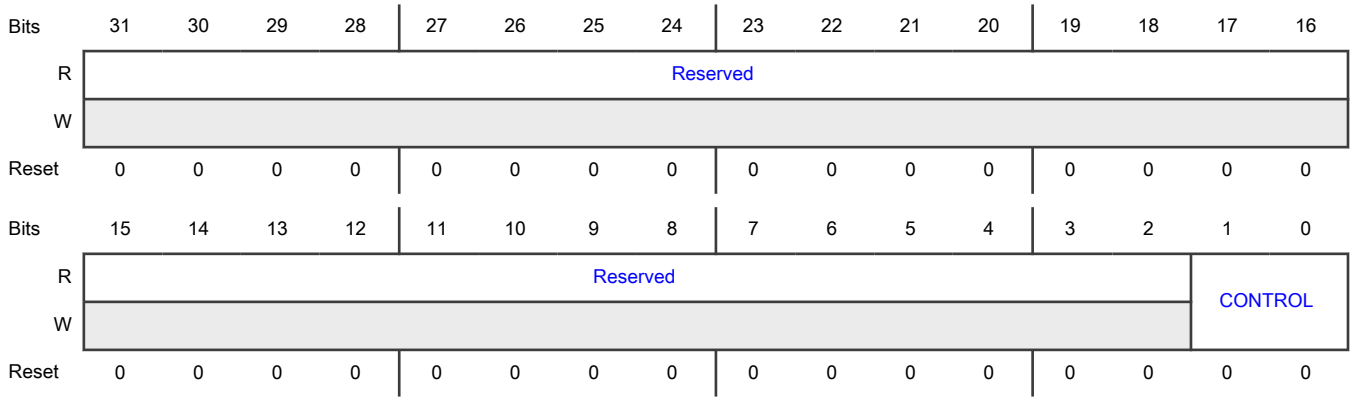
Offset

Register	Offset
SrcConfigEndian	2160h

Function

Sets source endian control.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 CONTROL	Control Controls source endian. 00b - No Swap. Swapping is not enabled. 01b - Swap Word. Swaps at 16-bit boundary: 0123 to 1032. 10b - Swap Dword. Swaps at 32-bit boundary: 0123 to 3210.

55.6.68 Soft Reset (SoftReset)

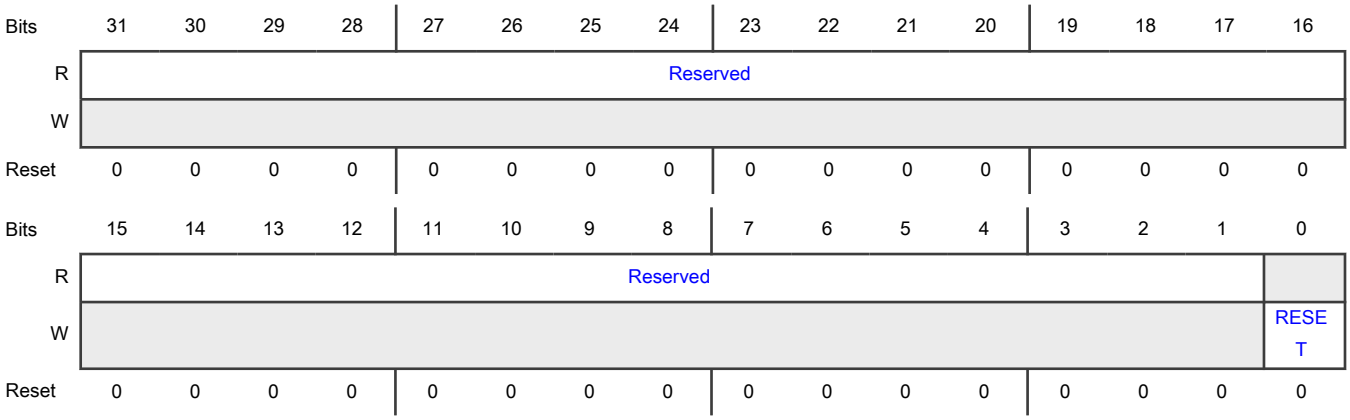
Offset

Register	Offset
SoftReset	2164h

Function

Enables or disables soft reset.

Diagram



Fields

Field	Function
31-1 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 RESET	Reset Display Controller Enables or disables soft reset of the display controller. 0b - Disable 1b - Enable

55.6.69 Display Controller Control (DcControl)

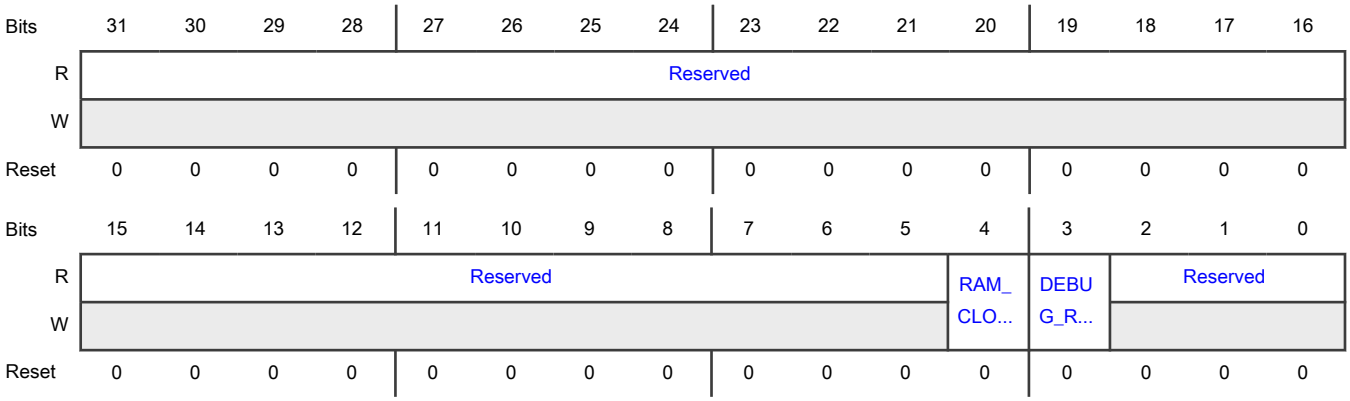
Offset

Register	Offset
DcControl	2168h

Function

Sets control for display controller.

Diagram



Fields

Field	Function
31-5 —	Reserved
4 RAM_CLOCK_GATING	Enables or disables RAM clock gating. 0b - Disables RAM clock gating. 1b - Enables RAM clock gating.
3	Debug

Table continues on the next page...

Table continued from the previous page...

Field	Function
DEBUG_REGISTER	Enables or disables debug. 0b - Disables debug. 1b - Enables debug.
2-0 —	Reserved

55.6.70 Layer Clock Gating (LayerClockGate)

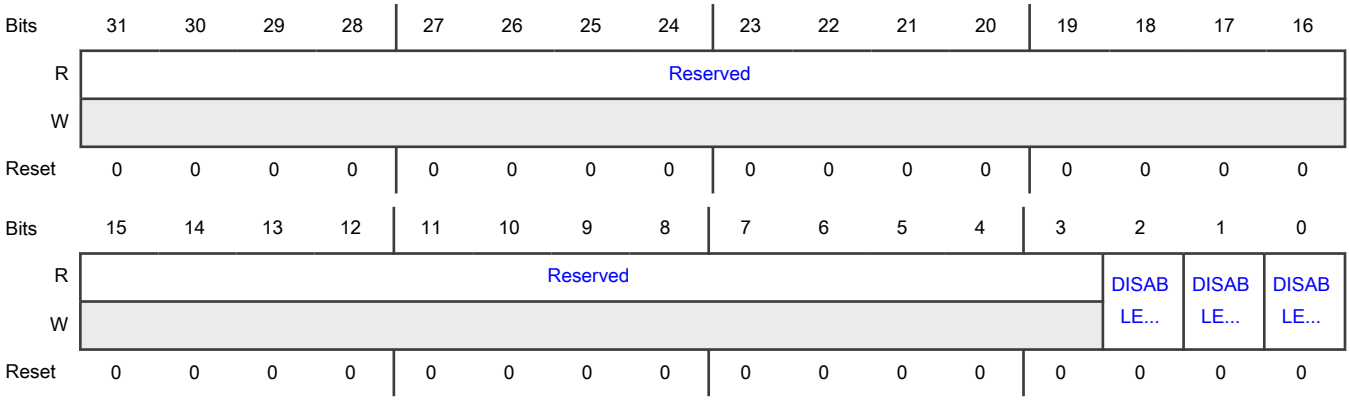
Offset

Register	Offset
LayerClockGate	21A0h

Function

Sets clock gating on the VG, overlay 0, and overlay 1 layers.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 DISABLE_OVERLAY1_CLK	Disable Overlay 1 Clock Enables or disables clock gating on the overlay1 layer.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables clock gating on the overlay 1 layer. 1b - Enables clock gating on the overlay 1 layer.
1 DISABLE_OVERLAY0_CLK	Disable Overlay 0 Clock Enables or disables clock gating on the overlay0 layer. 0b - Disables clock gating on the overlay 0 layer. 1b - Enables clock gating on the overlay 0 layer.
0 DISABLE_VIDEO_CLK	Disable Video Clock Enables or disables clock gating on the VG layer. 0b - Disables clock gating on the VG layer. 1b - Enables clock gating on the VG layer.

Chapter 56
Sigma Delta Analog-to-Digital Converter (SDADC)

56.1 Chip-specific SDADC information

Table 554. Reference links to related information

Topic	Related module	Reference
Full description	SDADC	SDADC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC

56.1.1 Module instances

This device contains one instance of the SDADC module.

Table 555. SDADC instance

Features	SDADC
Located in	VDD1_SENSE
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

56.1.2 SDADC input connections

See table below for SDADC input connections.

Table 556. SDADC channel mapping

SDADC channel	SDADC0 INP connection	SDADC INN connection
0	SDADC_INP0 (PIO10_6)	SDADC_INN0 (PIO10_7)
1	SDADC_INP1 (PIO10_12)	SDADC_INN1 (PIO10_13)
2	SDADC_INP2 (PIO10_14)	SDADC_INN2 (PIO10_15)
3	SDADC_INP3 (PIO10_16)	SDADC_INN3 (PIO10_17)

56.2 Overview

SDADC is a low-power high-performance audio ADC system. This module mainly targets battery-powered, hearable and wearable audio devices. It supports analog microphones and line inputs. This module also includes a low-latency decimation filter to enable acoustic noise and echo cancellation.

56.2.1 Block diagram

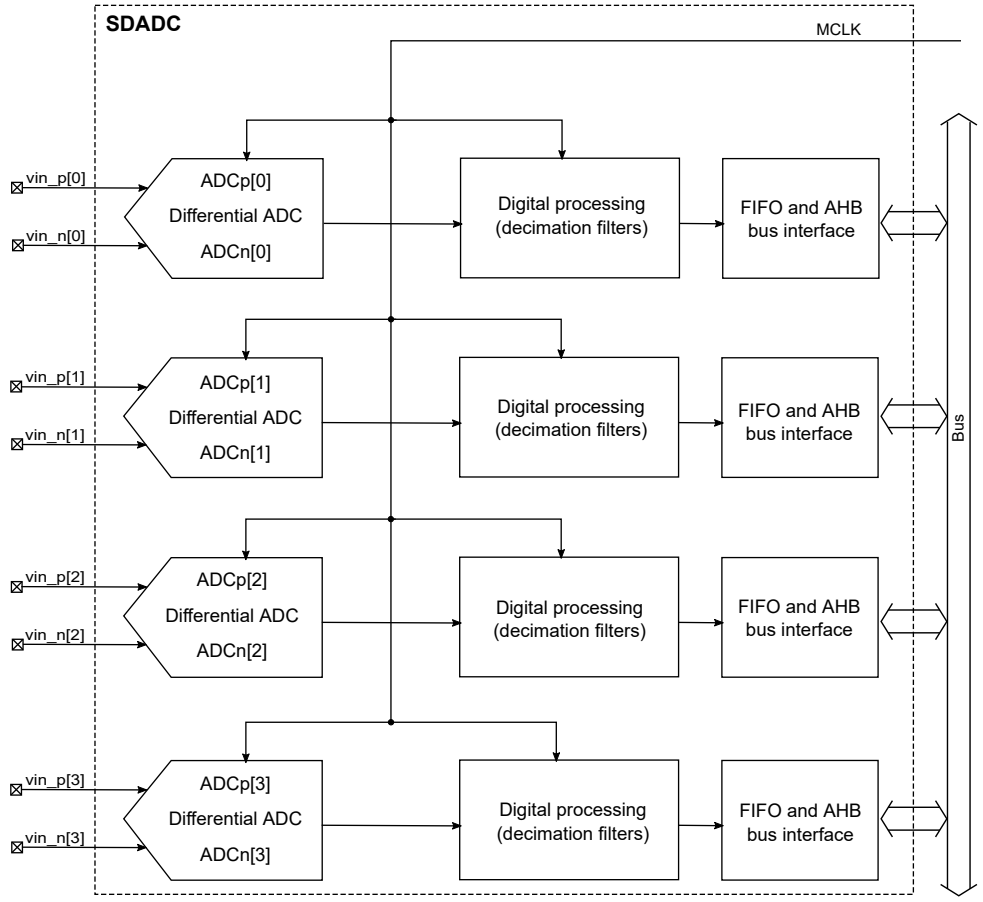


Figure 237. SDADC block diagram

56.2.2 Features

- Low-noise ADC in differential (Diff) mode
- Low-power modulator
- High-linearity ADC
- Supports sampling frequencies (F_s) of 16 kHz, 32 kHz, and 48 kHz
- Supports single-ended (SE) and differential (Diff) inputs
- Low latency to enable acoustic noise and echo cancellation
- Supports MEMS and JFET analog microphones
- Supports line inputs
- Decimation filter soft-core included
- Digital volume control
- DC loop filter to enable DC coupling of analog microphone output to ADC input in SE mode

56.3 Functional description

SDADC is a low-power high-performance Audio Codec for battery-powered audio devices. The basic function of the module is to convert single-ended (SE) or differential (Diff) analog input signals into 24-bit data word length with F_s Nyquist rate of 48 kHz, 32 kHz, or 16 kHz. The ADC is based on a 3rd-order continuous-time sigma-delta modulator (SDM). Single-unit ADC can be configured as two single-ended ADCs or one differential ADC. LDO is included in the module to isolate the sensitive analog circuit. Decimation filter is included to enable acoustic noise and echo cancellation.

56.3.1 ADC modulator

The ADC modulator is based on a quasi-passive 3rd-order continuous-time sigma-delta modulator (SDM) and is clocked with frequency of 512 × 48 kHz (24.576 MHz). The SDM bitstream output is downsampled by the decimation filter to a F_s Nyquist of 48 kHz, 32 kHz, or 16 kHz.

SDM has DC feedback loop to remove any DC offset when DC coupling is used to connect input source to V_{in} in single-ended (SE) mode only. This feature allows single-ended audio sampling without the use of a very large coupling capacitor, C_c, as in typical implementations. This feedback loop is controlled by DC loop filter soft-core block.

The complete ADC modulator submodule contains two single-ended modulators (ADCp and ADCn) which can support one differential ADC or two single-ended ADCs.

In differential (Diff) mode, differential input signals are applied to vin_p and vin_n. In this mode, only ADCp will generate bitstream and feed the associated decimation filter.

56.3.2 Decimation filter

The decimation filter submodule is a 24-bit mono decimation filter with 16-bit coefficients. The decimation is realized with programmable cascade integrator comb (CIC) filter and a cascade of 2 half-band filters and a finite impulse response (FIR) filter. It supports sampling frequencies (F_s) of 16 kHz, 32 kHz, and 48 kHz.

The decimator is clocked at a speed of 24.576 MHz. DC blocking filter is included at the front. When it is enabled it will remove unwanted DC components from the input bitstream like offset and DC-dithering. The DC blocking filter is implemented as a 1st order infinite impulse response (IIR) high-pass filter.

The features of the decimation filter are:

- Selectable sample rates: 16 kHz, 32 kHz, 48 kHz.
- DC filter
- Overflow signal
- Volume control with has a gain range from -63.5 dB to +24 dB in step of 0.5 dB.

56.3.3 DC input common mode control block

This is part of feedback loop that takes care of the input common-mode voltage when a signal source is connected directly to ADC input. This is only applicable in SE mode.

56.3.4 LDO

The LDO has supply voltage of 1.8 V and output voltage of 1.5 V. It requires clean 1.5 V (nominal) reference voltage, which comes from the reference generator. It can support up to 4 differential ADC modulator units.

56.3.5 Reference generator

The reference generator block consists of on-chip resistive divider (5/6) which divides the supply voltage of 1.8 V. When it is enabled, the reference generator generates a reference voltage (vref_ldo) of 1.5 V. External capacitor (C_{ext}) with recommended value of 25 µF is needed to suppress noise coming from supply.

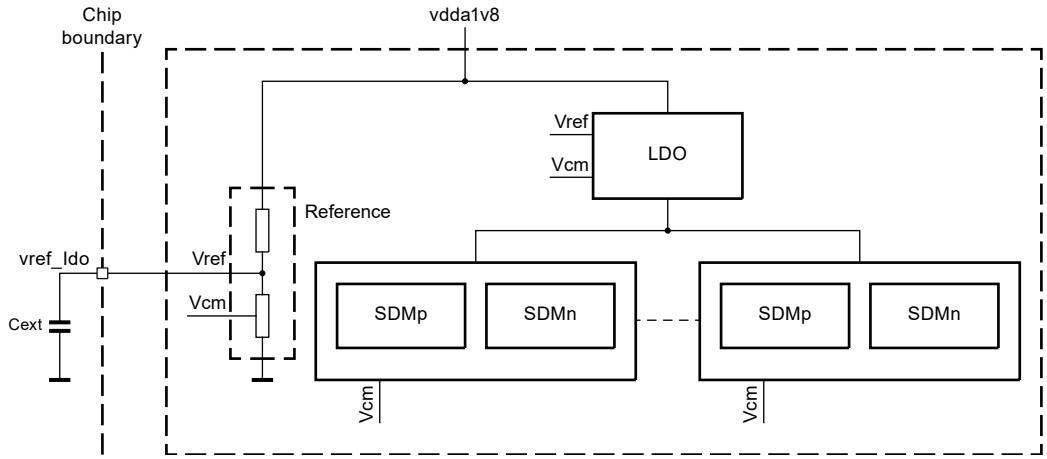


Figure 238. Reference circuit

56.3.6 Modes of operation

SDADC has 4 functional modes.

Table 557. Modes of operation

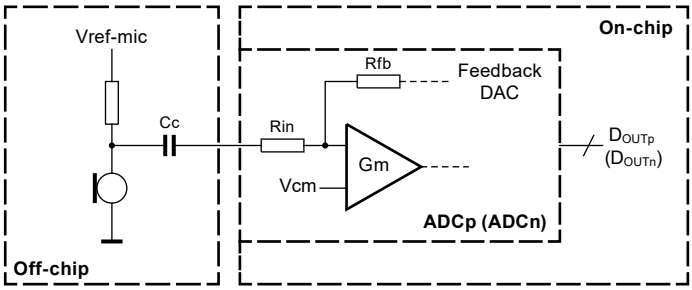
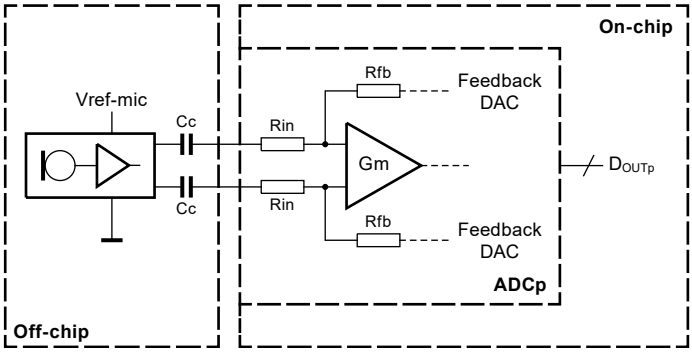
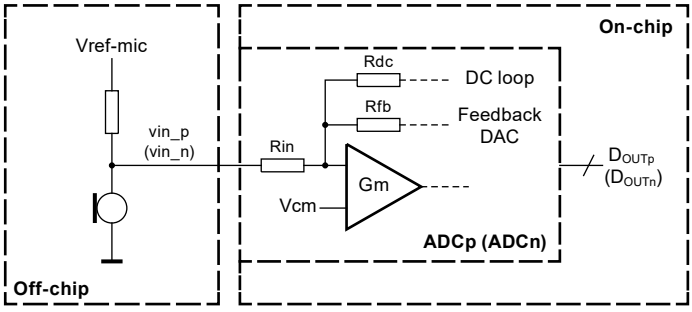
Operation modes	Configuration	Definition
SE Analog microphone input with AC coupling	Single-ended input mode	Microphone output is connected to ADC input via coupling capacitor C_c[1]. Each active decimator filter will output data.  Figure 239. SE mode with AC coupling
Differential Analog input with AC coupling	Differential input mode	Differential analog inputs are connected to ADC inputs via coupling capacitor C_c. One of the two decimators connected to the ADC modulators generates the output data.

Table continues on the next page...

Table 557. Modes of operation (continued)

Operation modes	Configuration	Definition
		 Figure 240. Diff mode with AC coupling
SE Analog microphone input with DC coupling	Single-ended input mode	Microphone output is connected directly to ADC input. DC loop is activated to regulate input common-mode voltage. Each active decimator will output data.  Figure 241. SE mode with DC coupling
Power-down mode		ADC is powered down.

1. C_c values typically range from 0.4 µF to 1 µF based on the required low cut-off frequency of the audio signal:

   $F_{cut\text{-}off} = 1/(2*\pi*R_{in}*C_c)$

   Typical R_{in} of ADC is 20 kΩ, so signal frequency below $F_{cut\text{-}off}$ will be attenuated by at least 3 dB.

56.3.7 Clocking

Table 558. SDADC clocks

Clock name	Description
Main clock (MCLK)	Used as system clock for SDADC. When this clock is gated, SDADC stops working. Frequency: 24.576 MHz
APB bus clock (APB_CLK)	Used for register interface.

56.3.8 Interrupts

This section describes all the interrupts that the SDADC module generates.

There are 8 WATERMARK register fields for each one of the FIFOs:

- WATERMARK_0_P
- WATERMARK_0_N
- WATERMARK_1_P
- WATERMARK_1_N
- WATERMARK_2_P
- WATERMARK_2_N
- WATERMARK_3_P
- WATERMARK_3_N

Each one of the FIFOs will assert their respective WM_x_y_ST signal if the entries in the respective FIFO are greater than the programmed value and WM_x_y_EN is enabled.

FIFO	Watermark enable	Interrupt enable
FIFO_0_P	FIFO_WATERMARK_ERROR_CTL[WM_0_P_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_0_P_EN]
FIFO_0_N	FIFO_WATERMARK_ERROR_CTL[WM_0_N_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_0_N_EN]
FIFO_1_P	FIFO_WATERMARK_ERROR_CTL[WM_1_P_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_1_P_EN]
FIFO_1_N	FIFO_WATERMARK_ERROR_CTL[WM_1_N_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_1_N_EN]
FIFO_2_P	FIFO_WATERMARK_ERROR_CTL[WM_2_P_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_2_P_EN]
FIFO_2_N	FIFO_WATERMARK_ERROR_CTL[WM_2_N_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_2_N_EN]
FIFO_3_P	FIFO_WATERMARK_ERROR_CTL[WM_3_P_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_3_P_EN]
FIFO_3_N	FIFO_WATERMARK_ERROR_CTL[WM_3_N_EN]	FIFO_WATERMARK_ERROR_CTL[ERR_3_N_EN]

The interrupt register bit will be set (if FIFO_INTERRUPT[FIFO_INTERRUPT_EN] is enabled) as a result of ORing ERR_0_P, ERR_0_N, ERR_1_P, ERR_1_N, ERR_2_P, ERR_2_N, ERR_3_P, and ERR_3_N.

For additional details, see SDADC register descriptions.

56.3.9 Reset

Table 559. SDADC internal generated resets

Reset	Description
ADC_CTL_0[RST_AN_ADCP]	Reset not signal (asynchronous) for ADCp
ADC_CTL_0[RST_AN_ADCN]	Reset not signal (asynchronous) for ADCn
DECIMATOR_CTL_1[FUNC_LOCAL_RST_AN_DEC_P]	Functional local reset not (asynchronous) for Decimator_p
DECIMATOR_CTL_1[FUNC_LOCAL_RST_AN_DEC_N]	Functional local reset not (asynchronous) for Decimator_n

56.3.10 DMA

When the watermark feature is enabled (FIFO_WATERMARK_ERROR_CTL[WM_x_y_EN] = 1b), and the entries in each one of the FIFOs meet the watermark value, SDADC asserts dma_req_fifo_x (x = 0, 1, 2, 3). At that point SDADC stops storing entries

in the FIFOs until the chip or the entity above asserts dma_req_ack_fifo_x. Once dma_req_ack_fifo_x is asserted and received by SDADC, the dms_req_fifo_x is de-asserted and the FIFOs start collecting entries again.

56.4 External signals

SDADC requires supply and ground connections, and supports analog channel inputs.

Table 560. SDADC signals

Signal	Description	I/O
vdda1v8	Analog power supply	Analog IO
VSSA	Analog ground	Analog IO
VDD	Digital power supply	Analog IO
VSS	Digital ground	Analog IO
vin_p[3:0]	ADCp[3:0] input voltage	Analog input
vin_n[3:0]	ADCn[3:0] input voltage	Analog input
vref_ldo	Used to attach filter capacitor (with a recommended size of 25 µF) to filter internally generated reference voltage (Vref) (1.5 V). See Reference generator. **NOTE** Keep this pin away from any noisy signal sources and use audio-quality capacitors.	Analog IO

56.5 Initialization

An initial reset is needed for 15 µs after start-up.

Coming out of reset, the four ADC modules are enabled. Default values are loaded into the respective registers.

56.6 Application information

56.6.1 Enabling ADCp[0] and ADCn[1] in SE mode at startup

Configure the following to enable ADCp[0] and ADCn[1] in single-ended mode at startup.

Action	Programming
Enable ADCp[0] and apply input to vin_p[0].	Write 0001b to ADC_CTL_0[ADCP_POWER_ENABLE].
Enable ADCn[1] and apply input to vin_n[1].	Write 0010b to ADC_CTL_0[ADCN_POWER_ENABLE].
Set all to SE mode.	Write 0000b to ADC_CTL_1[DIFF_ENABLE].
Enable compensation DACs for all ADCp.	Write 1111b to ADC_CTL_0[COMP_ADCP_ENABLE].
Enable compensation DACs for all ADCn.	Write 1111b to ADC_CTL_0[COMP_ADCN_ENABLE].
Assert first reset (0000b) for 15 µs after reference start.	Write 1111b to ADC_CTL_0[RST_AN_ADCP]. Write 1111b to ADC_CTL_0[RST_AN_ADCN].

Table continues on the next page...

Table continued from the previous page...

Action	Programming
Make sure inputs for all ADCp and ADCn are AC-coupled.	Write 0000b to ADC_CTL_1[DCLOOP_P_ANA_ENABLE]. Write 0000b to ADC_CTL_1[DCLOOP_N_ANA_ENABLE]. Write 0000b to DC_LOOP_CTL_0[DCLOOP_P_ENABLE]. Write 0000b to DC_LOOP_CTL_0[DCLOOP_N_ENABLE].
Enable LDO.	Write 1b to LDO_CTL[LDO_POWER_ENABLE].
Set LDO to full performance.	Write 01b to LDO_CTL[LDO_SETTING].
Enable reference block.	Write 1b to REF_CTL[REF_POWER_ENABLE].
Enable reference low-noise mode.	Write 0b to REF_CTL[FAST_REF_ENABLE]. Write 1b to REF_CTL[PWR_FAST_REF_ENABLE].

56.7 Memory map and register definition

56.7.1 SDADC register descriptions

56.7.1.1 SDADC memory map

SDADC base address: 4020_D000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0100_0000h
4h	ADC Control 0 (ADC_CTL_0)	32	RW	0000_FFFFh
8h	ADC Control 1 (ADC_CTL_1)	32	RW	0000_0000h
Ch	ADC Control 2 (ADC_CTL_2)	32	RW	0000_0000h
10h	LDO Control (LDO_CTL)	32	RW	0000_1300h
14h	Reference Control (REF_CTL)	32	RW	0011_0000h
1Ch	Decimator Control 0 (DECIMATOR_CTL_0)	32	RW	FF00_0000h
20h	Decimator Control 1 (DECIMATOR_CTL_1)	32	RW	0000_FF00h
24h	Decimator Control 2 (DECIMATOR_CTL_2)	32	RW	0000_0000h
28h	Decimator Control 3 (DECIMATOR_CTL_3)	32	RW	0000_0000h
2Ch	Decimator Control 4 (DECIMATOR_CTL_4)	32	RW	0000_0000h
30h	DC Loop Control 0 (DC_LOOP_CTL_0)	32	RW	0000_0000h
34h	DC Loop Control 1 (DC_LOOP_CTL_1)	32	RW	0000_00FFh
38h	FIFO Write Read Control (FIFO_WR_RD_CTL)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
3Ch	FIFO Watermark Control (FIFO_WATERMARK_CTL)	32	RW	0000_0000h
40h	FIFO Watermark Error Control (FIFO_WATERMARK_ERROR_CTL)	32	RW	0000_0000h
44h	FIFO Watermark Error Status (FIFO_WATERMARK_ERROR_ST)	32	R	0000_0000h
48h	FIFO 0 P (FIFO_0_P)	32	R	0000_0000h
4Ch	FIFO 0 N (FIFO_0_N)	32	R	0000_0000h
50h	FIFO 1 P (FIFO_1_P)	32	R	0000_0000h
54h	FIFO 1 N (FIFO_1_N)	32	R	0000_0000h
58h	FIFO 2 P (FIFO_2_P)	32	R	0000_0000h
5Ch	FIFO 2 N (FIFO_2_N)	32	R	0000_0000h
60h	FIFO 3 P (FIFO_3_P)	32	R	0000_0000h
64h	FIFO 3 N (FIFO_3_N)	32	R	0000_0000h
6Ch	FIFO Entries Availability 0 (FIFO_ENTRIES_AVAIL_0)	32	R	0000_0000h
70h	FIFO Entries Availability 1 (FIFO_ENTRIES_AVAIL_1)	32	R	0000_0000h
74h	FIFO Interrupt (FIFO_INTERRUPT)	32	RW	0000_0000h

56.7.1.2 Version ID (VERID)

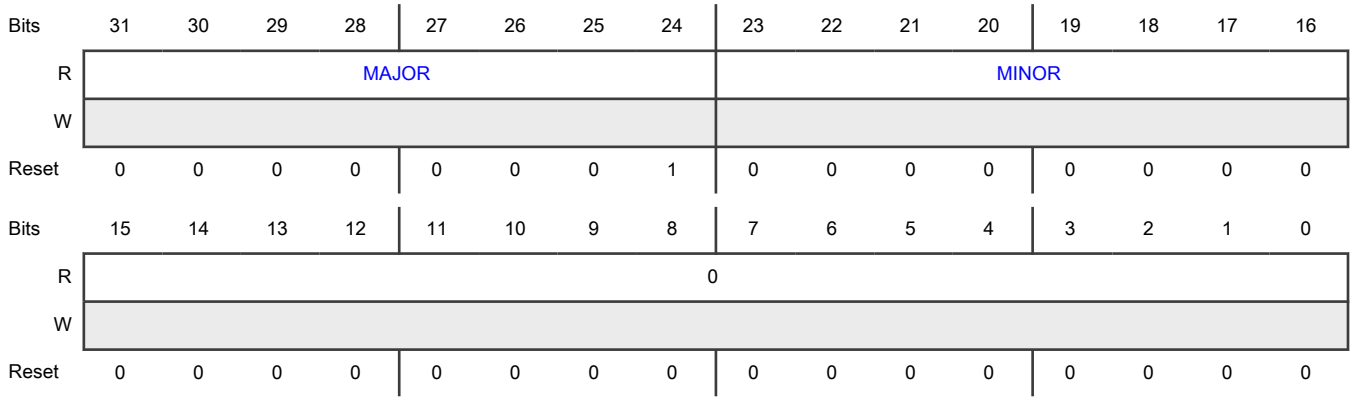
Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the device and also indicates inclusion or exclusion of several optional features.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module specification.
15-0 —	Reserved

56.7.1.3 ADC Control 0 (ADC_CTL_0)

Offset

Register	Offset
ADC_CTL_0	4h

Function

Controls power, resets and compensation feedback DAC for ADC.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADCP_POWER_ENABLE				ADCN_POWER_ENABLE				0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RST_AN_ADCP				RST_AN_ADCN				COMP_ADCP_ENABLE				COMP_ADCN_ENABLE			
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-28 ADCP_POWER_ENABLE	ADCp Power Enable Enables each of the four ADCp[3:0] and their respective inputs vin_p[3:0] in single-ended (SE) or differential (Diff) modes. **NOTE** In differential mode, the inputs are in pair: (vin_p[3], vin_n[3]), (vin_p[2], vin_n[2]), (vin_p[1], vin_n[1]), and (vin_p[0], vin_n[0]). Therefore, in Diff mode both inputs of the input pair should be enabled. For example: ADCP_POWER_ENABLE[3:0]=0001b and ADCN_POWER_ENABLE[3:0]=0001b will enable input pair vin_p[0] and vin_n[0]. 0xxxb - Disables ADCp[3] and vin_p[3] 1xxxb - Enables ADCp[3] and vin_p[3] x0xxb - Disables ADCp[2] and vin_p[2] x1xxb - Enables ADCp[2] and vin_p[2] xx0xb - Disables ADCp[1] and vin_p[1] xx1xb - Enables ADCp[1] and vin_p[1] xxx0b - Disables ADCp[0] and vin_p[0] xxx1b - Enables ADCp[0] and vin_p[0]
27-24 ADCN_POWER_ENABLE	ADCn Power Enable Enables each of the four single-ended ADCn[3:0] and their respective inputs vin_n[3:0] in single-ended (SE) or differential (Diff) modes. **NOTE** In differential mode, the inputs are in pair: (vin_p[3], vin_n[3]), (vin_p[2], vin_n[2]), (vin_p[1], vin_n[1]), and (vin_p[0], vin_n[0]). Therefore, in Diff mode both inputs of the input pair should be enabled. For example: ADCP_POWER_ENABLE[3:0]=0001b and ADCN_POWER_ENABLE[3:0]=0001b will enable input pair vin_p[0] and vin_n[0]. 0xxxb - Disables ADCn[3] and vin_n[3]

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1xxxb - Enables ADCn[3] and vin_n[3] x0xxb - Disables ADCn[2] and vin_n[2] x1xxb - Enables ADCn[2] and vin_n[2] xx0xb - Disables ADCn[1] and vin_n[1] xx1xb - Enables ADCn[1] and vin_n[1] xxx0b - Disables ADCn[0] and vin_n[0] xxx1b - Enables ADCn[0] and vin_n[0]
23-20 —	Reserved
19-16 —	Reserved
15-12 RST_AN_ADCP	Reset ADCp Reset not signal (asynchronous) for ADCp[3:0]. Needs to be reset for at least 15 µs after start-up. 0xxxb - Reset for ADCp[3] 1xxxb - Operational for ADCp[3] x0xxb - Reset for ADCp[2] x1xxb - Operational for ADCp[2] xx0xb - Reset for ADCp[1] xx1xb - Operational for ADCp[1] xxx0b - Reset for ADCp[0] xxx1b - Operational for ADCp[0]
11-8 RST_AN_ADCN	Reset ADCn Reset not signal (asynchronous) for ADCn[3:0]. Needs to be reset for at least 15 µs after start-up. 0xxxb - Reset for ADCn[3] 1xxxb - Operational for ADCn[3] x0xxb - Reset for ADCn[2] x1xxb - Operational for ADCn[2] xx0xb - Reset for ADCn[1] xx1xb - Operational for ADCn[1] xxx0b - Reset for ADCn[0] xxx1b - Operational for ADCn[0]
7-4	Compensation ADCp Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
COMP_ADCP_ENABLE	Enables compensation feedback DAC of each of the four ADCp[3:0] in single-ended (SE) mode. Compensation feedback DAC is used to avoid data-dependent ripple on ADC supply net in single-ended mode. Disabling this function will result in reduced performance of the ADC. **NOTE** In Diff mode, this setting will not take any effect. 0xxxb - Disables for ADCp[3] 1xxxb - Enables for ADCp[3] x0xxb - Disables for ADCp[2] x1xxb - Enables for ADCp[2] xx0xb - Disables for ADCp[1] xx1xb - Enables for ADCp[1] xxx0b - Disables for ADCp[0] xxx1b - Enables for ADCp[0]
3-0 COMP_ADCN_ENABLE	Compensation ADCn Enable Enables compensation feedback DAC of each of the four ADCn[3:0] in single-ended (SE) mode. Compensation feedback DAC is used to avoid data-dependent ripple on ADC supply net in single-ended mode. Disabling this function will result in reduced performance of the ADC. **NOTE** In Diff mode this setting will not take any effect. 0xxxb - Disables for ADCn[3] 1xxxb - Enables for ADCn[3] x0xxb - Disables for ADCn[2] x1xxb - Enables for ADCn[2] xx0xb - Disables for ADCn[1] xx1xb - Enables for ADCn[1] xxx0b - Disables for ADCn[0] xxx1b - Enables for ADCn[0]

56.7.1.4 ADC Control 1 (ADC_CTL_1)

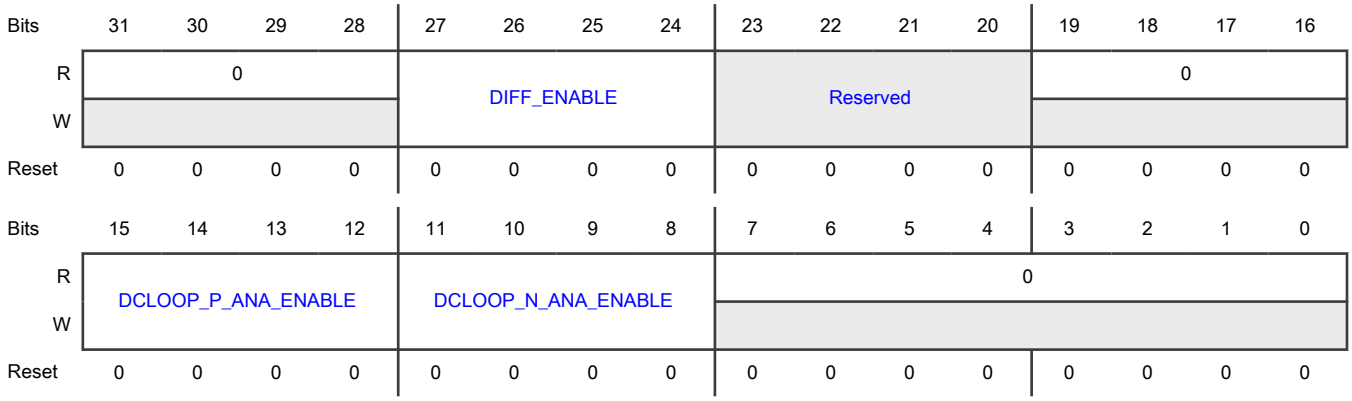
Offset

Register	Offset
ADC_CTL_1	8h

Function

Controls Diff mode, SE mode and DC loop functionalities for ADC.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-24 DIFF_ENABLE	Diff Enable Enables Diff mode of each of the four ADCp[3:0]. If for example ADCp[3] and ADCn[3] are configured as differential pair input, then the input differential pair is vin_p[3] and vin_n[3], and so on. 0xxxb - ADCp[3] and ADCn[3] in SE mode 1xxxb - Configures ADCp[3] and ADCn[3] as differential pair input x0xxb - ADCp[2] and ADCn[2] in SE mode x1xxb - Configures ADCp[2] and ADCn[2] as differential pair input xx0xb - ADCp[1] and ADCn[1] in SE mode xx1xb - Configures ADCp[1] and ADCn[1] as differential pair input xxx0b - ADCp[0] and ADCn[0] in SE mode xxx1b - Configures ADCp[0] and ADCn[0] as differential pair input
23-20 —	Reserved
19-16 —	Reserved
15-12 DCLOOP_P_ANA_ENABLE	DC Loop P Analog Enable Enables DC loop functionality of each of the four ADCp[3:0] in single-ended (SE) mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE In order to enable this functionality the corresponding DCLOOP_P_ENABLE setting in DC loop filter block should be enabled as well. Disable this setting in Diff mode. 0xxxb - Disables for ADCp[3] 1xxxb - Enables for ADCp[3] x0xxb - Disables for ADCp[2] x1xxb - Enables for ADCp[2] xx0xb - Disables for ADCp[1] xx1xb - Enables for ADCp[1] xxx0b - Disables for ADCp[0] xxx1b - Enables for ADCp[0]
11-8 DCLOOP_N_ANA_ENABLE	DC Loop N Analog Enable Enables DC loop functionality of each of the four ADCn[3:0] in single-ended (SE) mode. **NOTE** In order to enable this functionality the corresponding DCLOOP_N_ENABLE setting in DC loop filter block should be enabled as well. Disable this setting in Diff mode. 0xxxb - Disables for ADCn[3] 1xxxb - Enables for ADCn[3] x0xxb - Disables for ADCn[2] x1xxb - Enables for ADCn[2] xx0xb - Disables for ADCn[1] xx1xb - Enables for ADCn[1] xxx0b - Disables for ADCn[0] xxx1b - Enables for ADCn[0]
7-0 —	Reserved

56.7.1.5 ADC Control 2 (ADC_CTL_2)

Offset

Register	Offset
ADC_CTL_2	Ch

Function

Controls dither settings (0 - 3) for ADC.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SET_DITHER0			0	SET_DITHER1			0	SET_DITHER2			0	SET_DITHER3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-28 SET_DITHER0	Set Dither 0 Sets dither amplitude of ADCp[0] and ADCn[0]. Dither is used if ADC suffers from idle tones. This setting is to set how strong the dither to be introduced. 000b - Disables dither of ADCp[0] and ADCn[0] 001b-110b - The higher the value, the stronger the dither 111b - Maximum dither amplitude for ADCp[0] and ADCn[0]
27 —	Reserved
26-24 SET_DITHER1	Set Dither 1 Sets dither amplitude of ADCp[1] and ADCn[1]. 000b - Disables dither of ADCp[1] and ADCn[1] 001b-110b - The higher the value, the stronger the dither 111b - Maximum dither amplitude for ADCp[1] and ADCn[1]
23 —	Reserved
22-20 SET_DITHER2	Set Dither 2 Sets dither amplitude of ADCp[2] and ADCn[2].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	000b - Disables dither of ADCp[2] and ADCn[2] 001b-110b - The higher the value, the stronger the dither 111b - Maximum dither amplitude for ADCp[2] and ADCn[2]
19 —	Reserved
18-16 SET_DITHER3	Set Dither 3 Sets dither amplitude of ADCp[3] and ADCn[3]. 000b - Disables dither of ADCp[3] and ADCn[3]. 001b-110b - The higher the value, the stronger the dither 111b - Maximum dither amplitude for ADCp[3] and ADCn[3]
15-12 —	Reserved
11-8 —	Reserved
7-4 —	Reserved
3-0 —	Reserved

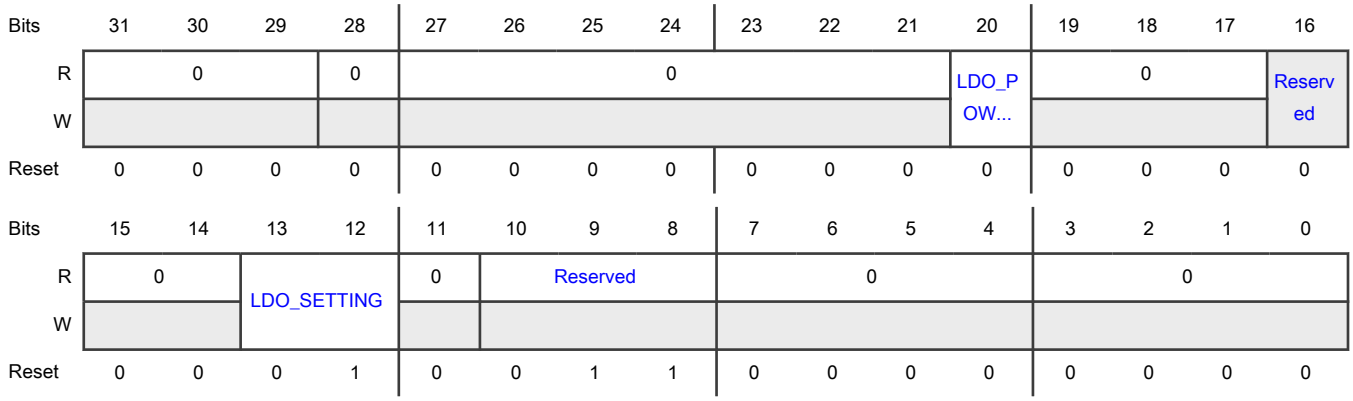
56.7.1.6 LDO Control (LDO_CTL)

Offset

Register	Offset
LDO_CTL	10h

Function
Controls Bypass mode, power and settings for LDO.

Diagram



Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27-21 —	Reserved
20 LDO_POWER_ENABLE	LDO Power Enable Enables LDO. 0b - Disables 1b - Enables
19-17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13-12 LDO_SETTING	LDO Setting Specifies LDO drive capability. For 1 single-ended (SE) or 1 differential (Diff) mode, setting 00b can be used to lower the current consumption. 00b - To drive 1 SE or 1 Diff mode ADC (low power mode) 01b - To drive more than 1 ADCs

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b,11b - Not in use
11 —	Reserved
10-8 —	Reserved
7-4 —	Reserved
3-0 —	Reserved

56.7.1.7 Reference Control (REF_CTL)

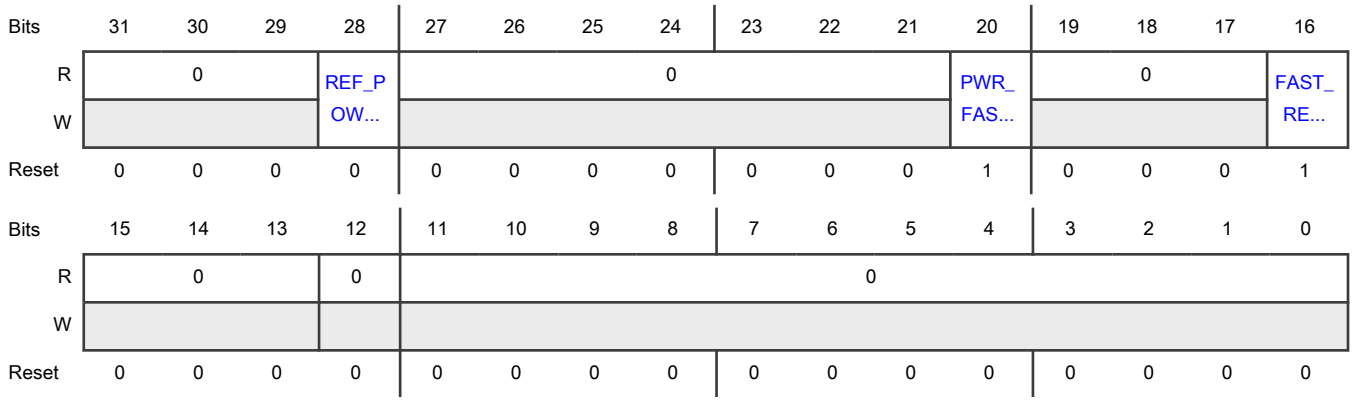
Offset

Register	Offset
REF_CTL	14h

Function

Implements control fields associated with reference.

Diagram



Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
28 REF_POWER_ENABLE	Reference Power Enable Enables reference generator block for LDO. 0b - Disables 1b - Enables
27-21 —	Reserved
20 PWR_FAST_REF_ENABLE	Power Fast Reference Enable PWR_FAST_REF_ENABLE and FAST_REF_ENABLE determines startup behaviour. 0b - 267 ms (retain mode) when bit16=0 | 66.5 ms (low power) when bit16=1 1b - 38 ms (low noise) when bit16=0 | 1 ms (fast charging) when bit16=1
19-17 —	Reserved
16 FAST_REF_ENABLE	Fast Reference Enable PWR_FAST_REF_ENABLE and FAST_REF_ENABLE determines startup behaviour. 0b - 267 ms (retain mode) when bit20=0 | 38 ms (low noise) when bit20=1 1b - 66.5 ms (low power) when bit20=0 | 1 ms (fast charging) when bit20=1
15-13 —	Reserved
12 —	Reserved
11-0 —	Reserved

56.7.1.8 Decimator Control 0 (DECIMATOR_CTL_0)

Offset

Register	Offset
DECIMATOR_CTL_0	1Ch

Function

Controls DC blocking and polarity inverters for the decimator.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCFILTI_N_ENABLE				DCFILTI_P_ENABLE				0				0			
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POL_INV_N_ENABLE				POL_INV_P_ENABLE				0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 DCFILTI_N_ENABLE	Decimation Filter Input N Enable Enables DC blocking at input Decimator_n[3:0]. Removes unwanted DC components from the input bitstream like offset and DC-dithering. 0xxxb - Disables at Decimator_n[3] 1xxxb - Enables at Decimator_n[3] x0xxb - Disables at Decimator_n[2] x1xxb - Enables at Decimator_n[2] xx0xb - Disables at Decimator_n[1] xx1xb - Enables at Decimator_n[1] xxx0b - Disables at Decimator_n[0] xxx1b - Enables at Decimator_n[0]
27-24 DCFILTI_P_ENABLE	Decimation Filter Input P Enable Enables DC blocking at input Decimator_p[3:0]. It removes unwanted DC components from the input bitstream like offset and DC-dithering. 0xxxb - Disables at Decimator_p[3] 1xxxb - Enables at Decimator_p[3] x0xxb - Disables at Decimator_p[2] x1xxb - Enables at Decimator_p[2] xx0xb - Disables at Decimator_p[1] xx1xb - Enables at Decimator_p[1] xxx0b - Disables at Decimator_p[0] xxx1b - Enables at Decimator_p[0]
23-20	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
19-16 —	Reserved
15-12 POL_INV_N_ENABLE	Polarity Inverter N Enable Enables polarity inverter of Decimator_n[3:0]. It enables/disables decimator output signal polarity. For example: if the input to ADC is pure sinus signal, then if the polarity is enabled, the decimation output will be sinus signal with 180° out of phase with the input. 0xxxb - Disables at Decimator_n[3] 1xxxb - Enables at Decimator_n[3] x0xxb - Disables at Decimator_n[2] x1xxb - Enables at Decimator_n[2] xx0xb - Disables at Decimator_n[1] xx1xb - Enables at Decimator_n[1] xxx0b - Disables at Decimator_n[0] xxx1b - Enables at Decimator_n[0]
11-8 POL_INV_P_ENABLE	Polarity Inverter P Enable Enables polarity inverter of Decimator_p[3:0]. It enables/disables decimator output signal polarity. For example: if the input to ADC is pure sinus signal, then if the polarity is enabled, the decimation output will be sinus signal with 180° out of phase with the input. 0xxxb - Disables at Decimator_p[3] 1xxxb - Enables at Decimator_p[3] x0xxb - Disables at Decimator_p[2] x1xxb - Enables at Decimator_p[2] xx0xb - Disables at Decimator_p[1] xx1xb - Enables at Decimator_p[1] xxx0b - Disables at Decimator_p[0] xxx1b - Enables at Decimator_p[0]
7-4 —	Reserved
3-0 —	Reserved

56.7.1.9 Decimator Control 1 (DECIMATOR_CTL_1)

Offset

Register	Offset
DECIMATOR_CTL_1	20h

Function
Controls functional local reset for the decimator.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FUNC_LOCAL_RST_AN_DEC_N				FUNC_LOCAL_RST_AN_DEC_P				0				0			
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-12 FUNC_LOCAL_RST_AN_DEC_N	Functional Local Reset for Decimator N Functional local reset not (asynchronous) for Decimator_n[3:0]. 0xxxb - Reset for Decimator_n[3] 1xxxb - Operational for Decimator_n[3] x0xxb - Reset for Decimator_n[2] x1xxb - Operational for Decimator_n[2] xx0xb - Reset for Decimator_n[1] xx1xb - Operational for Decimator_n[1] xxx0b - Reset for Decimator_n[0] xxx1b - Operational for Decimator_n[0]
11-8	Functional Local Reset for Decimator P Functional local reset not (asynchronous) for Decimator_p[3:0].

Table continues on the next page...

Table continued from the previous page...

Field	Function
FUNC_LOCAL_RST_AN_DEC_P	0xxxb - Reset for Decimator_p[3] 1xxxb - Operational for Decimator_p[3] x0xxb - Reset for Decimator_p[2] x1xxb - Operational for Decimator_p[2] xx0xb - Reset for Decimator_p[1] xx1xb - Operational for Decimator_p[1] xxx0b - Reset for Decimator_p[0] xxx1b - Operational for Decimator_p[0]
7-4 —	Reserved
3-0 —	Reserved

56.7.1.10 Decimator Control 2 (DECIMATOR_CTL_2)

Offset

Register	Offset
DECIMATOR_CTL_2	24h

Function

Controls gain for the decimator connected to ADCn[1:0] and ADCp[1:0]. Gain range from -63.5 dB to +24 dB in step of 0.5 dB, as shown in the table below.

Table 561. Volume control values

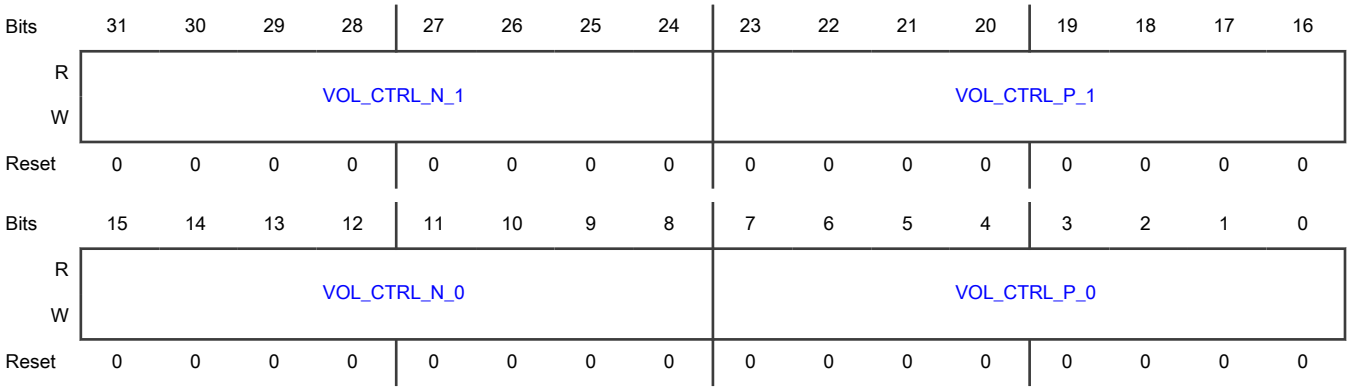
VOL_CTRL_x_y field value	Gain (dB)
0111_1111b	+24
...	...
0011_0001b	+24
0011_0000b	+24
0010_1111b	+23.5
...	...
0000_0010b	+1.0
0000_0001b	+0.5
0000_0000b	+0.0

Table continues on the next page...

Table 561. Volume control values (continued)

VOL_CTRL_x_y field value	Gain (dB)
1111_1111b	−0.5
1111_1110b	−1.0
...	...
1000_0010b	−63
1000_0001b	−63.5
1000_0000b	−∞

Diagram



Fields

Field	Function
31-24 VOL_CTRL_N_1	Volume Control N 1 Specifies gain for Decimator connected to ADCn[1] in SE mode. For field value definitions, see Table 561.
23-16 VOL_CTRL_P_1	Volume Control P 1 Specifies gain for Decimator connected to ADCp[1] in SE mode and Diff mode. For field value definitions, see Table 561.
15-8 VOL_CTRL_N_0	Volume Control N 0 Specifies gain for Decimator connected to ADCn[0] in SE mode. For field value definitions, see Table 561.
7-0 VOL_CTRL_P_0	Volume Control P 0 Specifies gain for Decimator connected to ADCp[0] in SE mode and Diff mode. For field value definitions, see Table 561.

56.7.1.11 Decimator Control 3 (DECIMATOR_CTL_3)

Offset

Register	Offset
DECIMATOR_CTL_3	28h

Function

Controls gain for the decimator connected to ADCn[3:2] and ADCp[3:2]. Gain range from -63.5 dB to +24 dB in step of 0.5 dB, as shown in the table below.

Table 562. Volume control values

VOL_CTRL_x_y field value	Gain (dB)
0111_1111b	+24
...	...
0011_0001b	+24
0011_0000b	+24
0010_1111b	+23.5
...	...
0000_0010b	+1.0
0000_0001b	+0.5
0000_0000b	+0.0
1111_1111b	−0.5
1111_1110b	−1.0
...	...
1000_0010b	−63
1000_0001b	−63.5
1000_0000b	−∞

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VOL_CTRL_N_3								VOL_CTRL_P_3							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VOL_CTRL_N_2								VOL_CTRL_P_2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 VOL_CTRL_N_3	Volume Control N 3 Specifies gain for Decimator connected to ADCn[3] in SE mode. For field value definitions, see Table 562.
23-16 VOL_CTRL_P_3	Volume Control P 3 Specifies gain for Decimator connected to ADCp[3] in SE mode and Diff mode. For field value definitions, see Table 562.
15-8 VOL_CTRL_N_2	Volume Control N 2 Specifies gain for Decimator connected to ADCn[2] in SE mode. For field value definitions, see Table 562.
7-0 VOL_CTRL_P_2	Volume Control P 2 Specifies gain for Decimator connected to ADCp[2] in SE mode and Diff mode. For field value definitions, see Table 562.

56.7.1.12 Decimator Control 4 (DECIMATOR_CTL_4)

Offset

Register	Offset
DECIMATOR_CTL_4	2Ch

Function
Controls the sample rates for the decimator.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SET_SPEED_N_3		SET_SPEED_P_3		SET_SPEED_N_2		SET_SPEED_P_2		SET_SPEED_N_1		SET_SPEED_P_1		SET_SPEED_N_0		SET_SPEED_P_0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-14 SET_SPEED_N_3	Set Speed N 3 Selectable sample rates for Decimator connected to ADCn[3] in SE mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
13-12 SET_SPEED_P_3	Set Speed P 3 Selectable sample rates for Decimator connected to ADCp[3] in SE mode and Diff mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
11-10 SET_SPEED_N_2	Set Speed N 2 Selectable sample rates for Decimator connected to ADCn[2] in SE mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
9-8 SET_SPEED_P_2	Set Speed P 2 Selectable sample rates for Decimator connected to ADCp[2] in SE mode and Diff mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
7-6	Set Speed N 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
SET_SPEED_N_1	Selectable sample rates for Decimator connected to ADCn[1] in SE mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
5-4 SET_SPEED_P_1	Set Speed P 1 Selectable sample rates for Decimator connected to ADCp[1] in SE mode and Diff mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
3-2 SET_SPEED_N_0	Set Speed N 0 Selectable sample rates for Decimator connected to ADCn[0] in SE mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz
1-0 SET_SPEED_P_0	Set Speed P 0 Selectable sample rates for Decimator connected to ADCp[0] in SE mode and Diff mode. 00b - 48 kHz 01b - 32 kHz 10b - 16 kHz 11b - 48 kHz

56.7.1.13 DC Loop Control 0 (DC_LOOP_CTL_0)

Offset

Register	Offset
DC_LOOP_CTL_0	30h

Function
Controls the digital DC loop filters.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0				0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				DCLOOP_N_ENABLE				DCLOOP_P_ENABLE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7-4 DCLOOP_N_ENABLE	DC Loop N Enable Enables digital DC LOOP filter_n[3:0] which is connected to ADCn[3:0]. **NOTE** This setting is only applicable in SE mode. Disable this setting in Diff mode. 0xxxb - Disables filter_n[3] 1xxxb - Enables filter_n[3] x0xxb - Disables filter_n[2] x1xxb - Enables filter_n[2] xx0xb - Disables filter_n[1]

Table continues on the next page...

Table continued from the previous page...

Field	Function
	xx1xb - Enables filter_n[1] xxx0b - Disables filter_n[0] xxx1b - Enables filter_n[0]
3-0 DCLOOP_P_ENABLE	DC Loop P Enable Enable or disable digital DC LOOP filter_p[3:0] which is connected to ADCp[3:0]. **NOTE** This setting is only applicable in SE mode. Disable this setting in Diff mode. 0xxxb - Disables filter_p[3] 1xxxb - Enables filter_p[3] x0xxb - Disables filter_p[2] x1xxb - Enables filter_p[2] xx0xb - Disables filter_p[1] xx1xb - Enables filter_p[1] xxx0b - Disables filter_p[0] xxx1b - Enables filter_p[0]

56.7.1.14 DC Loop Control 1 (DC_LOOP_CTL_1)

Offset

Register	Offset
DC_LOOP_CTL_1	34h

Function

Controls functional local resets for the DC loop filters.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				FUNC_LOCAL_RST_AN_DCLOOP_N				FUNC_LOCAL_RST_AN_DCLOOP_P			
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7-4 FUNC_LOCAL_RST_AN_DCLOOP_N	Functional Local Reset DCloop_n Functional local reset not (asynchronous) for DCloop_n. 0xxxb - Reset for ADCn[3] 1xxxb - Operational for ADCn[3] x0xxb - Reset for ADCn[2] x1xxb - Operational for ADCn[2] xx0xb - Reset for ADCn[1] xx1xb - Operational for ADCn[1] xxx0b - Reset for ADCn[0] xxx1b - Operational for ADCn[0]
3-0 FUNC_LOCAL_RST_AN_DCLOOP_P	Functional Local Reset DCloop_p Functional local reset no (asynchronous) for DCloop_p. 0xxxb - Reset for ADCp[3] 1xxxb - Operational for ADCp[3] x0xxb - Reset for ADCp[2] x1xxb - Operational for ADCp[2] xx0xb - Reset for ADCp[1] xx1xb - Operational for ADCp[1] xxx0b - Reset for ADCp[0] xxx1b - Operational for ADCp[0]

56.7.1.15 FIFO Write Read Control (FIFO_WR_RD_CTL)

Offset

Register	Offset
FIFO_WR_RD_CTL	38h

Function

Controls resets for FIFO read and writer pointers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RD_3_N_...	WR_3_N_...	RD_3_P_...	WR_3_P_...	RD_2_N_...	WR_2_N_...	RD_2_P_...	WR_2_P_...	RD_1_N_...	WR_1_N_...	RD_1_P_...	WR_1_P_...	RD_0_N_...	WR_0_N_...	RD_0_P_...	WR_0_P_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 RD_3_N_RST	Read Reset for FIFO_3_N When asserted, it resets the FIFO_3_N read pointer.
30 WR_3_N_RST	Write Reset for FIFO_3_N When asserted, it resets the FIFO_3_N write pointer.
29 RD_3_P_RST	Read Reset for FIFO_3_P When asserted, it resets the FIFO_3_P read pointer.
28 WR_3_P_RST	Write Reset for FIFO_3_P When asserted, it resets the FIFO_3_P write pointer.
27 RD_2_N_RST	Read Reset for FIFO_2_N When asserted, it resets the FIFO_2_N read pointer.
26 WR_2_N_RST	Write Reset for FIFO_2_N When asserted, it resets the FIFO_2_N write pointer.
25	Read Reset for FIFO_2_P

Table continues on the next page...

Table continued from the previous page...

Field	Function
RD_2_P_RST	When asserted, it resets the FIFO_2_P read pointer.
24 WR_2_P_RST	Write Reset for FIFO_2_P When asserted, it resets the FIFO_2_P write pointer.
23 RD_1_N_RST	Read Reset for FIFO_1_N When asserted, it resets the FIFO_1_N read pointer.
22 WR_1_N_RST	Write Reset for FIFO_1_N When asserted, it resets the FIFO_1_N write pointer.
21 RD_1_P_RST	Read Reset for FIFO_1_P When asserted, it resets the FIFO_1_P read pointer.
20 WR_1_P_RST	Write Reset for FIFO_1_P When asserted, it resets the FIFO_1_P write pointer.
19 RD_0_N_RST	Read Reset for FIFO_0_N When asserted, it resets the FIFO_0_N read pointer.
18 WR_0_N_RST	Write Reset for FIFO_0_N When asserted, it resets the FIFO_0_N write pointer.
17 RD_0_P_RST	Read Reset for FIFO_0_P When asserted, it resets the FIFO_0_P read pointer.
16 WR_0_P_RST	Write Reset for FIFO_0_P When asserted, it resets the FIFO_0_P write pointer.
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

56.7.1.16 FIFO Watermark Control (FIFO_WATERMARK_CTL)

Offset

Register	Offset
FIFO_WATERMARK_CTL	3Ch

Function
Specifies the value for FIFO watermark.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WATERMARK_3_N				WATERMARK_3_P				WATERMARK_2_N				WATERMARK_2_P			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WATERMARK_1_N				WATERMARK_1_P				WATERMARK_0_N				WATERMARK_0_P			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 WATERMARK_3_N	Watermark for FIFO_3_N Allowed values are 0-15. If the entries in FIFO_3_N are greater than the programmed value in this bit, and WM_3_N_EN is enabled, FIFO_3_N will assert WM_3_N_ST.
27-24 WATERMARK_3_P	Watermark for FIFO_3_P Allowed values are 0-15. If the entries in FIFO_3_P are greater than the programmed value in this bit, and WM_3_P_EN is enabled, FIFO_3_P will assert WM_3_P_ST.
23-20 WATERMARK_2_N	Watermark for FIFO_2_N Allowed values are 0-15. If the entries in FIFO_2_N are greater than the programmed value in this bit, and WM_2_N_EN is enabled, FIFO_2_N will assert WM_2_N_ST.
19-16 WATERMARK_2_P	Watermark for FIFO_2_P Allowed values are 0-15. If the entries in FIFO_2_P are greater than the programmed value in this bit, and WM_2_P_EN is enabled, FIFO_2_P will assert WM_2_P_ST.
15-12 WATERMARK_1_N	Watermark for FIFO_1_N Allowed values are 0-15. If the entries in FIFO_1_N are greater than the programmed value in this bit, and WM_1_N_EN is enabled, FIFO_1_N will assert WM_1_N_ST.
11-8 WATERMARK_1_P	Watermark for FIFO_1_P Allowed values are 0-15. If the entries in FIFO_1_P are greater than the programmed value in this bit, and WM_1_P_EN is enabled, FIFO_1_P will assert WM_1_P_ST.
7-4 WATERMARK_0_N	Watermark for FIFO_0_N Allowed values are 0-15. If the entries in FIFO_0_N are greater than the programmed value in this bit, and WM_0_N_EN is enabled, FIFO_0_N will assert WM_0_N_ST.
3-0 WATERMARK_0_P	Watermark for FIFO_0_P Allowed values are 0-15. If the entries in FIFO_0_P are greater than the programmed value in this bit, and WM_0_P_EN is enabled, FIFO_0_P will assert WM_0_P_ST.

56.7.1.17 FIFO Watermark Error Control (FIFO_WATERMARK_ERROR_CTL)

Offset

Register	Offset
FIFO_WATERMARK_ERROR_CTL	40h

Function
Controls the interrupts and watermarks for FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_3_N...	ERR_3_P...	ERR_2_N...	ERR_2_P...	ERR_1_N...	ERR_1_P...	ERR_0_N...	ERR_0_P...	WM_3_N_...	WM_3_P_...	WM_2_N_...	WM_2_P_...	WM_1_N_...	WM_1_P_...	WM_0_N_...	WM_0_P_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15 ERR_3_N_EN	Interrupt Enable for FIFO_3_N If enabled and FIFO_3_N is full, it will generate ERR_3_N used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
14 ERR_3_P_EN	Interrupt Enable for FIFO_3_P If enabled and FIFO_3_P is full, it will generate ERR_3_P used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
13 ERR_2_N_EN	Interrupt Enable for FIFO_2_N If enabled and FIFO_2_N is full, it will generate ERR_2_N used to generate FIFO_INTERRUPT. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
12 ERR_2_P_EN	Interrupt Enable for FIFO_2_P If enabled and FIFO_2_P is full, it will generate ERR_2_P used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
11 ERR_1_N_EN	Interrupt Enable for FIFO_1_N If enabled and FIFO_1_N is full, it will generate ERR_1_N used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
10 ERR_1_P_EN	Interrupt Enable for FIFO_1_P If enabled and FIFO_1_P is full, it will generate ERR_1_P used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
9 ERR_0_N_EN	Interrupt Enable for FIFO_0_N If enabled and FIFO_0_N is full, it will generate ERR_0_N, used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
8 ERR_0_P_EN	Interrupt Enable for FIFO_0_P If enabled and FIFO_0_P is full, it will generate ERR_0_P, used to generate FIFO_INTERRUPT. 0b - Disables 1b - Enables
7 WM_3_N_EN	Watermark Enable for FIFO_3_N If enabled, FIFO_3_N will assert WM_3_N_ST when the entries in FIFO_3_N are greater than the programmed value in WATERMARK_3_N. 0b - Disables 1b - Enables
6 WM_3_P_EN	Watermark Enable for FIFO_3_P If enabled, FIFO_3_P will assert WM_3_P_ST when the entries in FIFO_3_P are greater than the programmed value in WATERMARK_3_P. 0b - Disables 1b - Enables
5	Watermark Enable for FIFO_2_N

Table continues on the next page...

Table continued from the previous page...

Field	Function
WM_2_N_EN	If enabled, FIFO_2_N will assert WM_2_N_ST when the entries in FIFO_2_N are greater than the programmed value in WATERMARK_2_N. 0b - Disables 1b - Enables
4 WM_2_P_EN	Watermark Enable for FIFO_2_P If enabled, FIFO_2_P will assert WM_2_P_ST when the entries in FIFO_2_P are greater than the programmed value in WATERMARK_2_P. 0b - Disables 1b - Enables
3 WM_1_N_EN	Watermark Enable for FIFO_1_N If enabled, FIFO_1_N will assert WM_1_N_ST when the entries in FIFO_1_N are greater than the programmed value in WATERMARK_1_N. 0b - Disables 1b - Enables
2 WM_1_P_EN	Watermark Enable for FIFO_1_P If enabled, FIFO_1_P will assert WM_1_P_ST when the entries in FIFO_1_P are greater than the programmed value in WATERMARK_1_P. 0b - Disables 1b - Enables
1 WM_0_N_EN	Watermark Enable for FIFO_0_N If enabled, FIFO_0_N will assert WM_0_N_ST when the entries in FIFO_0_N are greater than the programmed value in WATERMARK_0_N. 0b - Disables 1b - Enables
0 WM_0_P_EN	Watermark Enable for FIFO_0_P If enabled, FIFO_0_P will assert WM_0_P_ST when the entries in FIFO_0_P are greater than the programmed value in WATERMARK_0_P. 0b - Disables 1b - Enables

56.7.1.18 FIFO Watermark Error Status (FIFO_WATERMARK_ERROR_ST)

Offset

Register	Offset
FIFO_WATERMARK_ERROR_ST	44h

Function
Indicates watermark and error status for FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_3_N	ERR_3_P	ERR_2_N	ERR_2_P	ERR_1_N	ERR_1_P	ERR_0_N	ERR_0_P	WM_3_N_...	WM_3_P_...	WM_2_N_...	WM_2_P_...	WM_1_N_...	WM_1_P_...	WM_0_N_...	WM_0_P_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15 ERR_3_N	Interrupt from FIFO_3_N 0b - No interrupt 1b - Interrupt
14 ERR_3_P	Interrupt from FIFO_3_P 0b - No interrupt 1b - Interrupt
13 ERR_2_N	Interrupt from FIFO_2_N 0b - No interrupt 1b - Interrupt
12 ERR_2_P	Interrupt from FIFO_2_P

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No interrupt 1b - Interrupt
11 ERR_1_N	Interrupt from FIFO_1_N 0b - No interrupt 1b - Interrupt
10 ERR_1_P	Interrupt from FIFO_1_P 0b - No interrupt 1b - Interrupt
9 ERR_0_N	Interrupt from FIFO_0_N 0b - No interrupt 1b - Interrupt
8 ERR_0_P	Interrupt from FIFO_0_P 0b - No interrupt 1b - Interrupt
7 WM_3_N_ST	Watermark Status for FIFO_3_N 0b - No watermark error 1b - Watermark error
6 WM_3_P_ST	Watermark Status for FIFO_3_P 0b - No watermark error 1b - Watermark error
5 WM_2_N_ST	Watermark Status for FIFO_2_N 0b - No watermark error 1b - Watermark error
4 WM_2_P_ST	Watermark Status for FIFO_2_P 0b - No watermark error 1b - Watermark error
3 WM_1_N_ST	Watermark Status for FIFO_1_N 0b - No watermark error 1b - Watermark error
2 WM_1_P_ST	Watermark Status for FIFO_1_P. 0b - No watermark error 1b - Watermark error

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 WM_0_N_ST	Watermark Status for FIFO_0_N 0b - No watermark error 1b - Watermark error
0 WM_0_P_ST	Watermark Status for FIFO_0_P 0b - No watermark error 1b - Watermark error

56.7.1.19 FIFO 0 P (FIFO_0_P)

Offset

Register	Offset
FIFO_0_P	48h

Function

Contains output data for FIFO_0_P. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_0_P							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_0_P															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_0_P	FIFO_0_P Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_0_P.

56.7.1.20 FIFO 0 N (FIFO_0_N)

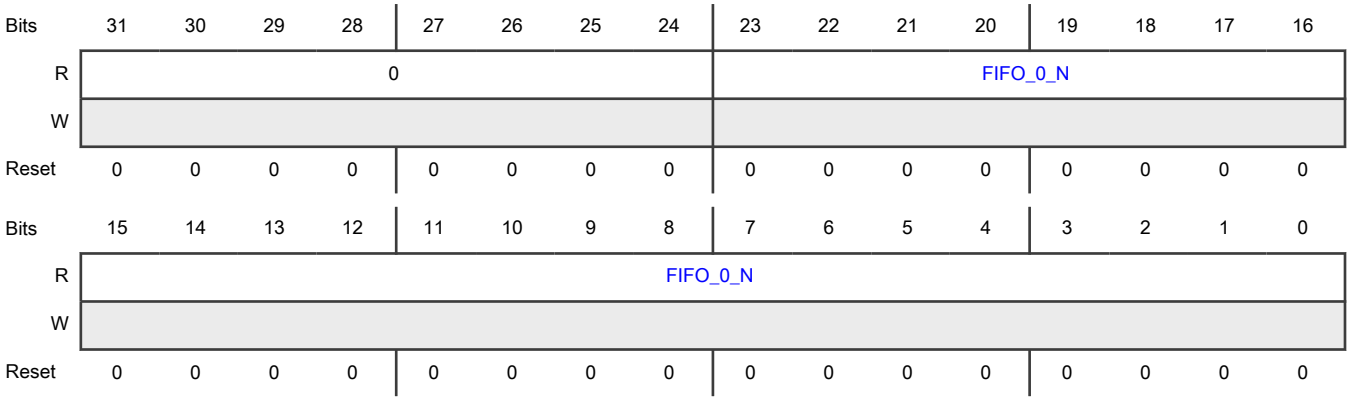
Offset

Register	Offset
FIFO_0_N	4Ch

Function

Contains output data for FIFO_0_N. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_0_N	FIFO_0_N Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_0_N.

56.7.1.21 FIFO 1 P (FIFO_1_P)

Offset

Register	Offset
FIFO_1_P	50h

Function

Contains output data for FIFO_1_P. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_1_P							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_1_P															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_1_P	FIFO_1_P Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_1_P.

56.7.1.22 FIFO 1 N (FIFO_1_N)

Offset

Register	Offset
FIFO_1_N	54h

Function

Contains output data for FIFO_1_N. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_1_N							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_1_N															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_1_N	FIFO_1_N Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_1_N.

56.7.1.23 FIFO 2 P (FIFO_2_P)

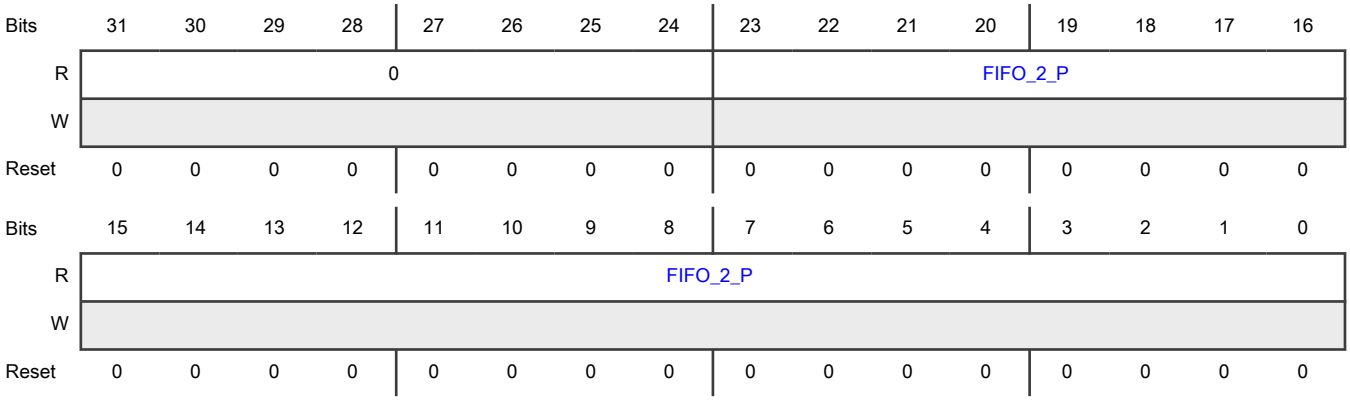
Offset

Register	Offset
FIFO_2_P	58h

Function

Contains output data for FIFO_2_P. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_2_P	FIFO_2_P Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_2_P.

56.7.1.24 FIFO 2 N (FIFO_2_N)

Offset

Register	Offset
FIFO_2_N	5Ch

Function

Contains output data for FIFO_2_N. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_2_N							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_2_N															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_2_N	FIFO_2_N Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_2_N.

56.7.1.25 FIFO 3 P (FIFO_3_P)

Offset

Register	Offset
FIFO_3_P	60h

Function

Contains output data for FIFO_3_P. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_3_P							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_3_P															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_3_P	FIFO_3_P Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_3_P.

56.7.1.26 FIFO 3 N (FIFO_3_N)

Offset

Register	Offset
FIFO_3_N	64h

Function

Contains output data for FIFO_3_N. For SE mode, there're up to eight of the FIFOs working simultaneously. For Diff mode, there're four.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FIFO_3_N							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_3_N															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0 FIFO_3_N	FIFO_3_N Output Data Indicates 24-bit output data in the format of signed 2’s complement for FIFO_3_N.

56.7.1.27 FIFO Entries Availability 0 (FIFO_ENTRIES_AVAIL_0)

Offset

Register	Offset
FIFO_ENTRIES_AVAIL_0	6Ch

Function
Indicates availability for FIFO_0-1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FIFO_1_N_AVAIL				0				FIFO_1_P_AVAIL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO_0_N_AVAIL				0				FIFO_0_P_AVAIL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28-24 FIFO_1_N_AVAIL	Entries Available for FIFO_1_N

Table continues on the next page...

Table continued from the previous page...

Field	Function
23-21 —	Reserved
20-16 FIFO_1_P_AVAIL	Entries Available for FIFO_1_P
15-13 —	Reserved
12-8 FIFO_0_N_AVAIL	Entries Available for FIFO_0_N
7-5 —	Reserved
4-0 FIFO_0_P_AVAIL	Entries Available for FIFO_0_P

56.7.1.28 FIFO Entries Availability 1 (FIFO_ENTRIES_AVAIL_1)

Offset

Register	Offset
FIFO_ENTRIES_AVAIL_1	70h

Function
Indicates availability for FIFO_2-3.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FIFO_3_N_AVAIL				0				FIFO_3_P_AVAIL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO_2_N_AVAIL				0				FIFO_2_P_AVAIL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28-24 FIFO_3_N_AVAIL	Entries Available for FIFO_3_N
23-21 —	Reserved
20-16 FIFO_3_P_AVAIL	Entries Available for FIFO_3_P
15-13 —	Reserved
12-8 FIFO_2_N_AVAIL	Entries Available for FIFO_2_N
7-5 —	Reserved
4-0 FIFO_2_P_AVAIL	Entries Available for FIFO_2_P

56.7.1.29 FIFO Interrupt (FIFO_INTERRUPT)

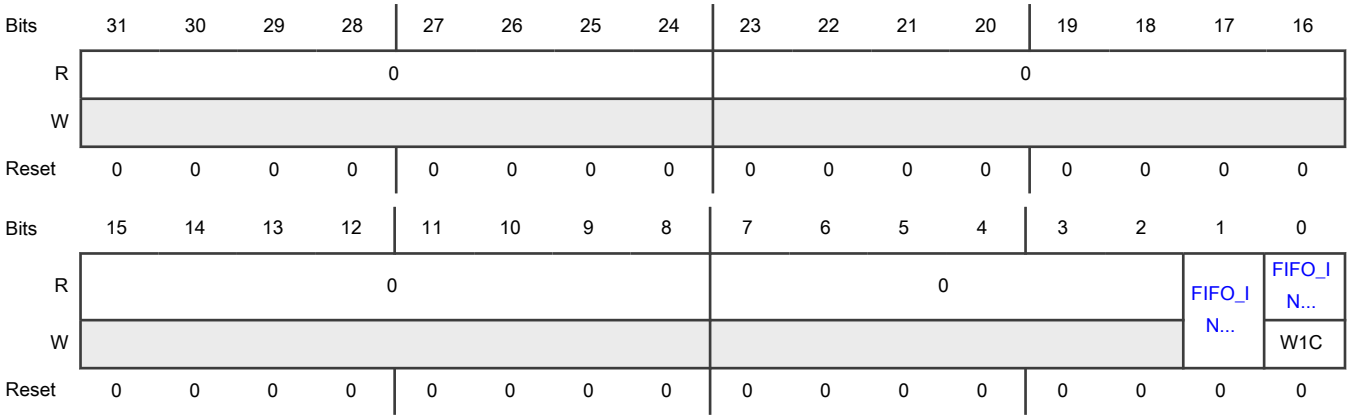
Offset

Register	Offset
FIFO_INTERRUPT	74h

Function

Controls interrupts for FIFO.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-2 —	Reserved
1 FIFO_INTERRUPT_EN	FIFO Interrupt Enable 0b - Disables 1b - Enables
0 FIFO_INTERRUPT	FIFO Interrupt This signal is the result of ORing ERR_0_P, ERR_0_N, ERR_1_P, ERR_1_N, ERR_2_P, ERR_2_N, ERR_3_P, and ERR_3_N.

Chapter 57
PDM Microphone Interface (MICFIL)

57.1 Chip-specific MICFIL information

Table 563. Reference links to related information

Topic	Related module	Reference
Full description	MICFIL	MICFIL
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

57.1.1 Module instances

This device has one instance of the MICFIL module.

Table 564. MICFIL instance

	MICFIL
Located in	VDD1_SENSE
Access	CPU0, CPU1, HiFi4, HiFi1, eDMA0, eDMA1, eDMA2, eDMA3

57.1.2 Reset

This module supports the following resets:

- Cold reset resets the whole MICFIL module.
- RSTCTL3.PRSTCTL1[MICFIL] resets DATACHn registers
- Software resets by MICFIL register fields, see CTRL_1[SRES] and VAD0_CTRL_1[VADRST] for more details.

57.2 Overview

MICFIL delivers audio from microphones to the processor in several applications, such as mobile telephones. As current digital audio systems use a multi-bit audio signal (also known as multi-bit PCM) to represent the signal, this module implements the required digital interface (a series of filters) to transform a pulse density modulated (PDM) microphone bitstream into a 24-bit PCM signal in the audio band, at a configurable output sampling rate.

The implementation of this digital interface is based on the application of digital signal processing techniques used in hardware. The MICFIL architecture is designed to save gate count and minimize power consumption.

You can implement MICFIL to work in a multi-channel mode. All channels have the same configuration, but you could independently turn on or turn off each input channel.

57.2.1 Block diagram

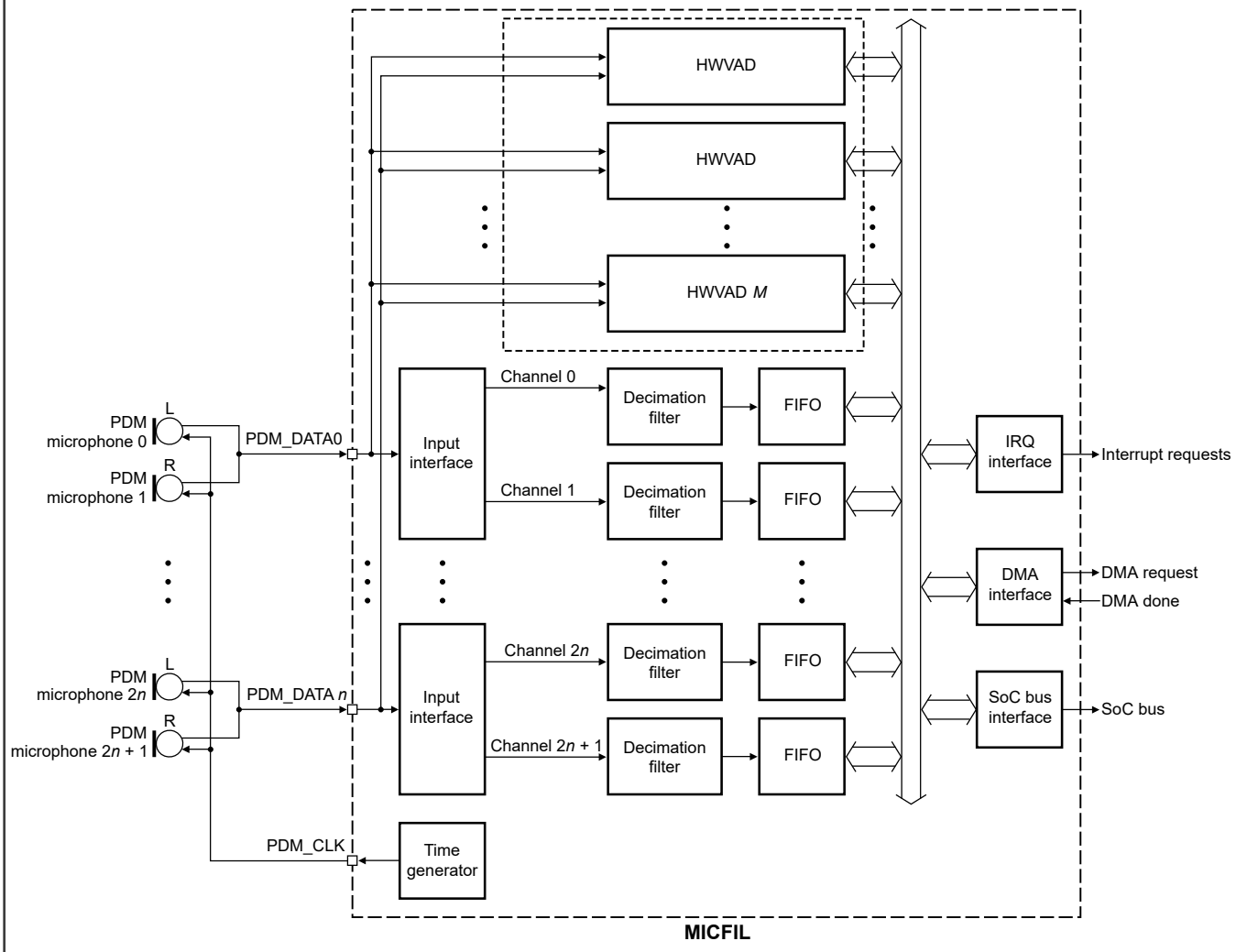


Figure 242. MICFIL block diagram

- HWVAD = hardware voice activity detector

57.2.2 Features

- Decimation filters:
  - Fixed-point filtering
  - 24-bit PCM audio output
  - Internal clock divider for a programmable PDM clock generation: clock divider bypass capability for low-frequency operations
  - Frame synchronization
  - Full or partial set of channel operations with individual enable controls
  - Programmable decimation rate
  - Programmable DC remover at output
  - Range adjustment capability

- FIFOs with interrupt and DMA capability: each FIFO having a length of 8 entries
- Hardware voice activity detector (HWVAD), equipped with:
  - Interrupt capability
  - Zero-crossing detection (ZCD) option

57.3 Functional description

57.3.1 Time generator

The time generator unit:

- Generates the PDM_CLK to microphones. This clock is the same and is active for all the PDM microphones, which means, it is not possible to turn off the PDM_CLK only for one single microphone.
- Delivers the PDM_CLK to all microphones that must operate at the same clock frequency. Each input interface receives a time multiplexed PDM bitstream from two PDM microphones and it separates audio information in two channels: left (0) and right (1). Every decimation filter, corresponding to its channel, does this processing.

There exists an internal clock divider (see CTRL_2[CLKDIV]) within the time generator that is enabled when CTRL_2[CLKDIVDIS] = 0; otherwise, the application clock directly drives the microphones.

NOTE

A soft reset affects the time generator unit (see CTRL_1[SRES]). The PDM_CLK starts with a rising edge after a soft reset is issued.

57.3.1.1 Clock divider

The sampling rate of the microphone input depends on the PDM_CLK rate, which you can trim using an internal clock divider, configuring the quality mode (see Quality modes) or trimming the MICFIL's root clock frequency (MICFIL_CLK_ROOT rate).

In case the clock divider is enabled, CTRL_2[CLKDIV] defines the internal clock divider frequency. You must configure CTRL_2[CLKDIV] by using Equation 7.

$$CLKDIV = \frac{MICFIL_CLK_ROOT\ rate}{8 \times OSR \times (output\ rate)}$$

Equation 7. CLKDIV value calculation

After calculating the value of CTRL_2[CLKDIV], you can estimate the generated PDM_CLK rate based on Equation 8, where the "K" factor value depends on the quality mode shown in Table 565 and floor(x) is the function that takes a real number x as input and gives the greatest integer less than or equal to x as output.

$$PDM_CLK\ rate = \frac{MICFIL_CLK_ROOT}{2 \times floor(K \times CLKDIV)}$$

Equation 8. PDM_CLK rate calculation

Table 565. K factor value

Quality mode	K factor
High Quality	1/2
Medium Quality, Very-Low Quality 0	1
Low Quality, Very-Low Quality 1	2
Very-Low Quality 2	4

57.3.1.2 I/O timing requirements

The PDM microphones must meet the setup and hold timing requirements shown in Table 566 and Figure 243. The "K" factor value in Table 566 depends on the quality mode you select based on Table 565.

NOTE

These timing requirements apply only if the clock divider functionality is enabled (see CTRL_2[CLKDIV]) with (CTRL_2[CLKDIVDIS] = 0); otherwise, there are no special timing requirements.

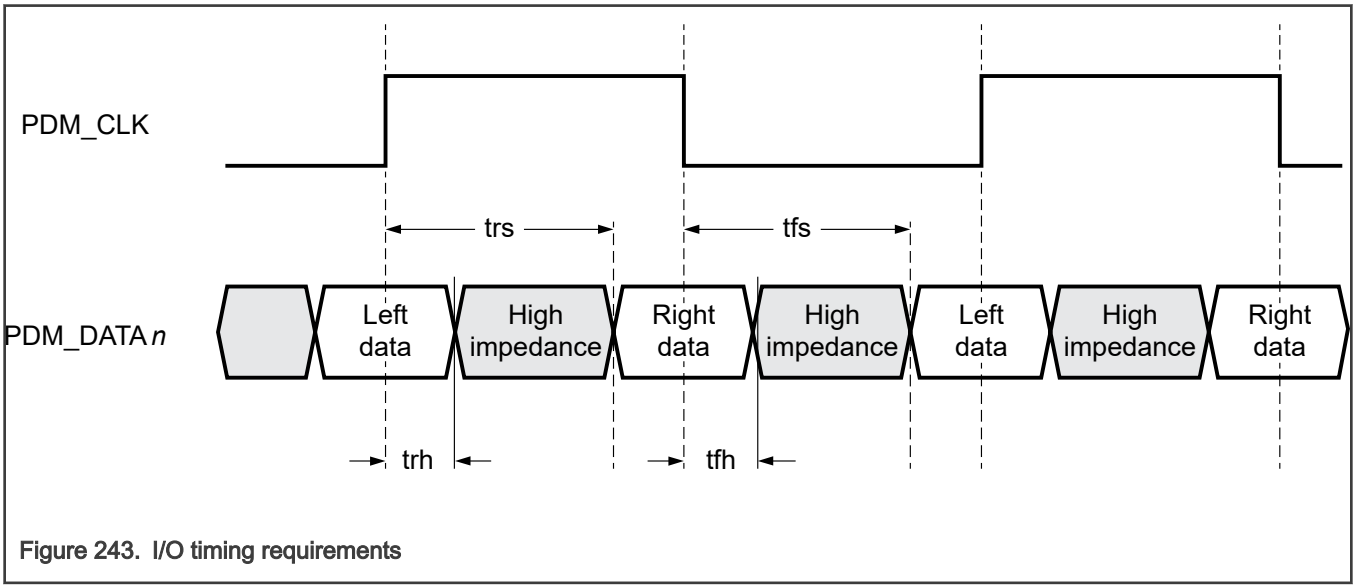


Figure 243. I/O timing requirements

Table 566. Timing parameters

Parameter	Value
trs, tfs	$\leq \frac{floor(K \times CLKDIV) - 1}{MICFIL_CLK_ROOT\ rate}$ Depending on the value of "K," you must ensure that "floor(K x CTRL_2[CLKDIV])" > 1 to avoid timing problems.
trh, tfh	≥ 0

57.3.2 Input interface

Figure 244 shows the timing diagram of the input interface signals when the clock divider is enabled. The bitstream incoming from the microphone data input "n" (PDM_DATA*n*) in the first half (right microphone) of the PDM_CLK is directed to channel "2n+1" and the data generated during the last half (left microphone) is directed to channel "2n".

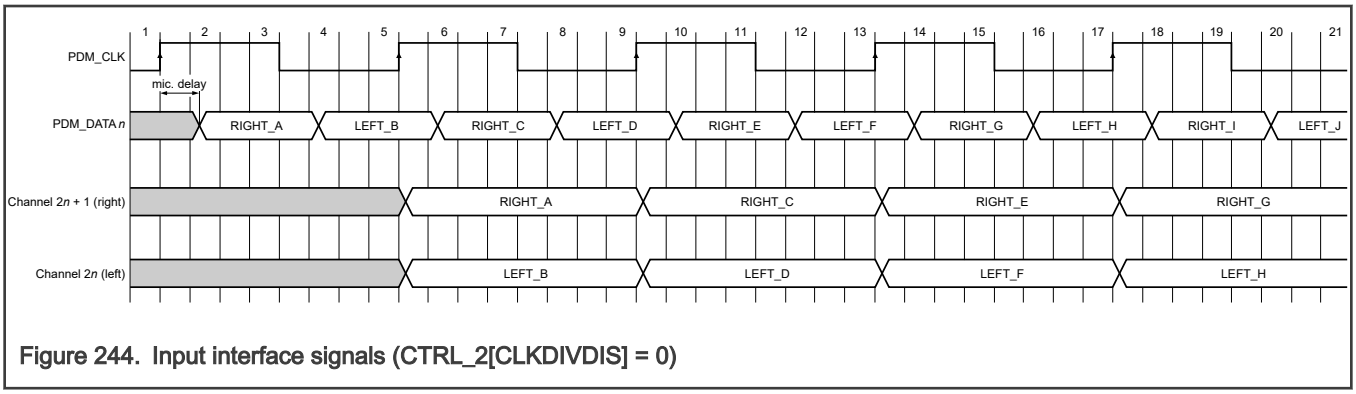


Figure 244. Input interface signals (CTRL_2[CLKDIVDIS] = 0)

57.3.3 Decimation filter

A decimation filter facilitates PDM to PCM conversion.

Figure 245 shows the block diagram of the decimation filter. The block consists of:

- A cascade integrator comb (CIC) filter, which is a low-pass filter that you commonly use to filter sigma-delta modulator output signals. The filter is implemented in the audio band (20 Hz–20.0 kHz @48 kHz output sampling rate by default) with a configurable decimation rate. You can implement it using a series of CIC, compensation, and DC remover filters.
- A DC remover, which is a high-pass filter that you use to remove the DC component of the processed signal with a configurable cut-off frequency. Two DC removers exist: one in the input of the decimation filter, embedded in the CIC filter, and another in the last stage of the decimation filter.
- Two compensation filters for each channel, which implement a low-pass digital filter with decimation by 2. You can use them to compensate the high CIC drop in passband. These filters help the FIR filter block to flatten the overall passband response.

You activate the "Sampler" block depending on the quality mode selected (using CTRL_2[QSEL]). In Low-Quality and Very-Low-Quality 2 modes, the sampling rate is doubled.

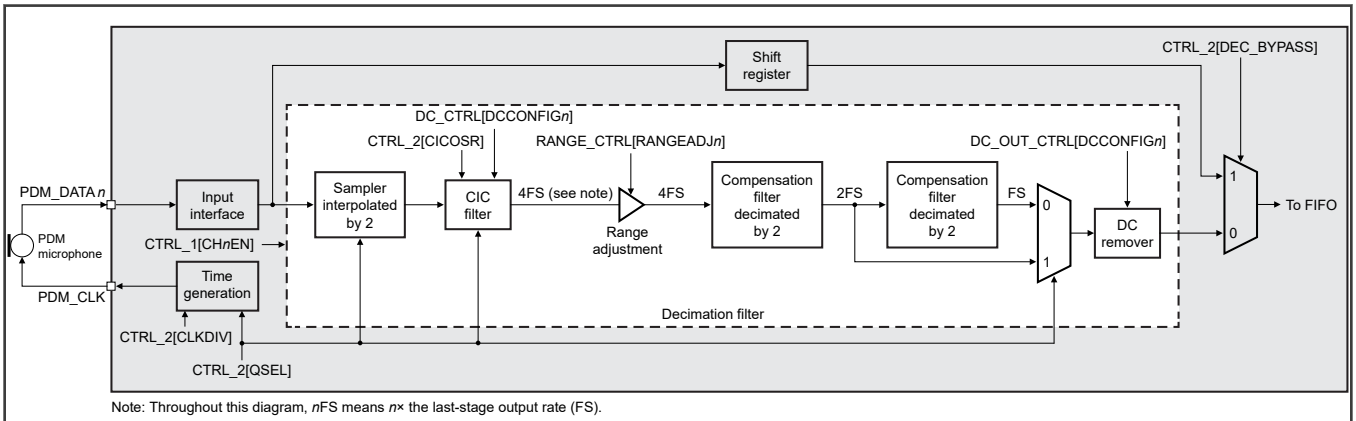
The decimation filter output depends on the quality mode that you select by using CTRL_2[QSEL], based on different compensation filter output. You can supply the DC remover with the 2 FS signal, instead of the 1 FS signal, to achieve lower-power consumption, bypassing the second compensation decimation filter. The second compensation filter is bypassed when CTRL_2[QSEL] = 110b, 101b, or 100b (in Very-Low Quality modes).

You can adjust the range of each decimation filter by using a range adjustment block so that you can raise an error interrupt in case an adjustment overflow or underflow is detected.

The output of a decimation filter is stored into a FIFO buffer, and each FIFO is mapped to MICFIL Output Result (DATACH0 - DATACH7). It is possible to generate either an interrupt or a DMA request when, in each FIFO of all enabled channels, the number of data stored surpasses a configured watermark.

Also, independently on decimation filters, there is a hardware voice activity detector (HWVAD) which implements voice-detection algorithms to generate wake-up interrupts. See Hardware voice activity detector (HWVAD).

In order to process PDM data with higher quality, you can bypass the decimation filter by configuring CTRL_2[DEC_BYPASS], which enables the writing of PDM data into FIFO in the form of 32 bits. Software is used to process PDM data in this situation.



Note: Throughout this diagram, *n*FS means *n*× the last-stage output rate (FS).

Figure 245. Decimation filter block diagram

57.3.3.1 CIC filter

The CIC filter is an optimized implementation of the low-pass SINC filter, with multiple stages. It is widely used because it requires just adders and subtracters. The CIC filter comprises an integrator and a comb as its sub-blocks. As shown in the following figure, CTRL_2[CICOSR] and the quality mode you select (see Quality modes) define the CIC decimation rate.

$$OSR = 32 - CICOSR$$

$$CIC\ decimation\ rate = \begin{cases} 2 \times OSR\ ;\ If\ HQ,\ VLQ0 \\ OSR\ \ \ \ ;\ others \end{cases}$$

Equation 9. CIC decimation rate

57.3.3.2 DC remover

Two DC removers exist in the filter chain, one in the input and another in the output. The former is integrated in the CIC filter, and the latter is placed just before the FIFO. Both can be bypassed independently.

The DC remover eliminates the DC level component of the signal using a high-pass filter with a very low cut-off frequency that you select using DC_CTRL[DCCONFIG*n*]. This filter is bypassed when the value of this field is 11b.

NOTE

See PARAM[DC_OUT_BYPASS] and PARAM[DC_BYPASS] to know whether one of the DC removers is permanently disabled in the chip.

57.3.3.3 Compensation filters

A compensation filter divides the baseband signal into two equal subbands before it could to be decimated by 2. It also compensates the CIC filter drop in passband and helps to flatten the overall passband response. MICFIL includes two compensation filter stages, with the possibility of bypassing the last one.

57.3.3.4 Filter performance

57.3.3.4.1 Overall filter gain

As shown in Equation 10, the overall filter gain depends on:

- The quality mode you select (see Quality modes).
- The CIC decimation rate (see Equation 9).
- Dynamic range adjustment of the CIC filter (see RANGE_CTRL[RANGEADJ*n*]).

NOTE

To get a better audio performance, you must configure and adjust RANGE_CTRL[RANGEADJ*n*] depending on the quality mode selected, as pointed out in the register description.

$$Overall\ filter\ gain\ (dB) = \begin{cases} 100 \times log_{10}(64 - 2 \times CICOSR) + 6.02 \times RANGE_CTRL[RANGEADJn] - 180.60\ ;\ if\ QSEL \in [HQ,VLQ0] \\ 100 \times log_{10}(32 - CICOSR) + 6.02 \times RANGE_CTRL[RANGEADJn] - 180.60\ \ \ \ ;\ others \end{cases}$$

Equation 10. Overall filter gain

Finally, the input and output dBFS levels relate to each other, as shown in Equation 11.

$$Output\ (dBFS) = input\ (dBFS) + overall\ filter\ gain\ (dB)$$

Equation 11. Filter input and output relationship

57.3.3.4.2 Overall passband

The decimation filter passband depends on the quality mode and output rate (FS). For instance, the overall filter passband for different output data rates when MICFIL is in High (HQ), Medium (MQ), and Low-Quality (LQ) modes is shown in Table 567.

Table 567. Filter passband in HQ, MQ, and LQ modes (DC_CTRL[DCCONFIG*n*] = 0)

Output data rate	Overall filter passband
FS	**0.0004FS - 0.4167FS**
16 kHz	6.67 Hz - 6.67 kHz
48 kHz	20 Hz - 20 kHz

The overall filter passband for different output data rates when MICFIL is in Very-Low Quality (VLQ) modes is shown in Table 568.

Table 568. Filter passband in VLQn modes (DC_CTRL[DCCONFIG*n*] = 0)

Output data rate	Overall filter passband
FS	**0.0002FS - 0.2083FS**
16 kHz	3.33 Hz - 3.33 kHz
48 kHz	10.00 Hz - 10.00 kHz
96 kHz	20 Hz - 20 kHz

57.3.3.5 Frame synchronization

If CTRL_1[FSYNCEN] = 1, MICFIL sends data to FIFOs in frame size defined by FSYNC[FSYNCLEN]. In this way, if a channel is enabled while MICFIL is running (CTRL_1[PDMIEN] = 1), MICFIL starts sending data to its respective FIFO after the current frame is sent completely. If the channel is disabled:

- if at least one other channel is still enabled, the corresponding channel stops sending data after the current frame is sent completely.
- if after that no channel is enabled, MICFIL stops sending data immediately, even if the act of sending frames is incomplete.

Below example shows the above behavior when two channels, DATACH0 and DATACH2 are enabled, and then disabled one after the other.

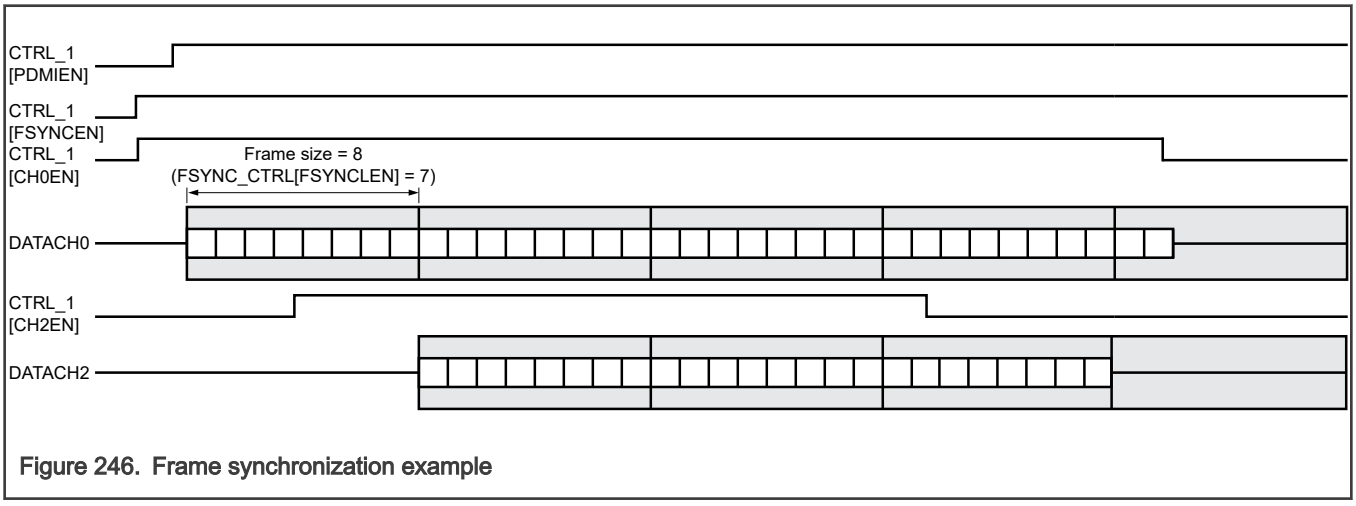


Figure 246. Frame synchronization example

57.3.3.6 Quality modes

If the decimation filter is running independently on the selected power mode, MICFIL can operate in multiple different quality modes that you can select by using CTRL_2[QSEL]. The decimation and interpolation rates in internal blocks are shown in Table 569, where the oversampling rate (OSR) depends on the value of CTRL_2[CICOSR], as shown in the following equation.

$$OSR = 32 - CICOSR$$

Equation 12. OSR

You can configure the selected quality dynamically when the decimation filter is running; the output rate is maintained in every quality selected.

The passband of the decimation filter depends on the quality mode (see Overall passband).

To maintain the same output rate in the decimation filter when the quality is changed, the PDM_CLK rate changes depending on the quality selected. For more on PDM_CLK, see Time generator.

You can enable or disable the sampler and compensation filters according to the selected quality. When they are disabled, you can achieve a low-power consumption. The selected quality also determines whether the CIC filter decimation rate is doubled. For more, see Decimation filter.

With an increase of the clock rate and OSR, you can expect a better signal-to-noise ratio (SNR) performance, which in turn leads to a higher audio quality.

Table 569. Quality modes

Quality mode	CTRL_2[QSEL]	Sampler interpolation	CIC filter decimation	First-compensation filter decimation	Second-compensation filter decimation	PDM_CLK rate	Passband
High Quality	001	—	:(2OSR)	:2	:2	Output rate × 8 × OSR	To ~0.5 × output rate
Medium Quality	000	—	:OSR	:2	:2	Output rate × 4 × OSR	To ~0.5 × output rate
Low Quality	111	× 2	:OSR	:2	:2	Output rate × 2 × OSR	To ~0.5 × output rate
Very-Low Quality 0	110	—	:(2OSR)	:2	—	Output rate × 4 × OSR	To ~0.25 × output rate
Very-Low Quality 1	101	—	:OSR	:2	—	Output rate × 2 × OSR	To ~0.25 × output rate
Very-Low Quality 2	100	× 2	:OSR	:2	—	Output rate × OSR	To ~0.25 × output rate

57.3.3.7 Output rate

The output rate is the sampling rate at the decimation filter output. It depends on the sampling rate of the microphone input and the configured quality mode (see Quality modes). Common audio sampling rates are 48 kHz and 16 kHz. In this chapter, the output rate is sometimes referred to as FS.

57.3.4 Hardware voice activity detector (HWVAD)

HWVAD is a block which detects voice activity in a channel selected by the user as shown in Figure 247 and Figure 248. HWVAD takes data from the input of a selected PDM microphone to detect if there is any voice activity. If HWVAD detects a voice activity, it can issue an interrupt to the system. You can configure HWVAD in Envelope-based or Energy-based mode as described in Initialization.

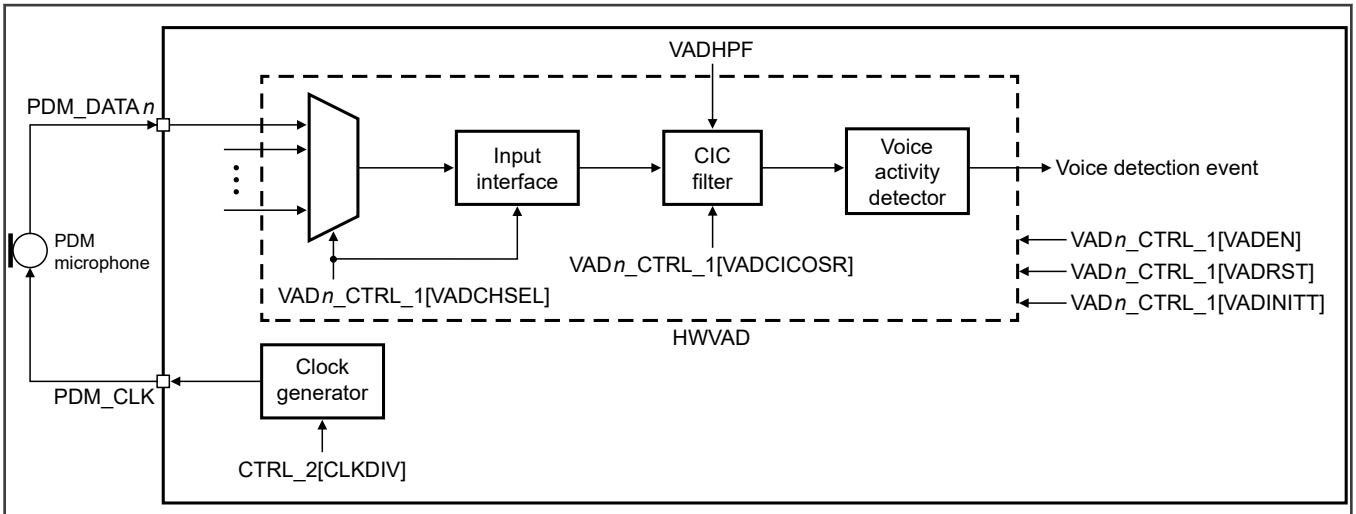


Figure 247. HWVAD block diagram

HWVAD can divide the incoming PCM signal from its internal CIC in frames. The frame energy is used to estimate the background noise. HWVAD examines both frame energy and noise energy. When HWVAD identifies a specific delta between them, it detects a voice activity and then issues an interrupt. Optionally, you can enable a zero-crossing detector (ZCD) to avoid missing of low-energy voiced speech in order to improve the voice detection performance.

You can enable HWVAD by configuring VAD*n*_CTRL_1[VADEN] and reset HWVAD by software through VAD*n*_CTRL_1[VADRST].

NOTE

You can enable or reset HWVAD only when MICFIL isn't running (STAT[BSY_FIL] = 0).

To perform voice detection, HWVAD uses a noise filter and a signal filter as shown in Figure 248. In addition, you can use a minimum block in the output of the noise filter and a maximum block in the output of the signal filter. HWVAD can scale the result of both blocks with gain blocks. Finally, HWVAD compares the estimated signal and noise, and it detects voice when signal is greater than noise .

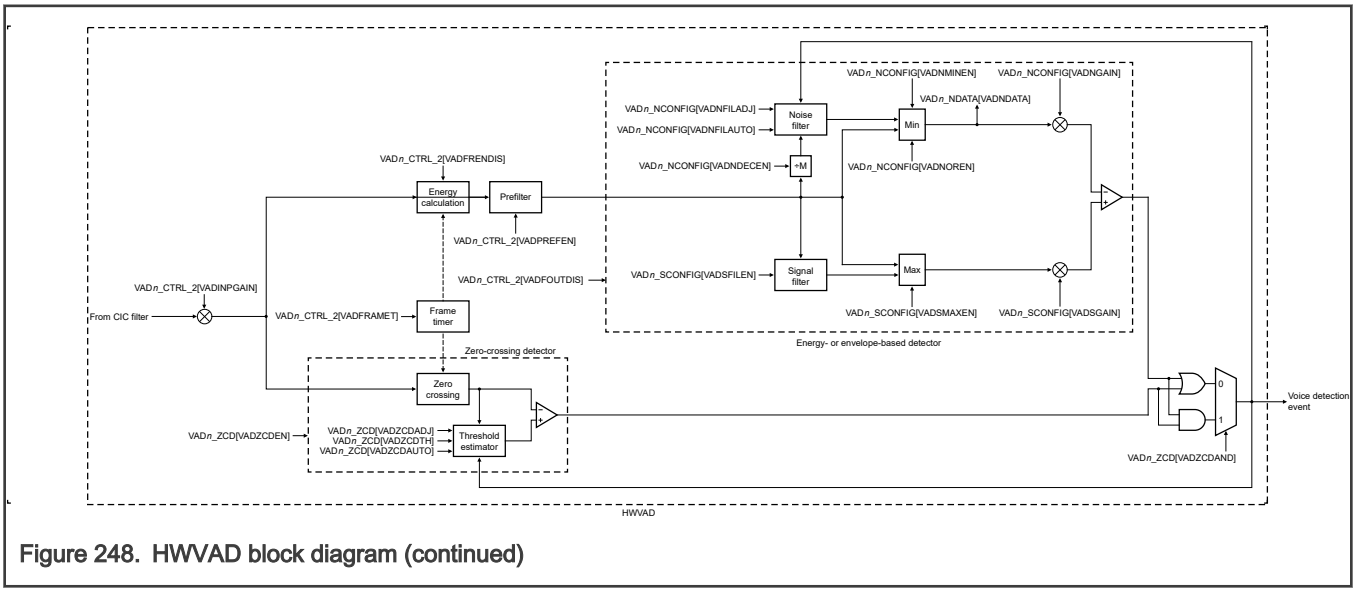


Figure 248. HWVAD block diagram (continued)

57.3.4.1 Input setting

HWVAD input has internally a user-configurable high-pass filter (see VAD*n*_CTRL_2[VADHPF] field description) and a user-configurable gain (see VAD*n*_CTRL_2[VADINPGAIN]). You can use both controls to configure the module based on external noise conditions.

57.3.4.2 Frame time setting

To perform voice detection the signal frames incoming from the CIC filter can be divided in frames whose duration can be configured. The frame time depends on the CIC output rate, the value of VAD*n*_CTRL_2[VADFRAMET] and a constant as defined in Equation 13.

$$frame\ time = \frac{VADFRAMET \times 192}{CIC\ output\ rate}$$

Equation 13. Frame time

For instance, in the typical case of the CIC filter the output rate is 192 kHz, the frame time is 10 ms when [VADFRAMET] is 10.

57.3.4.3 Initialization time

You can configure an initialization by configuring VAD*n*_CTRL_1[VADINITT]. During this initialization time, HWVAD doesn't issue any voice detection event and all detectors are disabled.

$$initialization\ time = (VADINITT + 1) \times frame\ time$$

Equation 14. Initialization time

For instance, in case that frame time is 10 ms and [VADINITT] = 10, the initialization time is 110 ms.

57.3.4.4 Energy calculation

In case VAD*n*_CTRL_2[VADFRENDIS] = 0, HWVAD uses the sum of the absolute values of all the samples of each frame to detect voice. Otherwise, if [VADFRENDIS] = 1, it uses the absolute value of the samples to detect voice, which means that the data incoming from CIC is not divided in frames.

57.3.4.5 Pre-filter

The pre-filter block is a low-pass filter which can be enabled when VAD*n*_CTRL_2[VADPREFEN] = 1.

57.3.4.6 Energy and envelope-based detector

To perform voice detection, HWVAD uses a noise filter and a signal filter as shown in Figure 248. In addition, it can use a minimum block in the output of the noise filter and a maximum block in the output of the signal filter. HWVAD can scale the result of both blocks with gain blocks. Finally, HWVAD compares the estimated signal and noise, and it detects voice when the signal is greater than the noise.

57.3.4.6.1 Noise filter

The noise filter is a low-pass filter . You can configure VAD*n*_NCONFIG[VADNFILADJ] to adjust the filter . In case VAD*n*_NCONFIG[VADNFILAUTO] = 1 the noise filter is enabled only when voice is not detected. When voice is detected the filter output does not change. In case [VADNFILAUTO] = 0 the filter is always enabled.

A decimation stage in the input of the noise filter can be enabled by VAD*n*_NCONFIG[VADNDECEN] field.

57.3.4.6.2 Minimum noise

To estimate the noise, HWVAD uses a block which calculates the minimum value between the input and output of the noise filter. This block can be enabled by VAD*n*_NCONFIG[VADNMINEN], otherwise the output of the noise filter passes directly.

The estimated noise of the output of this block is stored in VAD*n*_NDATA[VADNDATA] for post-processing via software.

57.3.4.6.3 Signal filter

The signal filter is a low-pass filter . This filter is enabled when VAD*n*_SCONFIG[VADSFILEN] = 1, otherwise the filter is bypassed.

57.3.4.6.4 Maximum signal

To estimate the signal, HWVAD uses a block which calculates the maximum value between the input and output of the signal filter. This block can be enabled by VAD*n*_SCONFIG[VADSMAXEN].

57.3.4.6.5 Filter result gain setting

There are two gain adjustment controls, one for the estimated noise (VAD*n*_NCONFIG[VADNGAIN]) and another for estimated signal (VAD*n*_SCONFIG[VADSGAIN]). Both gain adjustments outputs are compared, the comparison result determines whether or not there is voice in the channel.

57.3.4.7 Zero-crossing detector (ZCD)

Optionally, the use of a ZCD circuit can increase the HWVAD performance. It is enabled by VAD*n*_ZCD[VADZCDEN].

The threshold of the ZCD could be static or calculated dynamically depending on VAD*n*_ZCD[VADZCDAUTO]. When it is dynamically determined ([VADZCDAUTO] = 1) the threshold is calculated automatically taking VAD*n*_ZCD[VADZCDTH] as the initial value, otherwise ([VADZCDAUTO] = 0) the threshold is static in [VADZCDTH] user-configurable value.

You can use AND or OR to combine the ZCD result with the energy-based or envelope-based detection, it is selected by VAD*n*_ZCD[VADZCDAND]. The filter coefficient is equal to ([VADZCDADJ] + 1) × 2^{-4}.

57.3.4.8 Interrupt

HWVAD issues an interrupt when voice activity is detected and VAD*n*_CTRL_1[VADIE] is enabled. The voice activity event also sets the VADIF flag in VAD*n*_STAT[VADIF], this flag is cleared when you write 1 to it.

57.3.4.9 Extended operation

You can use VAD*n*_STAT[VADIF] for an HWVAD-extended operation, that is, using VAD*n*_NDATA[VADNDATA] value to determine a more accurate voice detection event by software. Below is the procedure to follow:

1. HWVAD issues an interrupt, [VADIF] is set.
2. Leave [VADIF] asserted while software is processing [VADNDATA] content.
3. When software finishes to process [VADNDATA] and detects voice activity:
   - If Envelope-based detection is used, pulse VAD*n*_CTRL_1[VADST10] by more than 2 cycles and clear the VADIF flag.
   - If Energy-based detection is used, clear the VADIF flag.

57.3.4.10 VAD soft-reset

You can request the VAD soft-reset by writing 1 to VAD*n*_CTRL_1[VADRST]. It resets the following blocks and registers:

- VAD*n*_STAT[VADIF] and VAD*n*_STAT[VADINSATF].
- All internal VAD buffers.

57.3.5 Bus interface

The bus interface manages DMA transactions, interrupt requests, and user register interface with the chip.

MICFIL can deliver an interrupt request or DMA request. Then, the chip or DMA, respectively, could access the filter results stored in FIFOs via the internal bus interface.

For detailed information on DMA transactions and interrupt requests, see DMA and Interrupts.

57.3.5.1 FIFO

This is the first-in, first-out (FIFO) buffer that saves the overall filtering results. The length of each FIFO is 8 entries, and there exists one FIFO for each channel. A push writes data into the FIFO, and a read pops data from FIFO. The push and pop operations from FIFO are in different clock domains. The push operation uses the MICFIL clock while the pop operation uses the CPU clock. The FIFO has a configurable watermark (see FIFO_CTRL[FIFOWMK]). When the amount of data in all FIFOs of the enabled channels is greater than the watermark value (FIFO_CTRL[FIFOWMK]), a DMA or IRQ is requested, depending on the value of CTRL_1[DISEL]. After CTRL_1[PDMIEN] enables the filters, the initial FIR results are discarded and not written into the FIFOs. Approximately 68 initial FIR results are not stored in FIFOs. After the filter generates valid data, the results are pushed into the FIFO.

A new filtered result is discarded (not stored) when the FIFO is full and CTRL_1[FIFOOVF*n*] becomes 1. Similarly, when trying a pop operation, if the FIFO is empty, no valid data is read and CTRL_1[FIFOUND*n*] becomes 1. In both cases, when overflow or underflow occurs, they are flagged in MICFIL FIFO Status (FIFO_STAT) to the respective channel, and an error interrupt is generated if CTRL_1[ERREN] is 1.

When you reset the FIFO by writing 1 to CTRL_1[SRES], zero values fill all FIFO positions, as well as push and pop pointers are set to their initial positions.

57.3.6 Modes of operation

This section describes the MICFIL operation modes that depend on:

- Whether MICFIL receives a Stop request.
- The chip's Debug mode.
- The configuration of CTRL_1[MDIS], CTRL_1[DBGE], CTRL_1[DBG],and CTRL_1[DOZEN], as summarized in the following table.

Table 570. Mode selection

Mode	Normal	Debug		Low-Power		Disable/Low-Leakage (DLL)		
CTRL_1[MDIS]	0	0		0		0		1
Stop request	0	0		1		1		Any
CTRL_1[DOZEN]	Any	Any		0		1		Any
CTRL_1[DECFILS]	Any	Any		1	0	Any		
Debug request or CTRL_1[DBG]	0	1		Any		Any		
CTRL_1[DBGE]	Any	1	0	Any		Any		
CPU clock	Enabled	Enabled		Gated (Externally)		Gated (Externally)	Enabled	Gated (Internally)
MICFIL internal clock	Enabled	Enabled		Enabled		Gated		
Decimation filter state	Run[1]	Run[1]	Stop	Run[1]	Stop	Stop		
HWVAD*n* state	Run[2]	Run[2]	Stop	Run[2]		Stop		

1. Depends on the configuration of CTRL_1[PDMIEN] and CTRL_1[CH*n*EN].
2. Depends on the configuration of CTRL_1[PDMIEN] and CTRL_1[VADEN].

57.3.6.1 Normal mode

This is the default operational mode for MICFIL. In this mode, you can enable the channel functionality to perform the filtering operation.

57.3.6.2 Debug mode

You could use this mode to check the status of the module, when requested. In this mode, you can stop MICFIL after the remaining data processing is complete, depending on the value of CTRL_1[DBGE]. You can activate this mode by writing 1 to CTRL_1[DBG].

If CTRL_1[DBGE] = 0, and CTRL_1[DBG] = 1 when MICFIL is running, MICFIL stops after processing the remaining data. Otherwise, if [DBGE] = 1, MICFIL continues to run.

MICFIL cannot enter Debug mode after entering DLL or Low-Power mode.

The Debug mode affects all channels in MICFIL. Access to output buffers, as well as their related flags, remains operational.

57.3.6.3 Low-Power mode

MICFIL enters Low-Power mode when MICFIL receives a Stop request and CTRL_1[DOZEN] = 0. In Low-Power mode, MICFIL is able to filter the PDM microphone data and perform voice detection such as in Normal mode, and can generate a wake-up event through asynchronous DM or IRQ requests, depending on the values of CTRL_1[CH*n*EN], VAD*n*_CTRL_1[VADEN], and CTRL_1[PDMIEN].

After MICFIL enters Low-Power mode:

- If CTRL_1[DECFILS] = 1, Stop request is acknowledged immediately and the decimation filter keeps running.
- If CTRL_1[DECFILS] = 0, the decimation filter processes the last input sample. After that, Stop request is acknowledged and the filter is stopped from running.

During this mode, VAD*n*_CTRL_1[VADEN] continues running if it was previously enabled ([VADEN] = 1) during Normal mode.

You can generate DMA and IRQ requests asynchronously to wake up the CPU.

NOTE

The asynchronous DMA or interrupt requests are generated only in this mode.

57.3.6.4 Disable/Low-Leakage (DLL) mode

In this mode, MICFIL stops after the remaining data processing is complete. MICFIL is in DLL mode when:

- MICFIL receives a Stop request and CTRL_1[DOZEN] = 1.
- CTRL_1[MDIS] = 1.

If you request DLL mode when MICFIL is running, MICFIL stops after completion to process the remaining data.

You can write only to CTRL_1 and CTRL_2, with the exception of writes to CTRL_1[DBG] and CTRL_1[SRES], which are ignored. However, you can still read these fields.

57.3.7 Clocking

Table 571. MICFIL clocks

Clock name	Description
CPU clock (MICFIL_CLK, MICFIL_CLK_S)	Clock that controls the registers
MICFIL internal clock (MICFIL_CLK_APP)	Application clock that operates the filter

Besides the above clock signals, MICFIL generates PDM_CLK for PDM microphones based on MICFIL internal clock frequency and the value in CTRL_2[CLKDIV].

57.3.8 Interrupts

If CTRL_1[DISEL] = 10b, an interrupt request indicates that filtering results are stored in the FIFOs to be read by the CPU. In this case, you can read the data in DATACH*n*[DATA], and STAT[CH*n*F] becomes 1 for the enabled channels when the CHn FIFO surpasses the watermark level.

When MICFIL is in Low-Power mode, the CPU clock is disabled, but the MICFIL clock remains enabled. In this case, an asynchronous interrupt is delivered to the system. This wakes up the CPU clock, which returns to Normal mode. The system could read the FIFO data, after which it must clear STAT[CH*n*F] to get a new interrupt request.

An error interrupt request can also be generated when an exceptional condition happens, such as an overflow or underflow in a channel output, or an overflow or underflow of a FIFO buffer.

CTRL_1[ERREN] enables error interruption and the respective status flag signalizes the type of error:

- OUTOVFn and OUTUNFn for channel outputs errors
- FIFO_STAT[FIFOOVF*n*] and FIFO_STAT[FIFOUND*n*] for FIFO errors

57.3.9 DMA

If CTRL_1[DISEL] = 1, the FIFO stored samples are read through DMA transactions. Then if there is at least one enabled channel and all FIFOs of all enabled channels reach their watermark levels, the output interface makes a request for a DMA transaction.

If CTRL_1[PDMIEN] = 1 and the filters start to operate, you must wait until the filter results stabilize and no filter results are stored in the FIFOs. Therefore, the first DMA request takes longer to occur than the ones that follow.

STAT[CH*n*F] becomes 0 after a DMA transaction is complete.

If MICFIL is in Low-Power mode and the watermark level is surpassed, an asynchronous DMA request is delivered to the system. Then, the system wakes up the CPU clock, returning to Normal mode to read the FIFO data.

57.4 External signals

Table 572. PDM microphone interface external signals

Signal	Description	Direction
PDM_DATA*n*	Data bitstream received from the nth PDM microphone pair throughout the input interface (see Input interface)	I
PDM_CLK	Clock that is delivered to the PDM microphones from the clock generator (see Time generator)	O

NOTE

MICFIL does not provide metastability protection or filtering for input data. You must generate the data using the provided PDM_CLK.

57.5 Initialization

57.5.1 Procedure without HWVAD

Perform this procedure for MICFIL initialization without enabling HWVAD, (that is, by using only decimation filters):

1. Program CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN], and so on, as required.
2. Start the module operation by writing 1 to CTRL_1[PDMIEN].

The module is ready to receive data from the input ports of the enabled channels.

57.5.2 Initialization procedure with HWVAD

In this section describes the initialization procedures to initialize MICFIL with HWVAD.

HWVAD can work in two modes: Energy-based mode or Envelope-based mode. Optionally, a ZCD can be enabled to improve the voice detection.

The Energy-based method separates the audio signal in frames and estimates the energy of these frames to detect voice activity. On the other hand, the Envelope-based method calculates the audio signal envelope to detect voice.

Both methods have the same detection performance, but the Energy-based is recommended for low-power applications where the chip can be asleep for long periods. Envelope-based detection is recommended for applications where the chip requires faster wake-up and it is not asleep for long periods.

ZCD is recommended to improve detection with fricatives phonemes for example /f/ or /s/) in noisy environments.

57.5.2.1 HWVAD in Energy-based mode

The sequence of steps to initialize MICFIL with HWVAD in Energy-based mode is as follows:

1. Program the control bits as desired for your application (CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN] etc.).
2. Configure general HWVAD parameters:
    a. CIC oversampling rate (VAD*n*_CTRL_1[VADCICOSR]).
    b. Configure the voice source channel (VAD*n*_CTRL_1[VADCHSEL]).
    c. Configure the internal initialization timer (VAD*n*_CTRL_1[VADINITT]) if required. More details in Initialization time.
    d. Write 1 to VAD*n*_CTRL_1[VADEN].
3. Configure input, noise and signal gains according to the environmental noise conditions:
    a. Configure the input gain (VAD*n*_CTRL_2[VADINPGAIN]).
    b. Configure the signal gain (VAD*n*_SCONFIG[VADSGAIN]).
    c. Configure the noise gain (VAD*n*_NCONFIG[VADNGAIN]).
4. Enable the high-pass filter if required (VAD*n*_CTRL_2[VADHPF]).
5. Configure VAD*n*_NCONFIG[VADNFILADJ] according to environmental noise conditions ([VADNFILADJ] = 16 is recommended).
6. Put HWVAD in Energy-based mode (default values):
    a. Keep VAD*n*_CTRL_2[VADFRENDIS] = 0.
    b. Keep VAD*n*_CTRL_2[VADPREFEN] = 0.
    c. Keep VAD*n*_SCONFIG[VADSFILEN] = 0.
    d. Keep VAD*n*_SCONFIG[VADSMAXEN] = 0.
    e. Keep VAD*n*_NCONFIG[VADNFILAUTO] = 1.
    f. Keep VAD*n*_NCONFIG[VADNMINEN] = 0.
    g. Keep VAD*n*_NCONFIG[VADNDECEN] = 0.
    h. Keep VAD*n*_NCONFIG[VADNOREN] = 0.
7. Initialize ZCD (ZCD initialization) if required.
8. Enable interrupts if required (VAD*n*_CTRL_1[VADIE] and [VADERIE]).
9. Start the overall module asserting CTRL_1[PDMIEN].
10. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when it detects voice.

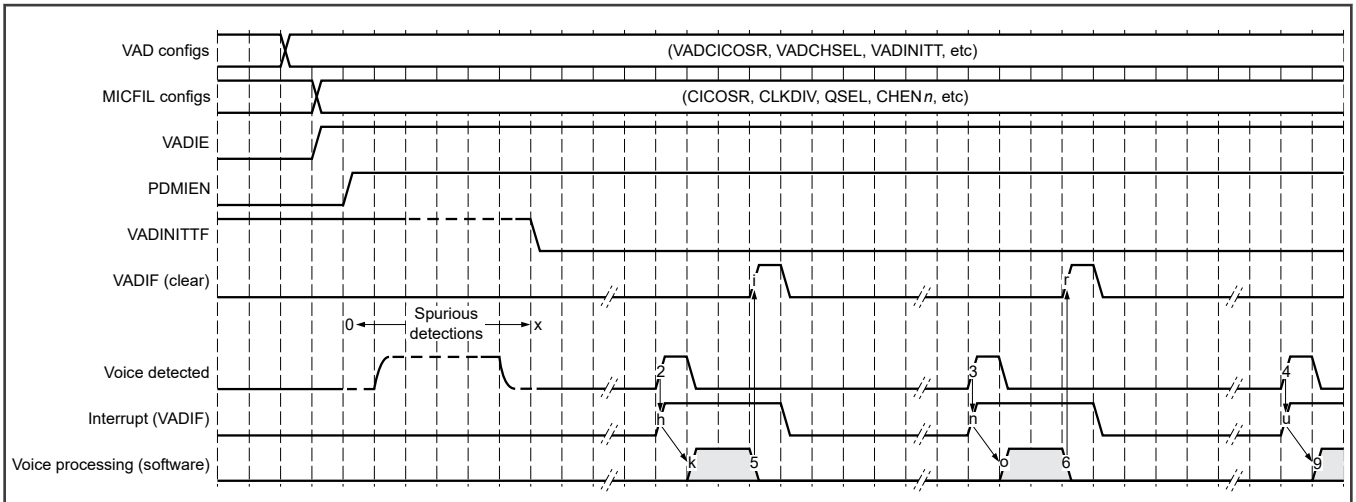


Figure 249. Energy-based initialization sequence

57.5.2.2 HWVAD in Envelope-based mode

The sequence of steps to initialize MICFIL with HWVAD in Envelope-based mode is as follows:

1. Program the control bits as desired for your application (CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN] etc.).
2. Configure general HWVAD parameters:
   a. CIC oversampling rate (VAD*n*_CTRL_1[VADCICOSR]).
   b. Configure the voice source channel (VAD*n*_CTRL_1[VADCHSEL]).
   c. Configure the internal initialization timer (VAD*n*_CTRL_1[VADINITT]) if required. More details in Initialization time.
   d. Write 1 to VAD*n*_CTRL_1[VADEN] field.
3. Configure input, noise and signal gains according to the environment noise conditions:
   a. Configure the input gain (VAD*n*_CTRL_2[VADINPGAIN]).
   b. Configure the signal gain (VAD*n*_SCONFIG[VADSGAIN]).
   c. Configure the noise gain (VAD*n*_NCONFIG[VADNGAIN]).
4. Enable the high-pass filter if required (VAD*n*_CTRL_2[VADHPF]).
5. Put HWVAD in Envelope-based mode:
   a. Keep VAD*n*_NCONFIG[VADNFILADJ] = 0.
   b. Write 1 to VAD*n*_CTRL_2[VADFRENDIS].
   c. Write 1 to VAD*n*_CTRL_2[VADPREFEN].
   d. Write 1 to VAD*n*_SCONFIG[VADSFILEN].
   e. Write 1 to VAD*n*_SCONFIG[VADSMAXEN].
   f. Write 0 to VAD*n*_NCONFIG[VADNFILAUTO].
   g. Write 1 to VAD*n*_NCONFIG[VADNMINEN].
   h. Write 1 to VAD*n*_NCONFIG[VADNDECEN].
   i. Write 1 to VAD*n*_NCONFIG[VADNOREN].
6. Initialize ZCD (ZCD initialization) if required.

7. Start the overall module asserting the CTRL_1[PDMIEN] bit.
8. Wait until VAD*n*_STAT[VADINITF] becomes 0.
9. Write VAD*n*_CTRL_1[VADST10] three times.
10. Enable interrupts if required (VAD*n*_CTRL_1[VADIE] and [VADERIE]).
11. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when it detects voice.

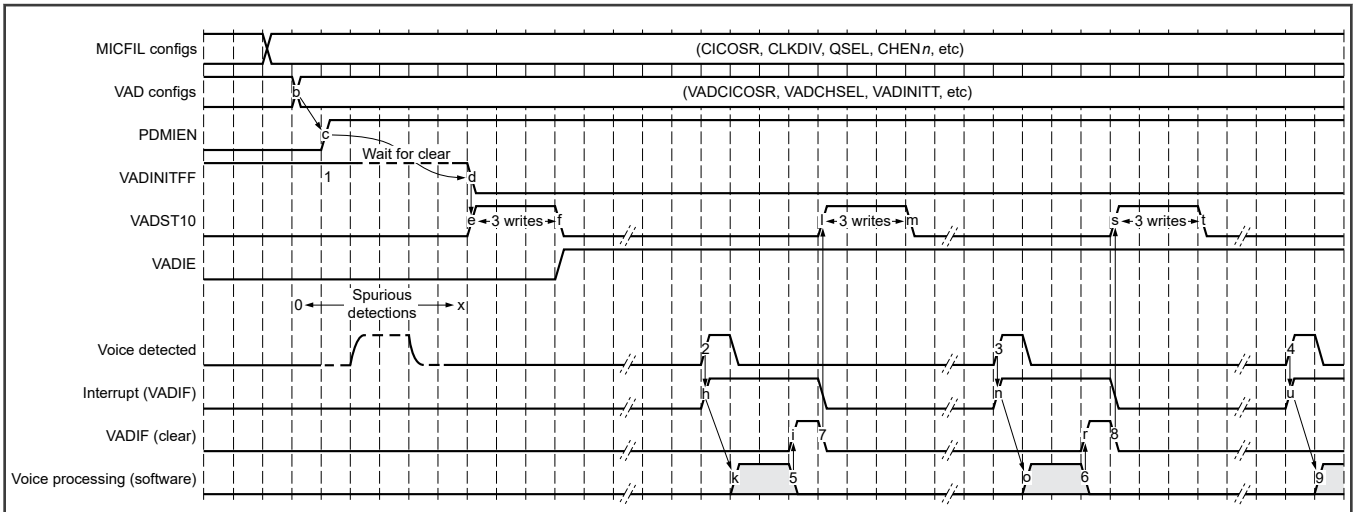


Figure 250. Envelope-based initialization sequence

57.5.2.3 ZCD initialization

The sequence of steps to initialize ZCD in Automatic mode is as follows:

1. Configure VAD*n*_ZCD[VADZCDADJ].
2. Configure an initial threshold in VAD*n*_ZCD[VADZCDTH] according to the environmental noise conditions if required.
3. Configure VAD*n*_ZCD[VADZCDAND].
4. Keep VAD*n*_ZCD[VADZCDAUTO] = 1.
5. Write 1 to VAD*n*_ZCD[VADZCDEN].

The sequence of steps to initialize ZCD in Manual mode is as follows:

1. Configure a threshold in VAD*n*_ZCD[VADZCDTH] according to the environmental noise conditions if required.
2. Configure VAD*n*_ZCD[VADZCDAND].
3. Write 0 to VAD*n*_ZCD[VADZCDAUTO].
4. Write 1 to VAD*n*_ZCD[VADZCDEN].

57.6 Application information

57.6.1 Reconfiguration procedure

Perform this procedure to reconfigure MICFIL:

1. Write 0 to CTRL_1[PDMIEN].
2. Wait until STAT[BSY_FIL] becomes 0.
3. Run a soft-reset cycle.

4. Run an initialization procedure as described in Procedure without HWVAD or Initialization procedure with HWVAD.

57.7 MICFIL register descriptions

This section provides the memory map and detailed description of all MICFIL registers. Any access to reserved areas can issue a target bus error indication.

You must implement the registers allocated in this memory map for up to 4 pairs of microphones, indicating that 8 channels are available.

57.7.1 MICFIL memory map

MICFIL base address: 4020_E000h

Offset	Register	Width (In bits)	Access	Reset value
0h	MICFIL Control 1 (CTRL_1)	32	RW	8000_0000h
4h	MICFIL Control 2 (CTRL_2)	32	RW	0008_0000h
8h	MICFIL Status (STAT)	32	RW	0000_0000h
10h	MICFIL FIFO Control (FIFO_CTRL)	32	RW	0000_0007h
14h	MICFIL FIFO Status (FIFO_STAT)	32	RW	0000_0000h
24h - 40h	MICFIL Output Result (DATACH0 - DATACH7)	32	R	0000_0000h
64h	MICFIL DC Remover Control (DC_CTRL)	32	R	0000_FFFFh
68h	MICFIL Output DC Remover Control (DC_OUT_CTRL)	32	RW	0000_0000h
74h	MICFIL Range Control (RANGE_CTRL)	32	RW	0000_0000h
7Ch	MICFIL Range Status (RANGE_STAT)	32	RW	0000_0000h
80h	Frame Synchronization Control (FSYNC_CTRL)	32	RW	0000_0000h
84h	Version ID (VERID)	32	R	0216_0000h
88h	Parameter (PARAM)	32	R	010B_0734h
90h	Voice Activity Detector 0 Control (VAD0_CTRL_1)	32	RW	0000_0000h
94h	Voice Activity Detector 0 Control (VAD0_CTRL_2)	32	RW	000A_0000h
98h	Voice Activity Detector 0 Status (VAD0_STAT)	32	RW	8000_0000h
9Ch	Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)	32	RW	0000_0000h
A0h	Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)	32	RW	8000_0000h
A4h	Voice Activity Detector 0 Noise Data (VAD0_NDATA)	32	R	0000_0000h
A8h	Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)	32	RW	0000_0004h

57.7.2 MICFIL Control 1 (CTRL_1)

Offset

Register	Offset
CTRL_1	0h

Function
Contains control fields for MICFIL.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MDIS	DOZEN	PDMIEN	DBG	0	DBGE	DISEL		ERREN	0		DECFILS	0			FSYNCEN
W					SRES											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MDIS	Module Disable Places MICFIL in DLL mode (see Disable/Low-Leakage (DLL) mode). The generated PDM_CLK stops, and consequently, the microphone generates no data. **NOTE** When this field = 1, only writes to MICFIL Control 1 (CTRL_1) and MICFIL Control 2 (CTRL_2) are allowed with the exception of writes to CTRL_1[DBG] and CTRL_1[SRES]. A target bus error indication is issued if you write to any other register. 0b - Normal mode 1b - DLL mode
30 DOZEN	Stop Enable When MICFIL receives a Stop request and this field is active, MICFIL enters DLL mode (see Disable/Low-Leakage (DLL) mode). 0b - Disables 1b - Enables
29	MICFIL Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
PDMIEN	Starts the module operation. Only the channels that CTRL_1[CH*n*EN] enables or VAD enabled by VADEN start operating. When this field becomes 0, MICFIL stops after it completes the remaining filter processing. **NOTE** You must enable this field as the last step when initializing the module (see Initialization). 0b - Stops MICFIL operation 1b - Starts MICFIL operation
28 DBG	Debug Mode Specifies the mode (Debug or Normal) that MICFIL operates in. MICFIL operates in Debug mode, depending on the value of CTRL_1[DBGE]. When Debug mode is disabled, MICFIL operates in Normal mode. See Debug mode for more on Debug mode. **NOTE** You can write to this field only when CTRL_1[MDIS] = 0. 0b - Normal 1b - Debug
27 SRES	Software Reset Provides the CPU with the capability to initialize MICFIL through the target-bus interface. The field always reads 0, and is effective only when CTRL_1[MDIS] = 0. It is a self-clearing field. **NOTE** You must write 1 to this field only when STAT[BSY_FIL] = 0. You can use this field to clear data in all filters. It resets the following blocks and registers: • All internal buffers in decimation filters and voice activity detectors • All FIFOs • STAT[CH*n*F] and MICFIL FIFO Status (FIFO_STAT) • Time generator 0b - No action 1b - Software reset
26 DBGE	Module Enable in Debug Enables operation in Debug mode (see Debug mode). 0b - Disables after completing the current frame 1b - Enables operation
25-24 DISEL	DMA Interrupt Selection

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Specifies the type of interrupt requests (DMA or filter) that MICFIL performs when the number of FIFO elements surpasses the configured watermark. Note that requests are only generated from the channels for which CTRL_1[CH*n*EN] = 1. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 00b - Disables DMA and interrupt requests 01b - Enables DMA requests 10b - Enables interrupt requests 11b - Reserved
23 ERREN	Error Interruption Enable Enables error interrupts (exceptions). The following conditions can lead to errors: • An overflow or underflow in the FIFOs (indicated by FIFO_STAT[FIFOOVF*n*] or FIFO_STAT[FIFOUND*n*], respectively) • An overflow or underflow in the decimations filters (indicated by RANGE_STAT[RANGEOVF*n*] or RANGE_STAT[RANGEUNF*n*], respectively) **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0b - Disables 1b - Enables
22-21 —	Reserved
20 DECFILS	Decimation Filter Enable in Stop Enables decimation filters to run when MICFIL receives a Stop request (see Low-Power mode for more information). 0b - Stops decimation filter 1b - Keeps decimation filter running
19-17 —	Reserved
16 FSYNCEN	Frame Synchronization Enable Enables frame synchronization (see Frame synchronization). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
15-8 —	Reserved
7-0 CHnEN	Channel n Enable Enables the decimation filter channel n (see Initialization for more information).

57.7.3 MICFIL Control 2 (CTRL_2)

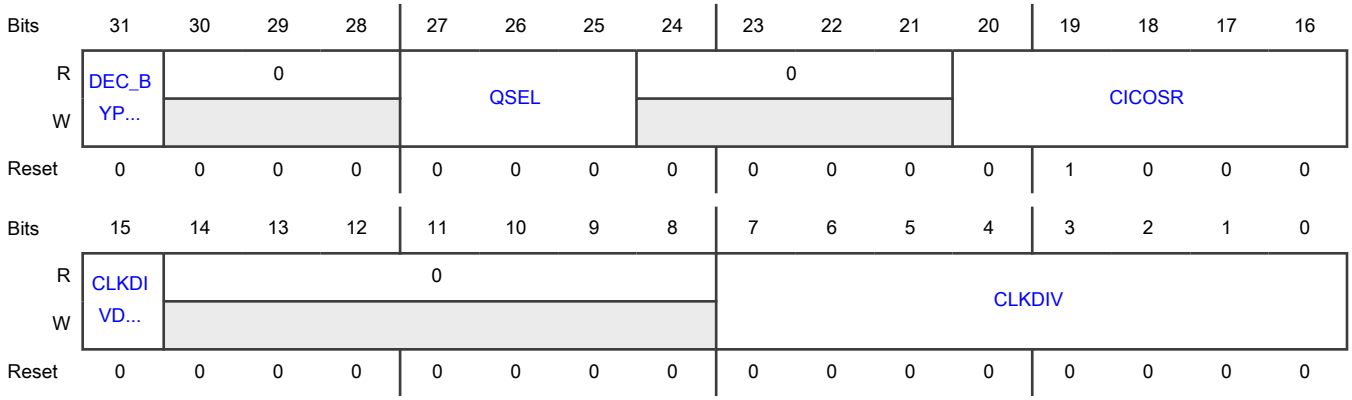
Offset

Register	Offset
CTRL_2	4h

Function

Contains control fields for MICFIL.

Diagram



Fields

Field	Function
31 DEC_BYPASS	Decimation Filter Bypass Enables the bypass path for the decimation filter. When enabled, PDM data can be written into FIFO directly without any processing. In this way, software can process PDM data later. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
30-28 —	Reserved
27-25 QSEL	Quality Mode Defines the actual quality mode (see Quality modes) of the decimation filter. 000b - Medium-Quality mode 001b - High-Quality mode 100b - Very-Low-Quality 2 mode 101b - Very-Low-Quality 1 mode 110b - Very-Low-Quality 0 mode 111b - Low-Quality mode
24-21 —	Reserved
20-16 CICOSR	CIC Decimation Rate Configures the CIC filter decimation rate (see Equation 9 for more information). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. OSR = (32 - CICOSR) ≤ 24 0_0000b-0_0111b - Not supported 0_1000b - CIC oversampling rate = 8 0_1001b - CIC oversampling rate = 9 0_1010b-1_1110b - ... 1_1111b - CIC oversampling rate = 31
15 CLKDIVDIS	Clock Divider Disable Disables and bypasses the internal clock divider (see Clock divider) to drive microphones directly with the application clock. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0b - Enables 1b - Disables
14-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
7-0 CLKDIV	Clock Divider Configures the internal clock divider value (see Clock divider for more information) using the following equation: $$PDM_CLK\ rate = \frac{MICFIL_CLK_ROOT}{2 \times floor(K \times CLKDIV)}$$ **NOTE** • You must not change the value of this field when STAT[BSY_FIL] = 1. • [CLKDIV] = 0 has the same effect as [CLKDIV] = 1. In Medium Quality mode and Very-Low Quality 0 mode, [CLKDIV] = 0 or 1 has the same effect as [CLKDIV] = 2. In High Quality mode, [CLKDIV] = 0 or 1 is not allowed. 0000_0000b - Internal clock divider value = 0 0000_0001b - Internal clock divider value = 1 0000_0010b-1111_1110b - ... 1111_1111b - Internal clock divider value = 255

57.7.4 MICFIL Status (STAT)

Offset

Register	Offset
STAT	8h

Function

Contains status flags for MICFIL.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BSY_FIL	0	0	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 BSY_FIL	Busy Flag Signalizes when at least a decimation filter channel or at least a HWVAD is running and the PDM_CLK is being generated. You could use this field as a confirmation that all decimation filters are stopped in a transition to Debug mode (when CTRL_1[DBGE] = 0) or DLL mode. 0b - MICFIL is stopped 1b - MICFIL is running
30 —	Reserved
29 —	Reserved
28-16 —	Reserved
15-8 —	Reserved
7-0 CHnF	Channel n Output Data Flag Indicates whether this channel's FIFO has surpassed its watermark value and the data is available in MICFIL Output Result (DATACH0 - DATACH7). This flag contributes to generate an interrupt or DMA request if CTRL_1[DISEL] enables it. Write 1 to CTRL_1[DISEL] to clear this flag. If the DMA request is set (CTRL_1[DISEL] = 1), [CHnF] is set according to the watermark level . In other cases (IRQ included), [CHnF] is set according to the watermark level and is cleared when you write 1 to it. **NOTE** If DMA is enabled, this flag clears itself after the DMA transaction is complete. 0b - Not surpassed 1b - Surpassed

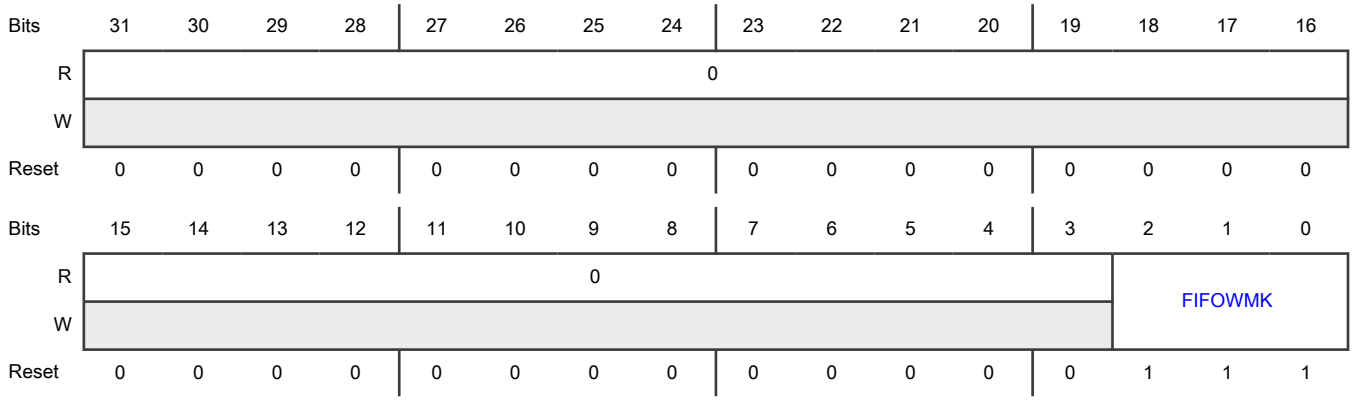
57.7.5 MICFIL FIFO Control (FIFO_CTRL)

Offset

Register	Offset
FIFO_CTRL	10h

Function
Contains MICFIL FIFO control fields.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 FIFOWMK	FIFO Watermark Control Controls the watermark of the FIFO used to set STAT[CH*n*F]. If the number of results in the FIFO is greater than the value of this field, STAT[CH*n*F] is set. A DMA request or interrupt can be also generated according to the configuration of CTRL_1[DISEL]. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. Before changing the watermark value, you must service all filtered data and clear channel flags.

57.7.6 MICFIL FIFO Status (FIFO_STAT)

Offset

Register	Offset
FIFO_STAT	14h

Function

Contains MICFIL FIFO status flags.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOUND7	FIFOUND6	FIFOUND5	FIFOUND4	FIFOUND3	FIFOUND2	FIFOUND1	FIFOUND0	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-8 FIFOUNDn	FIFO Underflow Exception Flag for Channel n Indicates an exceptional underflow condition in the FIFO. If the FIFO is empty and a read operation is requested, then this flag is set. 0b - No exception by FIFO underflow 1b - Exception by FIFO underflow
7-0 FIFOOVFn	FIFO Overflow Exception Flag for Channel n Indicates an exceptional overflow condition in the FIFO. If the FIFO is full and you receive an additional result to be written, this flag is set. 0b - No exception by FIFO overflow 1b - Exception by FIFO overflow

57.7.7 MICFIL Output Result (DATACH0 - DATACH7)

Offset

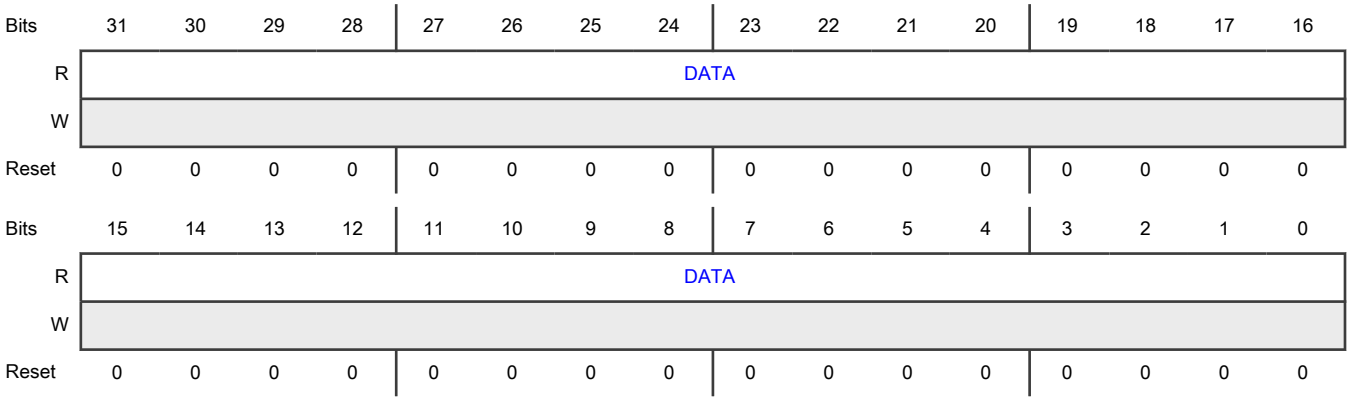
For n = 0 to 7:

Register	Offset
DATACHn	24h + (n × 4h)

Function

Contains the word value stored at the top of the channel's n FIFO.

Diagram



Fields

Field	Function
31-0 DATA	Channel n Data Reflects the word value stored at the top of the channel's n FIFO. The samples are integer-signed numbers in the two’s-complement format. **NOTE** Even if the FIFO data width is 32-bit, only the 24 more significant bits have information, and the other bits are always 0.

57.7.8 MICFIL DC Remover Control (DC_CTRL)

Offset

Register	Offset
DC_CTRL	64h

Function

Contains DC remover control fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCCONFIG7		DCCONFIG6		DCCONFIG5		DCCONFIG4		DCCONFIG3		DCCONFIG2		DCCONFIG1		DCCONFIG0	
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6 11-10: DCCONFIG5 9-8: DCCONFIG4 7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1 1-0: DCCONFIG0	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC remover. 00b - 20 Hz (PDM_CLK = 3.072 MHz) 01b - 13.3 Hz (PDM_CLK = 3.072 MHz) 10b - 40 Hz (PDM_CLK = 3.072 MHz) 11b - DC remover is bypassed

57.7.9 MICFIL Output DC Remover Control (DC_OUT_CTRL)

Offset

Register	Offset
DC_OUT_CTRL	68h

Function

Contains output DC remover configuration fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCCONFIG7		DCCONFIG6		DCCONFIG5		DCCONFIG4		DCCONFIG3		DCCONFIG2		DCCONFIG1		DCCONFIG0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6 11-10: DCCONFIG5 9-8: DCCONFIG4 7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1 1-0: DCCONFIG0	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC remover. 00b - 20 Hz (FS = 48 kHz) 01b - 13.3 Hz (FS = 48 kHz) 10b - 40 Hz (FS = 48 kHz) 11b - DC remover is bypassed

57.7.10 MICFIL Range Control (RANGE_CTRL)

Offset

Register	Offset
RANGE_CTRL	74h

Function

Contains range configuration information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RANGEADJ7				RANGEADJ6				RANGEADJ5				RANGEADJ4			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RANGEADJ3				RANGEADJ2				RANGEADJ1				RANGEADJ0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28: RANGEADJ7 27-24: RANGEADJ6 23-20: RANGEADJ5 19-16: RANGEADJ4 15-12: RANGEADJ3 11-8: RANGEADJ2 7-4: RANGEADJ1 3-0: RANGEADJ0	Channel n Range Adjustment Adjusts the dynamic range of the CIC filter. It is an unsigned non-negative value. You must configure this field within the following ranges: • If CTRL_2[QSEL] is in High-Quality and Very-Low-Quality 0 modes (HQ and VLQ0): — 2*OSR = 44, 46 unavailable — For 2*OSR = 12, 14 , 24, 28, 36, 42, 48: RANGEADJ*n* ≤ 28 - ceil(5*$\log_2$(2OSR)) — For other OSR values: RANGEADJ*n* ≤ 27 - ceil(5*$\log_2$(2OSR)) • If CTRL_2[QSEL] is in Medium-Quality and Very-Low Quality 1 modes (MQ and VLQ1): — For OSR = 12, 14, 17, 24: RANGEADJ*n* ≤ 28 - ceil(5*$\log_2$(OSR)) — For other OSR values: RANGEADJ*n* ≤ 27 - ceil(5*$\log_2$(OSR)) • If CTRL_2[QSEL] is in Low Quality and Very-Low-Quality 2 modes (LQ and VLQ2): — For OSR = 12, 14, 17, 24: RANGEADJ*n* ≤ 27 - ceil(5*$\log_2$(OSR)) — For other OSR values: RANGEADJ*n* ≤ 26 - ceil(5*$\log_2$(OSR)) where OSR = 32 - CICOSR, which is the CIC oversampling rate (see Equation 12), and the ceiling function, ceil(x), maps to the least integer greater than or equal to x.

57.7.11 MICFIL Range Status (RANGE_STAT)

Offset

Register	Offset
RANGE_STAT	7Ch

Function
Contains range status information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-16 RANGEUNFn	Channel n Range Underflow Error Flag Indicates an exceptional underflow condition in MICFIL range. **NOTE** If this error flag is asserted, channel n data is not reliable. You must adjust the value of RANGE_CTRL[RANGEADJ*n*] until this flag is not asserted anymore. 0b - No exception by range underflow 1b - Exception by range underflow
15-8 —	Reserved
7-0 RANGEOVFn	Channel n Range Overflow Error Flag Indicates an exceptional overflow condition in MICFIL range.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE If this error flag is asserted, channel n data is not reliable. You must adjust the value of RANGE_CTRL[RANGEADJ*n*] until this flag is not asserted anymore. 0b - No exception by range overflow 1b - Exception by range overflow

57.7.12 Frame Synchronization Control (FSYNC_CTRL)

Offset

Register	Offset
FSYNC_CTRL	80h

Function

Contains the frame synchronization window length value.

Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	FSYNCLEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FSYNCLEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
0-31 FSYNCLEN	Frame Synchronization Window Length Specifies the frame synchronization window length, which is equal to the value of FSYNC_CTRL[FSYNCLEN] + 1. See Frame synchronization for more information.

57.7.13 Version ID (VERID)

Offset

Register	Offset
VERID	84h

Function
Contains the major number, minor number, and feature specification number.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number.

57.7.14 Parameter (PARAM)

Offset

Register	Offset
PARAM	88h

Function
Contains MICFIL's general configuration information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUM_HWVAD				Reserved				HWVAD_Z...	Reserved	HWVAD_E...	HWVAD
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				DC_OUT_...	DC_BYPA...	LOW_POW...	FIL_OUT...	FIFO_PTRWID				NPAIR			
W																
Reset	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 NUM_HWVAD	Number of HWVADs Indicates the number of HWVADs.
23-20 —	Reserved
19 HWVAD_ZCD	HWVAD ZCD Active 0b - Disabled 1b - Active
18 —	Reserved
17 HWVAD_ENERGY_MODE	HWVAD Energy Mode Active 0b - Disabled 1b - Active
16 HWVAD	HWVAD Active 0b - Disabled 1b - Active
15-12 —	Reserved
11	Output DC Remover Bypass

Table continues on the next page...

Table continued from the previous page...

Field	Function
DC_OUT_BYPASS	Indicates whether the DC remover at the output is permanently disabled (bypassed) in silicon. 0b - Active 1b - Disabled
10 DC_BYPASS	Input DC Remover Bypass Indicates whether the DC remover at the input is permanently disabled (bypassed) in silicon. 0b - Active 1b - Disabled
9 LOW_POWER	Low-Power Decimation Filter 0b - Disables 1b - Enables
8 FIL_OUT_WIDTH_24B	Filter Output Width 0b - 16 bits 1b - 24 bits
7-4 FIFO_PTRWID	FIFO Pointer Width 0000b - 0 bits 0001b - 1 bit 0010b - 2 bits 0011b-1110b - ... 1111b - 15 bits
3-0 NPAIR	Number of Microphone Pairs 0000b - None 0001b - 1 pair 0010b - 2 pairs 0011b-1110b - ... 1111b - 15 pairs

57.7.15 Voice Activity Detector 0 Control (VAD0_CTRL_1)

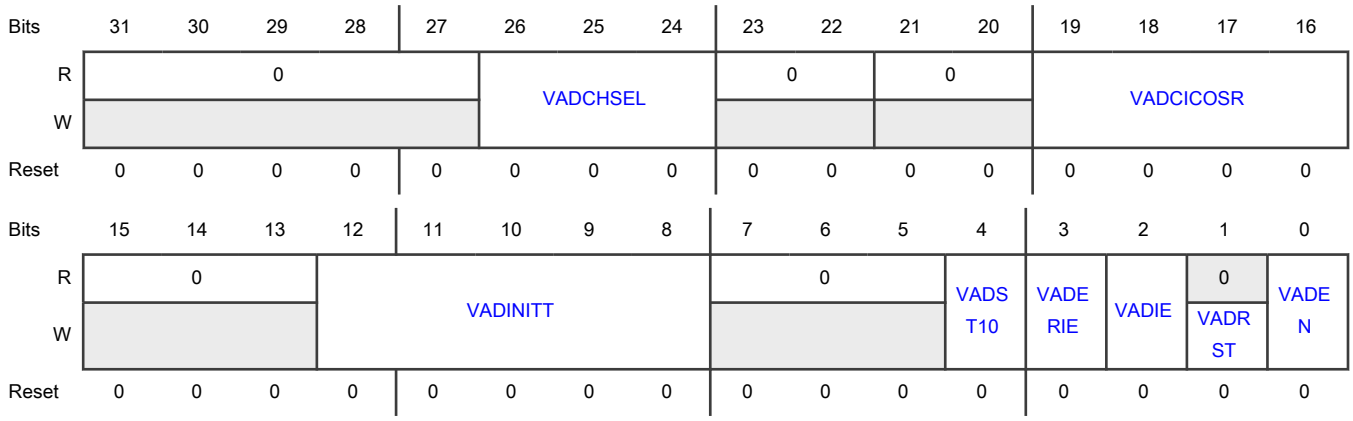
Offset

Register	Offset
VAD0_CTRL_1	90h

Function

Contains HWVAD configuration bits including the channel selector, CIC oversampling rate (OSR) and initialization time configuration, internal filters initialization, interrupt, reset, and HWVAD enable bits.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-24 VADCHSEL	Channel Selector Selects the number of channels which HWVAD works with. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 000b - PDM Microphone 0 Left 001b - PDM Microphone 0 Right 010b - PDM Microphone 1 Left 011b - PDM Microphone 1 Right 100b-101b - ... 110b - PDM Microphone 3 Left 111b - PDM Microphone 3 Right
23-22 —	Reserved
21-20 —	Reserved
19-16	CIC Oversampling Rate

Table continues on the next page...

Table continued from the previous page...

Field	Function
VADCICOSR	Defines the OSR of the CIC filter. The OSR is (16-CICOSR). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1.
15-13 —	Reserved
12-8 VADINITT	Initialization Time Selects the number of frames to be used to initialize the voice detection. The initialization time is the time taken by {[VADINITT] + 1} frames. During this period the output of the voice activity detector is forced to inactive. For more information, see Initialization time. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0_0000b - 0 0_0001b - 1 0_0010b-1_1110b - ... 1_1111b - 31
7-5 —	Reserved
4 VADST10	Internal Filters Initialization Initializes the signal and noise filter with pre-filter value. When [VADST10] = 1, there is no voice activity detection. **NOTE** If the use of this bit is required, it should stay pulsed by more than 2 cycles of the system clock. 0b - Normal operation 1b - Filters initialized
3 VADERIE	Error Interruption Enable Enables Error Interrupts in MICFIL. This error could be caused by an overflow in the input of the HWVAD and it is signalized by the VADINSATF flag. 0b - Disables 1b - Enables
2 VADIE	Interruption Enable Enables interrupts in MICFIL when voice activity event has been detected by HWVAD. See the VADIF for more information.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disables 1b - Enables
1 VADRST	HWVAD Reset Provides the CPU with the capability to initialize HWVAD through the target-bus interface. This bit always reads zero. This bit is self-cleared. See VAD soft-reset.
0 VADEN	HWVAD Enable Enables the use of HWVAD. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0b - Disables 1b - Enables

57.7.16 Voice Activity Detector 0 Control (VAD0_CTRL_2)

Offset

Register	Offset
VAD0_CTRL_2	94h

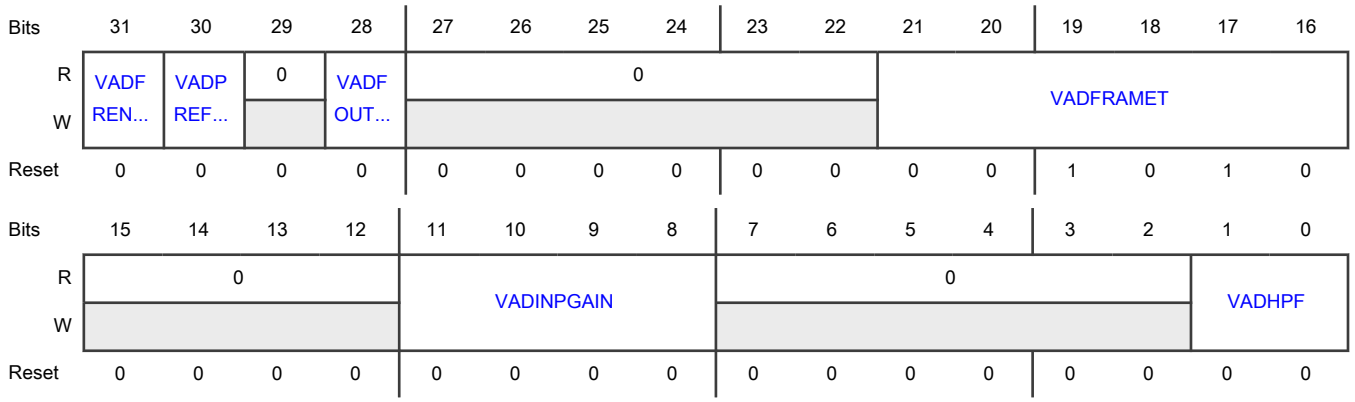
Function

Contains HWVAD configuration bits including the frame energy, pre-filter, force output, frame time, input gain, and high-pass filter configuration for the VAD.

NOTE

You must not change the value of this register when STAT[BSY_FIL] = 1.

Diagram



Fields

Field	Function
31 VADFRENDIS	Frame Energy Disable Disables the calculus of energy in a frame. For more information, see Noise filter. 0b - Enables 1b - Disables
30 VADPREFEN	Pre Filter Enable Enables a pre-filter in the input of noise and signal filters. For more information, see Pre-filter. 0b - Pre-filter bypassed 1b - Pre-filter enabled
29 —	Reserved
28 VADFOUTDIS	Force Output Disable Disables the output of the Energy and Envelope-based Detector. 0b - Enables 1b - Disables
27-22 —	Reserved
21-16 VADFRAMET	Frame Time Selects the scale value to control the frame time duration as described in Frame time setting. **NOTE** [VADFRAMET] = 0 is not allowed. 00_0000b - 1 00_0001b - 2 00_0010b-11_1110b - ... 11_1111b - 63
15-12 —	Reserved
11-8 VADINPGAIN	Input Gain Configures the gain value in the HWVAD input as a positive or negative (two's complement) number of bits to shift. 0000b - No shift 0001b - Shift 1 bit to the left

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0010b - Shift 2 bits to the left 0011b-0110b - ... 0111b - Shift 7 bits to the left 1000b - Shift 8 bits to the right 1001b - Shift 7 bits to the right 1010b-1110b - ... 1111b - Shift 1 bits to the right
7-2 —	Reserved
1-0 VADHPF	High-Pass Filter Configures the internal high-pass filter. 00b - Filter bypassed 01b - Cut-off frequency at 1750 Hz 10b - Cut-off frequency at 215 Hz 11b - Cut-off frequency at 102 Hz

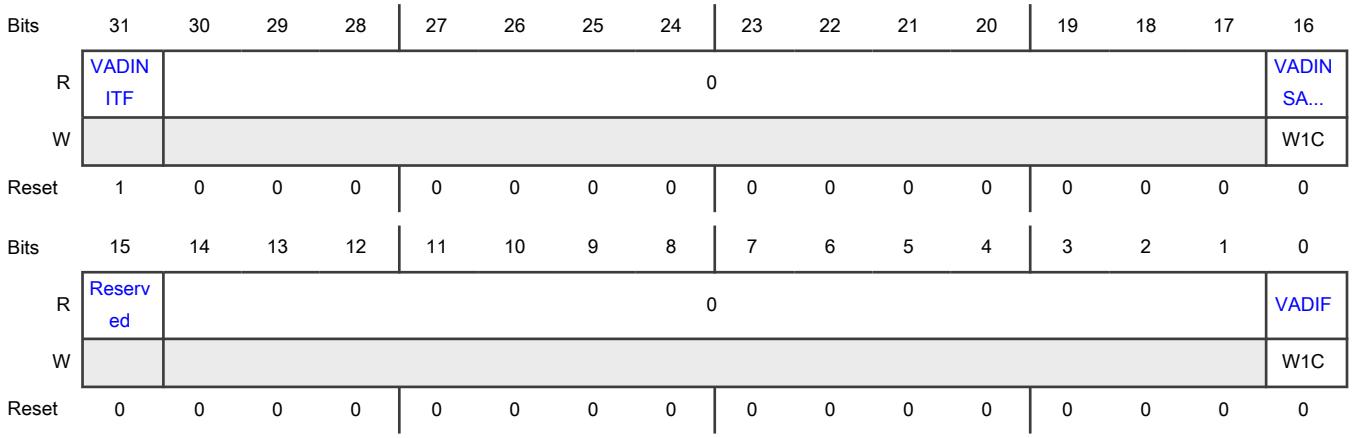
57.7.17 Voice Activity Detector 0 Status (VAD0_STAT)

Offset

Register	Offset
VAD0_STAT	98h

Function
Indicates the HWVAD status.

Diagram



Fields

Field	Function
31 VADINITF	Initialization Flag Signalizes when the HWVAD is being initialized according to VAD*n*_CTRL_1[VADINITT] value. 0b - Not being initialized 1b - Being initialized
30-17 —	Reserved
16 VADINSATF	Input Saturation Flag Indicates an exceptional saturation condition by an overflow or underflow in the HWVAD input. 0b - No exception 1b - Exception
15 —	Reserved
14-1 —	Reserved
0 VADIF	Interrupt Flag Indicates when voice activity has been detected by HWVAD. Write a logic 1 to this field to clear this flag. 0b - Not detected 1b - Detected

57.7.18 Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)

Offset

Register	Offset
VAD0_SCONFIG	9Ch

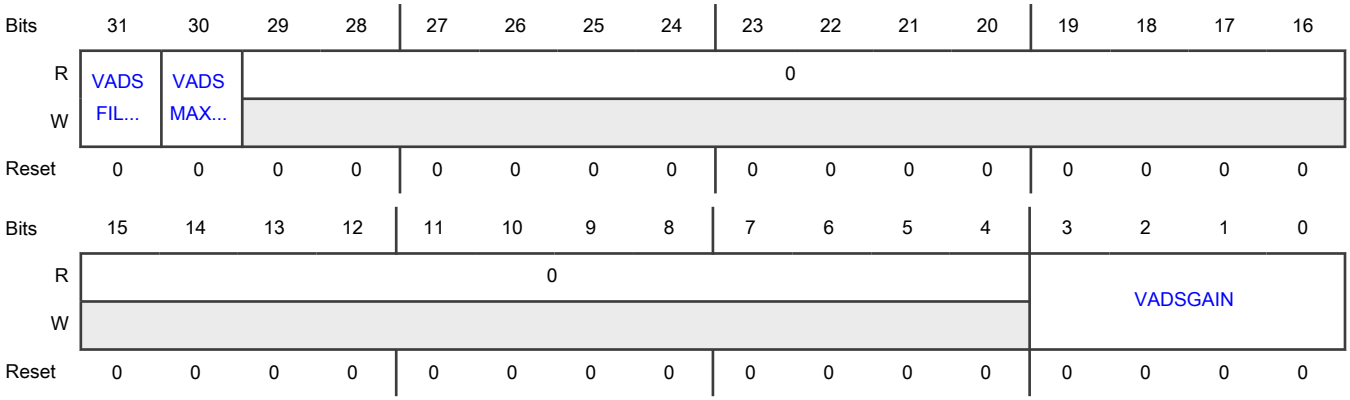
Function

Configures the HWVAD signal gain value and the signal filter.

NOTE

These fields must be configured when MICFIL is not running.

Diagram



Fields

Field	Function
31 VADSFILEN	Signal Filter Enable Enables the signal filter. For more information, see Signal filter. 0b - Disables 1b - Enables
30 VADSMAXEN	Signal Maximum Enable Selects whether or not a block to calculate the maximum value between the input and output of signal filter is used as signal estimation. For more information, see Signal filter. 0b - Maximum block bypassed 1b - Maximum block enabled
29-4 —	Reserved
3-0	Signal Gain

Table continues on the next page...

Table continued from the previous page...

Field	Function
VADSGAIN	Configures the gain value for the signal energy or envelope estimated. It is a unsigned value which is multiplied with the estimated signal data. **NOTE** When [VADSGAIN] = 0, the signal gain corresponds to 1. 0000b,0001b - Multiplier = 1 0010b - Multiplier = 2 0011b-1110b - ... 1111b - Multiplier = 15

57.7.19 Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)

Offset

Register	Offset
VAD0_NCONFIG	A0h

Function

Configures the HWVAD noise gain value and the noise filter.

NOTE

These fields must be configured when MICFIL is not running.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VADNFIL...	VADNMIN...	VADNDEC...	VADNOREN	0											
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			VADNFILADJ					0				VADNGAIN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Noise Filter Auto

Table continues on the next page...

Table continued from the previous page...

Field	Function
VADNFILAUTO	Selects whether the noise filter is activated automatically based on voice activity information. For more information, see Noise filter . 0b - Always enabled 1b - Enabled or disabled based on voice activity information
30 VADNMINEN	Noise Minimum Enable Selects whether or not a block to calculate the minimum value between the input and output of noise filter is used as noise estimation. For more information, see Noise filter . 0b - Minimum block bypassed 1b - Minimum block enabled
29 VADNDECEN	Noise Decimation Enable Selects whether or not the input of the noise filter is decimated. For more information, see Noise filter . 0b - Not decimated 1b - Decimated
28 VADNOREN	Noise OR Enable Enables an OR logic in the output of minimum noise estimator block. For more information, see Filter result gain setting . 0b - Not decimated 1b - Decimated
27-13 —	Reserved
12-8 VADNFILADJ	Noise Filter Adjustment Selects the adjustment value of the noise filter. For more information, see Noise filter. 0_0000b - 0 0_0001b - 1 0_0010b-1_1110b - ... 1_1111b - 31
7-4 —	Reserved
3-0 VADNGAIN	Noise Gain Configures the gain value for the noise energy or envelope estimated. It is an unsigned value which is multiplied with the estimated noise data.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When [VADNGAIN] = 0, the noise gain corresponds to 1. 0000b,0001b - Multiplier = 1 0010b - Multiplier = 2 0011b-1110b - ... 1111b - Multiplier = 15

57.7.20 Voice Activity Detector 0 Noise Data (VAD0_NDATA)

Offset

Register	Offset
VAD0_NDATA	A4h

Function

Stores the noise envelope calculated by HWVAD.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VADNDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 VADNDATA	Noise Data Stores the noise energy or noise envelope calculated by HWVAD. It can be used by software for a further voice activity detection.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field value should be ignored when VAD0_CTRL_2[VADFOUTDIS] = 1 or VAD0_CTRL_1[VADST10] = 1.

57.7.21 Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)

Offset

Register	Offset
VAD0_ZCD	A8h

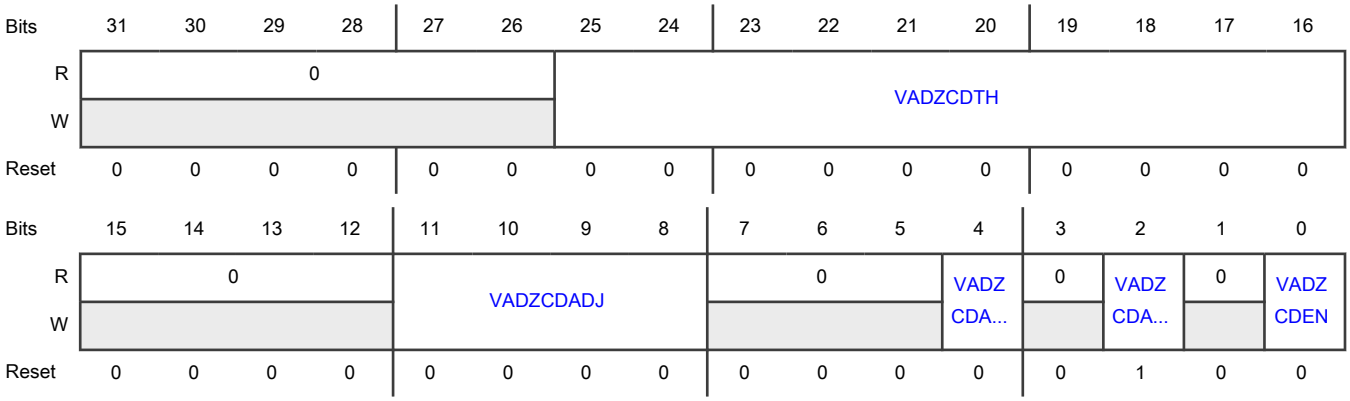
Function

Configures the zero-crossing detector (ZCD).

NOTE

These fields must be configured when MICFIL is not running.

Diagram



Fields

Field	Function
31-26 —	Reserved
25-16 VADZCDTH	ZCD Threshold Configures the initial value taken to estimate the ZCD threshold when it is estimated automatically ([VADZCDAUTO] = 1), otherwise ([VADZCDAUTO] = 0) this field is the static value of the ZCD threshold.
15-12	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
11-8 VADZCDADJ	ZCD Adjustment Selects the adjustment value of the threshold estimator. For more information, see Zero-crossing detector (ZCD). 0000b - 0 0001b - 1 0010b-1110b - ... 1111b - 15
7-5 —	Reserved
4 VADZCDAND	ZCD AND Behavior Selects whether the ZCD results are combined with the energy-based detection using AND or OR to form the voice activity detection event. 0b - OR 1b - AND
3 —	Reserved
2 VADZCDAUTO	ZCD Automatic Threshold Disables the automatic estimation of the ZCD threshold. 0b - Disables 1b - Enables
1 —	Reserved
0 VADZCDEN	ZCD Enable Enables the ZCD in HWVAD. 0b - Disables 1b - Enables

Chapter 58
Synchronous Audio Interface (SAI)

58.1 Chip-specific SAI information

Table 573. Reference links to related information

Topic	Related module	Reference
Full description	SAI	SAI
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

58.1.1 Module instances

This device has four instances of the SAI module, SAI0, SAI1, SAI2 and SAI3.

Table 574. SAI instances

	SAI0	SAI1	SAI2	SAI3
Located in	VDD2_COMP	VDD2_COMP	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, eDMA2, eDMA3

58.1.2 SAI TCR2, RCR2, and MCR configuration

Table 575. SAI configuration

Instance	Register bit	Configuration
SAI0-SAI2	TCR2[MSEL], RCR2[MSEL]	• 00b: compute_main_clk • 01b: MCLK from pin if the related pin is configured as input; or the clock configured by SAI012FCLKSEL and SAI012CLKDIV registers if the related pin is configured as output. • 10b: Reserved • 11b: Reserved
SAI3	TCR2[MSEL], RCR2[MSEL]	• 00b: sense_main_clk

Table continues on the next page...

Table 575. SAI configuration (continued)

Instance	Register bit	Configuration
		• 01b: MCLK from pin if the related pin is configured as input; or the clock configured by SAI3FCLKSEL and SAI3CLKDIV registers if the related pin is configured as output. • 10b: Reserved • 11b: Reserved
SAI0-SAI3	MCR[MSEL] MCR[DIVEN] MCR[DIV]	Reserved for this chip. Changes to these bit fields have no effect.

SAI1 and SAI2 bit clock, frame sync, and transmitter/receiver enable are synchronous with SAI0 in multiple SAI synchronous mode.

58.1.3 MCLK configuration

MCLK direction is configured by SYSCON register bits:

- SYSCON0.SAI0_MCLK_CTRL[SAIMCLKDIR] for SAI0, SAI1 and SAI2
- SYSCON1.SAI3_MCLK_CTRL[SAIMCLKDIR] for SAI3

Configure SAI register bit MCR[MOE]=1 whenever the SAI instance is using MCLK from any internal or external source.

If you want to use MCLK as the input, besides configuring the MCLK direction as input, you also need to configure MCLK as the clock source of audio controller clock for SAI as below:

- Select MCLK as the clock source of audio controller clock for SAI0, SAI1 and SAI2 by configuring
  1. CLKCTL0.AUDIOVDD2CLKCSEL[SEL] = 00b
  2. CLKCTL0.SAI012FCLKSEL[SEL] = 11b
- Select MCLK as the clock source of audio controller clock for SAI3 by configuring
  1. CLKTL1.AUDIOVDD1CLKSEL[SEL] =00b
  2. CLKCTL1.SAI3FCLKSEL[SEL] = 11b

58.1.4 Timestamp counter synchronization

This chip does not support timestamp counter synchronization between different SAI instances. See available timestamp select as below

Table 576. TSSEL configuration

Register bit	Configuration
TTCR[TSSEL]	• 00b: Increment when enabled • 01b: Increment when the receive timestamp counter is enabled • Others: Reserved
RTCR[TSSEL]	• 00b: Increment when enabled • 01b: Increment when the transmit timestamp counter is enabled • Others: Reserved

58.1.5 Initialization

1. Enable the clock to the SAI.
   - SAI*n* in compute domain via the CLKCTL0_PSCCTL2[SAI*n*] (*n* = 0-2)
   - SAI3 in sense domain via the CLKCTL1_PSCCTL1[SAI3]
2. Select a clock source and its divider for the SAI.
   - SAI*n* (*n* = 0-2) in compute domain via the CLKCTL0_SAI012FCLKSEL and CLKCTL0_SAI012CLKDIV registers
   - SAI3 in sense domain via the CLKCTL1_SAI3FCLKSEL and CLKCTL1_SAI3CLKDIV registers
3. Resets the internal transmitter logic and receiver logic by TCSR[SR] and RCSR[SR].

58.2 Overview

SAI provides an interface that supports full-duplex serial digital audio interfaces with frame synchronization formats such as I^2S, AC97, TDM, and Codec/DSP interfaces.

58.2.1 Block diagram

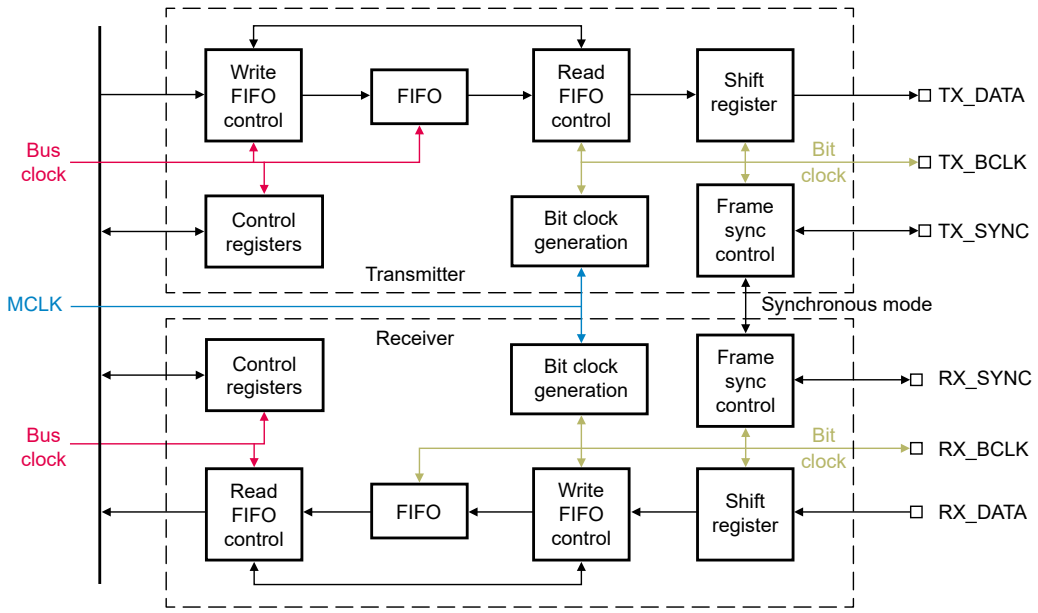


Figure 251. Block diagram

58.2.2 Features

Features can vary among chips and among SAI modules on the same chip. See the chip-specific SAI information at the beginning of this chapter.

- Transmitter with independent bit clock and frame synchronization supporting 1 data line
- Receiver with independent bit clock and frame synchronization supporting 1 data line
- Maximum frame size of 32 words per data line
- Word length of 8 to 32 bits
- Word length configured separately for the first word and remaining words in a frame
- Asynchronous 8 × 32-bit FIFO for each transmit and receive data line supports:

  - Graceful restart after FIFO error.
  - Automatic restart after FIFO error without software intervention.
  - 8-bit and 16-bit data packing into each 32-bit FIFO word.
- Independent 32-bit timestamp counters and bit counters for monitoring transmit and receive progress

58.3 Functional description

This section provides a complete functional description of SAI.

58.3.1 Modes of operation

- Run mode
- Stop mode
- Debug mode

58.3.1.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

58.3.1.2 Stop mode

There are two situations in which you can configure the SAI transmitter or receiver to continue operating in Stop mode:

- The SAI transmitter or receiver uses an externally generated bit clock.
- The SAI transmitter or receiver uses an audio clock (MCLK) that operates when the chip is in Stop mode.

Write 1 to TCSR[STOPE] to configure the transmitter to run in Stop mode, and write 1 to RCSR[STOPE] to configure the receiver to run in Stop mode.

The SAI transmitter and receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

When the chip enters Stop mode, if TCSR[STOPE] = 0, the transmitter is disabled after completing the current transmit frame. Similarly, when the chip enters Stop mode, if RCSR[STOPE] = 0, the receiver is disabled after completing the current receive frame. Entry into Stop mode is blocked (not acknowledged) when waiting for the transmitter and receiver to be disabled at the end of the current frame.

58.3.1.3 Debug mode

You can configure either or both the SAI transmitter and receiver to continue operating in Debug mode:

- Write 1 to TCSR[DBGE] to configure the transmitter to run in Debug mode.
- Write 1 to RCSR[DBGE] to configure the receiver to run in Debug mode.

When TCSR[DBGE] and RCSR[DBGE] are 0 and the chip enters Debug mode, SAI is disabled after completing the current transmit or receive frame. Debug mode does not affect the transmitter and receiver bit clocks.

58.3.2 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other. You can configure the SAI transmitter and receiver to operate with a synchronous bit clock and frame sync (see TCR2[SYNC] and RCR2[SYNC]).

If both the transmitter and receiver use the transmitter bit clock and frame sync:

- Configure the transmitter for asynchronous operation and the receiver for synchronous operation.
- Enable the receiver in Synchronous mode only after configuring both the transmitter and receiver.
- Enable the transmitter last and disable the transmitter first.

If both the transmitter and receiver use the receiver bit clock and frame sync:

- Configure the receiver for asynchronous operation and the transmitter for synchronous operation.
- Enable the transmitter in Synchronous mode only after configuring both the receiver and transmitter.
- Enable the receiver last and disable the receiver first.

When operating in Synchronous mode, the transmitter and receiver share only the bit clock, frame sync, and transmitter and receiver enable. Otherwise, the transmitter and receiver operate independently, although the configuration register settings must be consistent across both the transmitter and receiver.

58.3.3 Frame sync configuration

When enabled, SAI continuously transmits and receives frames of data. Each frame consists of a fixed number of words, with each word consisting of a fixed number of bits. Within each frame, you can mask any word, causing the receiver to ignore that word and the transmitter to 3-state during that word.

The frame sync signal indicates the start of a frame. A valid frame sync requires the detection of a rising edge (if active high) or falling edge (if active low). The transmitter or receiver cannot be busy with a previous frame. SAI ignores a valid frame sync (in Target mode) or does not generate one (in Controller mode) for the first four bit-clock cycles after enabling the transmitter or receiver.

You can configure the transmitter and receiver frame sync independently by using any of the following options:

- Generate externally or internally
- Configure to be active high or active low
- Assert with the first bit in the frame or one bit early
- Assert for a duration between one bit-clock cycle and the first word length
- Configure the frame length from 1 to 32 words
- Configure the word length from 8 to 32 bits, with separate configurations for the length of the first word and that of the remaining words
- Transmit or receive words MSB first or LSB first

You cannot change these configuration options after enabling the SAI transmitter or receiver.

58.3.4 Data FIFO

Each transmit and receive channel includes a 8 × 32-bit FIFO. Use Transmit Data (TDR0) and Receive Data (RDR0) to access FIFO data.

58.3.4.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit-wide register via the First Bit Shifted (FBT) configuration field. This field selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required FBT configuration are shown in Figure 252 for LSB-first configurations and Figure 253 for MSB-first configurations.

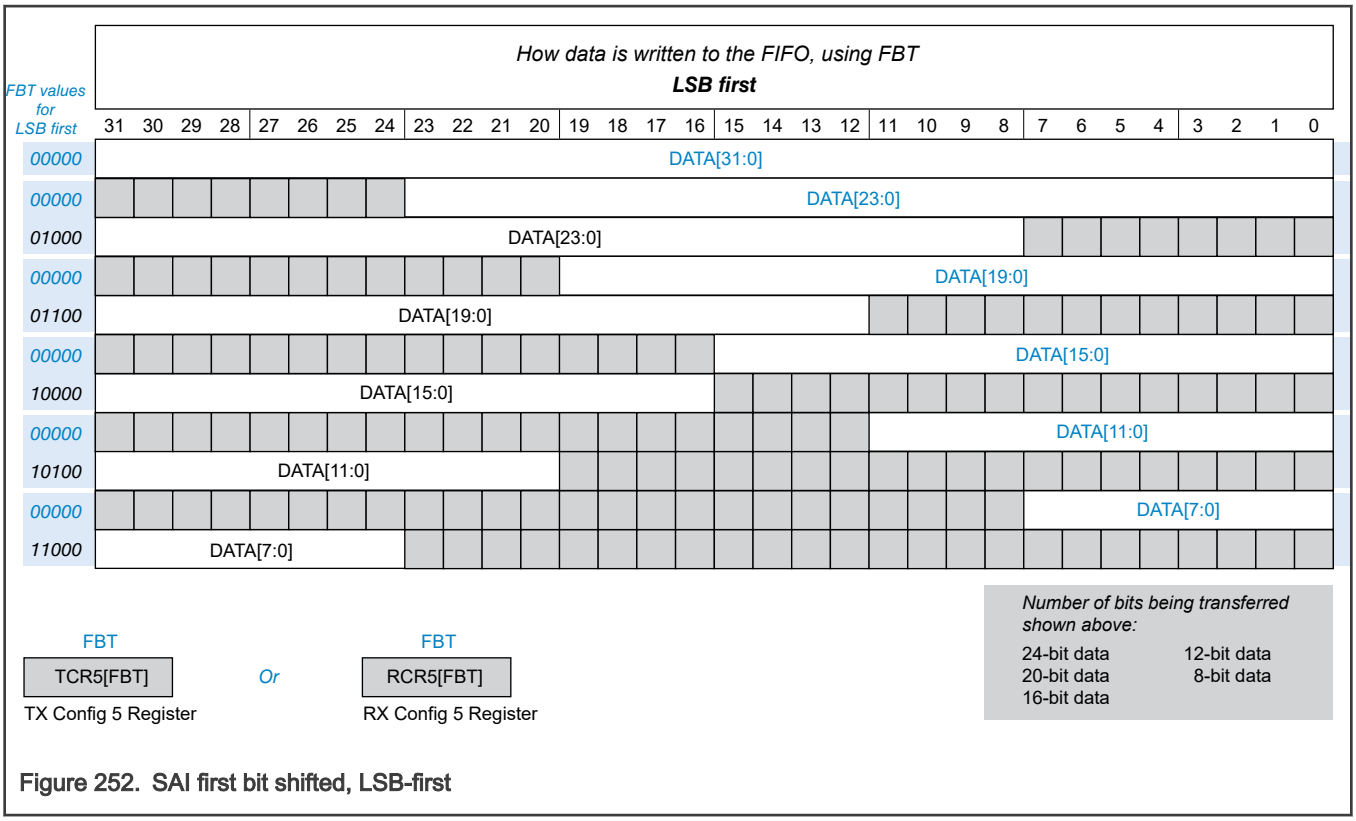


Figure 252. SAI first bit shifted, LSB-first

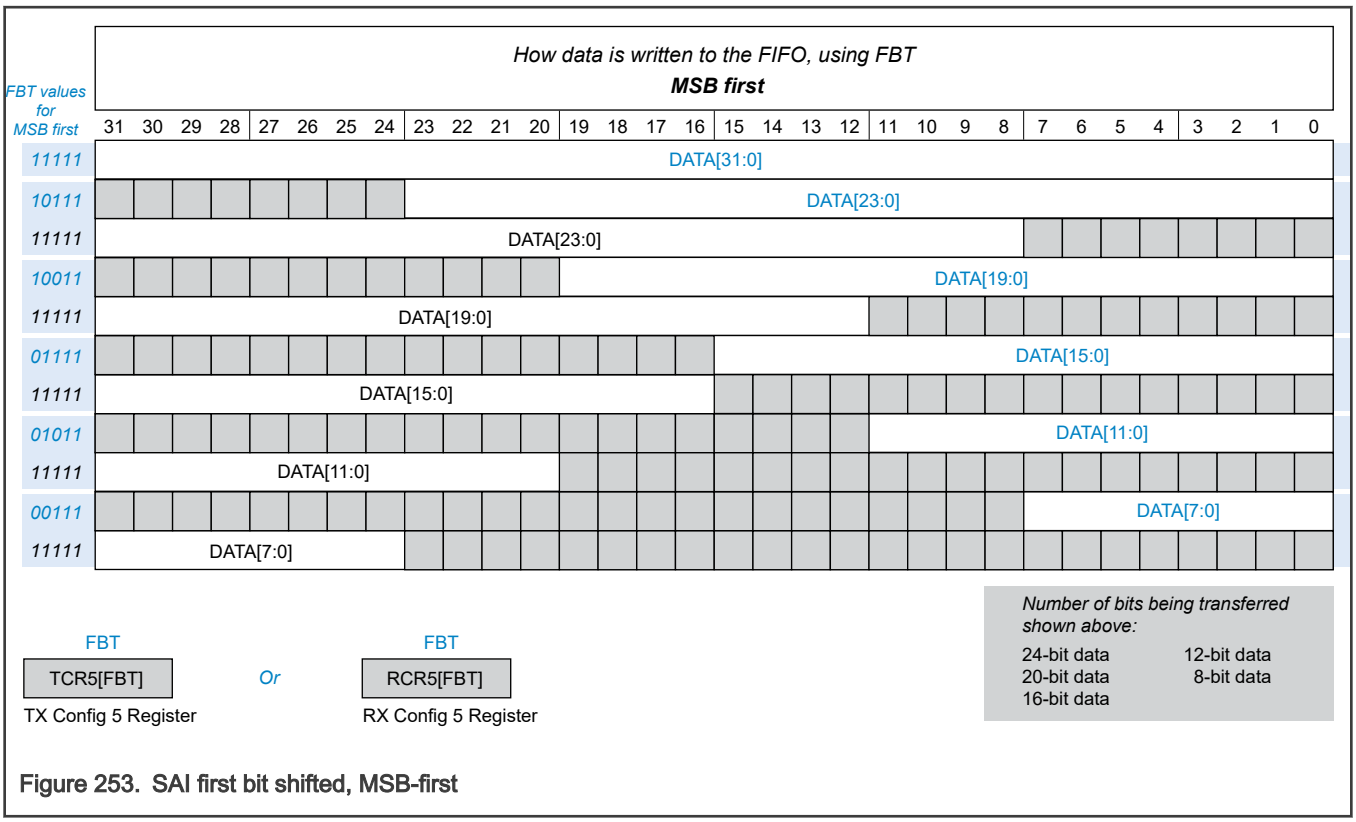


Figure 253. SAI first bit shifted, MSB-first

58.3.4.2 FIFO pointers

When writing to one of the Transmit Data (TDR0) registers, the write FIFO pointer of the corresponding Transmit FIFO register (TFR*n*[WFP]) increments after each valid write. SAI supports 8-bit, 16-bit, and 32-bit writes to TDR*n*, and the FIFO pointer increments after each individual write. Use only 8-bit writes when transmitting up to 8-bit data, and use only 16-bit writes when transmitting up to 16-bit data:

- If the transmit FIFO is full, SAI ignores writes to TDR*n*.
- If the transmit FIFO is empty, write to TDR*n* at least three bit-clock cycles before the start of the next unmasked word. This write avoids a FIFO underrun.
- Before enabling the transmitter, initialize the transmit FIFO with data. The transmitter starts a new frame after enabling the transmitter. If no data is in the FIFO, the transmitter immediately generates an error.

When reading one of the Receive Data (RDR0) registers, the read FIFO pointer of the corresponding Receive FIFO register (RFR*n*[RFP]) increments after each valid read. SAI supports 8-bit, 16-bit, and 32-bit reads from RDR*n*, and the FIFO pointer increments after each individual read. Use only 8-bit reads when receiving up to 8-bit data, and use only 16-bit reads when receiving up to 16-bit data:

- If the receive FIFO is empty, SAI ignores reads to RDR*n*.
- If the receive FIFO is full, read RDR*n* at least three bit-clock cycles before the end of an unmasked word. This read avoids a FIFO overrun.

58.3.4.3 FIFO packing

Using FIFO packing, you can store multiple 8-bit or 16-bit data words in one 32-bit FIFO word for the transmitter or receiver. You could emulate this feature by adjusting the number of bits per word and number of words per frame. FIFO packing, however, does not require even multiples of words per frame, and it fully supports word masking.

When FIFO packing is enabled, the FIFO pointers only increment when the full 32-bit FIFO word has been written (transmit) or read (receive). In this way, FIFO packing supports scenarios where different words within each frame are stored in different areas of memory.

Using 16-bit FIFO packing for transmitting, the transmit shift register loads at the start of each frame and after every second unmasked transmit word. The first transmitted word is taken from the 16-bit word at byte offset 0h. (The first bit is selected by TCR5[FBT], and you must configure it within this 16-bit word.) The second transmitted word is taken from the 16-bit word at byte offset 2h. (The first bit is selected by TCR5[FBT][3:0].) After the 16-bit word has been transmitted, the transmitter transmits logic zeroes until the start of the next word.

Using 16-bit FIFO packing for receiving, the receive shift register is stored after every second unmasked received word. If there is an odd number of unmasked-received words in each frame, the receive register is also stored, at the end of each frame. The first received word is stored in the 16-bit word at byte offset 0h. (The first bit is selected by RCR5[FBT], and you must configure it within this 16-bit word.) The second received word is stored in the 16-bit word at byte offset 2h. (The first bit is selected by RCR5[FBT][3:0].) The receiver ignores received data until the start of the next word after the 16-bit word has been received.

8-bit FIFO packing is similar to 16-bit packing, except four words are loaded or stored into each 32-bit FIFO word. The first word is stored in byte offset 0h, the second word in byte offset 1h, and so on. You must configure TCR5[FBT], RCR5[FBT], or both within byte offset 0h.

58.3.5 Word mask register

The SAI transmitter and receiver each have a word mask register (Transmit Mask (TMR) and Receive Mask (RMR)) that you can use to mask any word in the frame. The word mask register is double buffered. You can update it before the end of each frame to mask a particular word in the next frame.

TMR causes the transmit data pin to be 3-stated for the length of each selected word; the transmit FIFO is not read for masked words.

RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

58.3.6 Timestamp counter

The transmitter and receiver implement separate timestamp counters and bit counters. The bit counter increments at the end of each bit in a frame when the transmitter or receiver is enabled. The bit counter continues to increment if a word is masked or a FIFO underrun or overrun condition occurs. The bit counter does not increment if an external frame sync is configured and no frame sync is detected. You can reset the bit counter via software. Due to synchronization delays it can take up to 3 bit-clock cycles for the bit clock counter to continue incrementing after a bit counter reset.

The timestamp counter increments on the bus interface clock when it is enabled. You can configure it to synchronize to the bit counter by waiting for the bit counter to increment first (if configured, this synchronization is performed after the time counter is reset). The current value of the timestamp counter is latched whenever the bit counter increments. Reading the bit counter register causes the latched timestamp value to be saved in the bit counter timestamp register. You can reset the timestamp counter via software; a reset via software also resets the latched timestamp value and the bit counter timestamp register.

The timestamp counter can be synchronized between the transmitter and receiver; it can also be synchronized with the transmitter or receiver of another instance. When synchronizing timestamp counters, the controlling timestamp counter must be the last one enabled and the first one disabled. The timestamp counter software reset is not synchronized.

You can use the timestamp counter and bit counter to track the progress of the transmitter and receiver. You can also use it to calculate the relative frequency of the bit clock compared to the bus interface clock.

58.3.7 Clocking

- Audio clock
- Bit clock
- Bus clock

58.3.7.1 Audio clock

The audio clock (MCLK) generates the bit clock when you configure the receiver or transmitter for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio clocks to generate the bit clock.

The SAI peripheral controls the pin direction and postdivider of an audio clock. The postdivider can divide down the clock output on the MCLK signal pin without affecting the audio clock that the transmitter and receiver use.

The audio clock generation and selection is chip-specific. See chip-specific clocking information about how the audio clocks are generated.

58.3.7.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that an audio clock can generate internally or an external source can supply. SAI can also have synchronous bit clock and frame sync operation between the receiver and transmitter, or between multiple SAI peripherals. The transmitter and receiver configuration affects the bit clock and frame sync in the following ways:

- If you configure both the transmitter and receiver for asynchronous operation, each uses its own bit clock and frame sync.
- If you configure the transmitter for asynchronous operation and the receiver for synchronous operation, both use the transmitter bit clock and frame sync.
- If you configure the receiver for asynchronous operation and the transmitter for synchronous operation, both use the receiver bit clock and frame sync.

Your choice of synchronous or asynchronous operation selects the bit clock and frame sync used.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled.
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames.

In asynchronous operation, a SAI transmitter or receiver can use a bit clock externally generated by a SAI module instance that is disabled in Stop mode. In this case, disable the transmitter or receiver before entering Stop mode. This issue does not apply when the transmitter or receiver is in synchronous operation because all synchronous SAI modules are enabled and disabled simultaneously.

58.3.7.3 Bus clock

The control and configuration registers use the bus clock to generate synchronous interrupts and DMA requests.

NOTE

Although no minimum bus clock frequency is specified, the frequency must be fast enough (relative to the bit clock frequency) to serve the FIFOs. You must meet this requirement without generating a transmitter FIFO underrun or receiver FIFO overflow condition.

58.3.8 Reset

SAI is asynchronously reset on system reset. SAI has a Software reset and a FIFO reset.

58.3.8.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. The SAI receiver includes a software reset that resets all receiver internal logic, including bit clock generation, status flags, and FIFO pointers.

These software resets do not reset the configuration registers. The software resets remain asserted until you clear them via software.

58.3.8.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the value of the FIFO read pointer. This FIFO reset empties the FIFO contents. Use this reset after TCSR[FEF] becomes 1, and before SAI reinitializes the FIFO and TCSR[FEF] becomes 0. SAI asserts the FIFO reset for one cycle only.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the value of the FIFO write pointer. This FIFO reset empties the FIFO contents. Use this reset after RCSR[FEF] becomes 1 and SAI reads any remaining data from the FIFO, and before RCSR[FEF] becomes 0. SAI asserts the FIFO reset for one cycle only.

58.3.9 Interrupts and DMA requests

In SAI, the transmitter and receiver generate separate interrupts and DMA requests, but use the same status flags. SAI generates asynchronous versions of the transmitter and receiver interrupts to wake the CPU from Stop mode. SAI only generates asynchronous interrupts when the system clock is gated but the corresponding BCLK signal is active.

58.3.9.1 FIFO request flag

SAI sets the transmit FIFO request flag (TCSR[FRF]) when the number of entries in any enabled transmit FIFO is less than or equal to the transmit FIFO watermark configuration (TCR1[TFW]). SAI clears this flag when the number of entries in each enabled transmit FIFO is greater than TCR1[TFW].

SAI sets the receive FIFO request flag (RCSR[FRF]) when the number of entries in any enabled receive FIFO is greater than the receive FIFO watermark configuration (RCR1[RFW]). SAI clears this flag when the number of entries in each enabled receive FIFO is less than or equal to RCR1[RFW].

The FIFO request flag can generate an interrupt when TCSR[FRIE] = 1. It can generate a DMA request when TCSR[FRDE] = 1.

58.3.9.2 FIFO warning flag

SAI sets the transmit FIFO warning flag (TCSR[FWF]) when the number of entries in any of the enabled transmit FIFOs is empty. SAI clears this flag when the number of entries in each enabled transmit FIFO is not empty.

SAI sets the receive FIFO warning flag (RCSR[FWF]) when the number of entries in any of the enabled receive FIFOs is full. SAI clears this flag when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an interrupt when TCSR[FWIE] = 1. The flag can generate a DMA request when TCSR[FWDE] = 1.

58.3.9.3 FIFO error flag

SAI sets the transmit FIFO error flag (TCSR[FEF]) when any enabled transmit FIFOs underrun. After the error flag is set, all enabled transmit channels transmit zero data before SAI clears TCSR[FEF].

When you write 1 to TCR4[FCONT], the FIFO continues transmitting data following an underrun without software intervention. To ensure that data transmits in the correct order, the transmitter continues from the same word number in the frame that caused the FIFO to underrun. It continues only after new data is written to the transmit FIFO. In this case, clear TCSR[FEF] without reinitializing the transmit FIFOs.

SAI sets the receive FIFO error flag (RCSR[FEF]) when any enabled receive FIFO overflows. After the flag is set, all enabled receive channels discard their received data until SAI clears RCSR[FEF] and the next receive frame starts. Empty all enabled receive FIFOs before SAI clears RCSR[FEF].

When you write 1 to RCR4[FCONT], the FIFO continues receiving data following an overflow without software intervention. To ensure that data is received in the correct order, the receiver continues from the same word number in the frame that caused the FIFO to overflow. It continues only after data has been read from the receive FIFO. In this case, clear RCSR[FEF] without emptying the receive FIFOs.

TCSR[FEF] and RSCR[FEF] can only generate an interrupt.

58.3.9.4 Sync error flag

SAI sets a sync error flag (TCSR[SEF] or RCSR[SEF]) when both of these conditions are true:

- The frame sync is generated externally.
- The external frame sync asserts when the transmitter or receiver is busy with the previous frame.

SAI ignores the external frame sync assertion and sets the sync error flag. When this flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

TCSR[SEF] and RCSR[SEF] can only generate an interrupt.

58.3.9.5 Word start flag

SAI sets the word start flag (TCSR[WSF]) at the start of the second bit-clock cycle for the selected word. You can select this word via TCR3[WDFL].

TCSR[WSF] can only generate an interrupt.

58.4 External signals

Name	Function	I/O
TX_BCLK	Transmit bit clock: the bit clock is an input when externally generated and an output when internally generated.	I/O
TX_SYNC	Transmit frame sync: the frame sync is an input sampled synchronously by the bit clock when externally generated. It is an output generated	I/O

Table continues on the next page...

Table continued from the previous page...

Name	Function	I/O
	synchronously by the bit clock when internally generated.	
TX_DATA	Transmit data: the bit clock synchronously generates the transmit data; this signal is 3-stated when not transmitting a word.	O
RX_BCLK	Receive bit clock: the bit clock is an input when externally generated and an output when internally generated.	I/O
RX_SYNC	Receive frame sync: the frame sync is an input sampled synchronously by the bit clock when externally generated. It is an output generated synchronously by the bit clock when internally generated.	I/O
RX_DATA	Receive data: the bit clock synchronously samples the receive data.	I
MCLK	Audio clock: the audio clock is an input when externally generated and an output when internally generated.	I/O

58.5 Initialization

To initialize SAI:

1. Enable the clock to SAI.
2. Reset the internal transmitter logic and receiver logic by writing 1 to TCSR[SR] and RCSR[SR], respectively.

58.6 Memory map and register definition

A read or write access to an address from offset 108h and above results in a bus error.

58.6.1 SAI register descriptions

SAI provides an interface that supports full-duplex serial digital audio interfaces with frame synchronization formats.

58.6.1.1 SAI memory map

SAI0 base address: 4015_2000h

SAI1 base address: 4015_3000h

SAI2 base address: 4015_4000h

SAI3 base address: 4031_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0302_0002h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Parameter (PARAM)	32	R	0005_0301h
8h	Transmit Control (TCSR)	32	RW	0000_0000h
Ch	Transmit Configuration 1 (TCR1)	32	RW	0000_0000h
10h	Transmit Configuration 2 (TCR2)	32	RW	0000_0000h
14h	Transmit Configuration 3 (TCR3)	32	RW	0000_0000h
18h	Transmit Configuration 4 (TCR4)	32	RW	0000_0000h
1Ch	Transmit Configuration 5 (TCR5)	32	RW	0000_0000h
20h	Transmit Data (TDR0)	32	RW	0000_0000h
40h	Transmit FIFO (TFR0)	32	R	0000_0000h
60h	Transmit Mask (TMR)	32	RW	0000_0000h
70h	Transmit Timestamp Control (TTCR)	32	RW	0000_0000h
74h	Transmit Timestamp (TTSR)	32	R	0000_0000h
78h	Transmit Bit Count (TBCR)	32	R	0000_0000h
7Ch	Transmit Bit Count Timestamp (TBCTR)	32	R	0000_0000h
88h	Receive Control (RCSR)	32	RW	0000_0000h
8Ch	Receive Configuration 1 (RCR1)	32	RW	0000_0000h
90h	Receive Configuration 2 (RCR2)	32	RW	0000_0000h
94h	Receive Configuration 3 (RCR3)	32	RW	0000_0000h
98h	Receive Configuration 4 (RCR4)	32	RW	0000_0000h
9Ch	Receive Configuration 5 (RCR5)	32	RW	0000_0000h
A0h	Receive Data (RDR0)	32	R	0000_0000h
C0h	Receive FIFO (RFR0)	32	R	0000_0000h
E0h	Receive Mask (RMR)	32	RW	0000_0000h
F0h	Receive Timestamp Control (RTCR)	32	RW	0000_0000h
F4h	Receive Timestamp (RTSR)	32	R	0000_0000h
F8h	Receive Bit Count (RBCR)	32	R	0000_0000h
FCh	Receive Bit Count Timestamp (RBCTR)	32	R	0000_0000h
100h	MCLK Control (MCR)	32	RW	0000_0000h

58.6.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function
Contains version numbers for the module design and feature set.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Standard feature set 0000_0000_0000_0010b - Standard feature set with timestamp registers

58.6.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains parameter values implemented in the module.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												FRAME			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO				0				DATALINE			
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Fields

Field	Function
31-20 —	Reserved
19-16 FRAME	Frame Size Describes the maximum number of words per frame, which is 2^FRAME.
15-12 —	Reserved
11-8 FIFO	FIFO Size Describes the number of words in each FIFO, which is 2^FIFO.
7-4 —	Reserved
3-0 DATALINE	Number of Data Lines Contains the number of data lines implemented.

58.6.1.4 Transmit Control (TCSR)

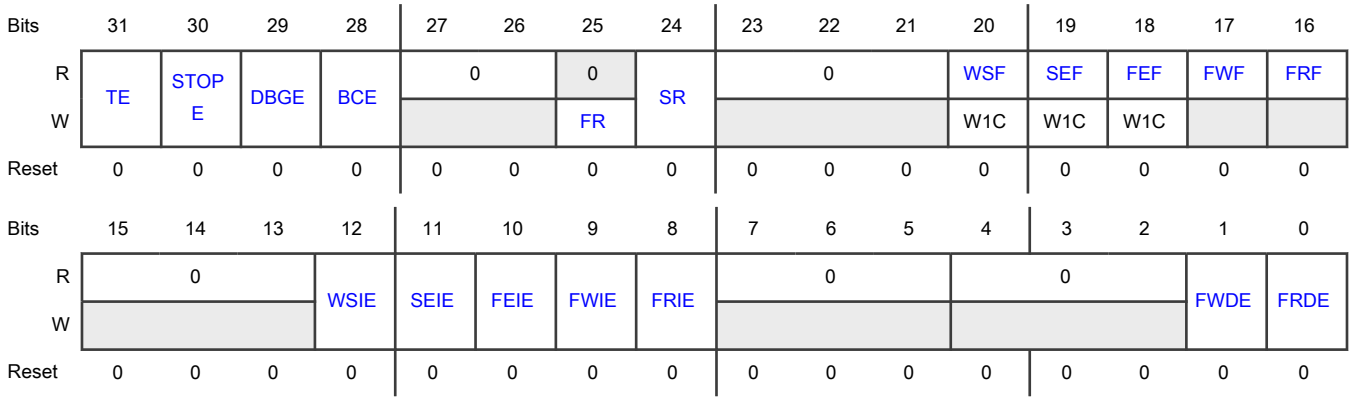
Offset

Register	Offset
TCSR	8h

Function

Contains transmitter enable fields including resets, error and interrupt enable fields, and error flag fields.

Diagram



Fields

Field	Function
31 TE	Transmitter Enable Enables the transmitter. When you write 0 to this field, the transmitter remains enabled, and this field remains 1 until the end of the current frame. 0b - Disable 1b - Enable (or transmitter has been disabled and has not yet reached the end of the frame)
30 STOPE	Stop Enable Enables transmitter operation in Stop mode. 0b - Disable 1b - Enable
29 DBGE	Debug Enable Enables transmitter operation in Debug mode, which does not affect the transmit bit clock. If you write 0 to this field, the transmitter operation is disabled after completing the current frame. 0b - Disable 1b - Enable
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from TCSR[TE]. This field automatically becomes 1 when TCSR[TE] becomes 1. When you write 0 to this field, the transmit bit clock remains enabled, and the field remains 1 until the end of the current frame. 0b - Disable 1b - Enable
27-26 —	Reserved
25	FIFO Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
FR	Empties the FIFO and sets the FIFO read and write pointers to the same value, which may or may not be zero. **NOTE** The FIFO reset is asserted for one cycle only. Reading this field always returns zero. SAI resets the FIFO pointers when the transmitter is disabled or the FIFO error flag is set. 0b - No effect 1b - FIFO reset
24 SR	Software Reset Resets the internal transmitter logic, including the FIFO read and write pointers. Software-visible registers are not affected, except for the status registers. **NOTE** The software reset remains asserted until software writes 0 to the field. 0b - No effect 1b - Software reset
23-21 —	Reserved
20 WSF	Word Start Flag Indicates whether the start of the configured word has been detected. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
19 SEF	Sync Error Flag Indicates whether an error in the externally generated frame sync has been detected. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
18 FEF	FIFO Error Flag Indicates whether an enabled transmit FIFO has underrun. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
17 FWF	FIFO Warning Flag Indicates whether an enabled transmit FIFO is empty. 0b - Not empty 1b - Empty
16 FRF	FIFO Request Flag Indicates whether the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0b - Watermark not reached 1b - Watermark reached
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables word start interrupts. 0b - Disable 1b - Enable
11 SEIE	Sync Error Interrupt Enable Enables sync error interrupts.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
10 FEIE	FIFO Error Interrupt Enable Enables FIFO error interrupts. 0b - Disable 1b - Enable
9 FWIE	FIFO Warning Interrupt Enable Enables FIFO warning interrupts. 0b - Disable 1b - Enable
8 FRIE	FIFO Request Interrupt Enable Enables FIFO request interrupts. 0b - Disable 1b - Enable
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables the DMA warning. 0b - Disable 1b - Enable
0 FRDE	FIFO Request DMA Enable Enables DMA requests. 0b - Disable 1b - Enable

58.6.1.5 Transmit Configuration 1 (TCR1)

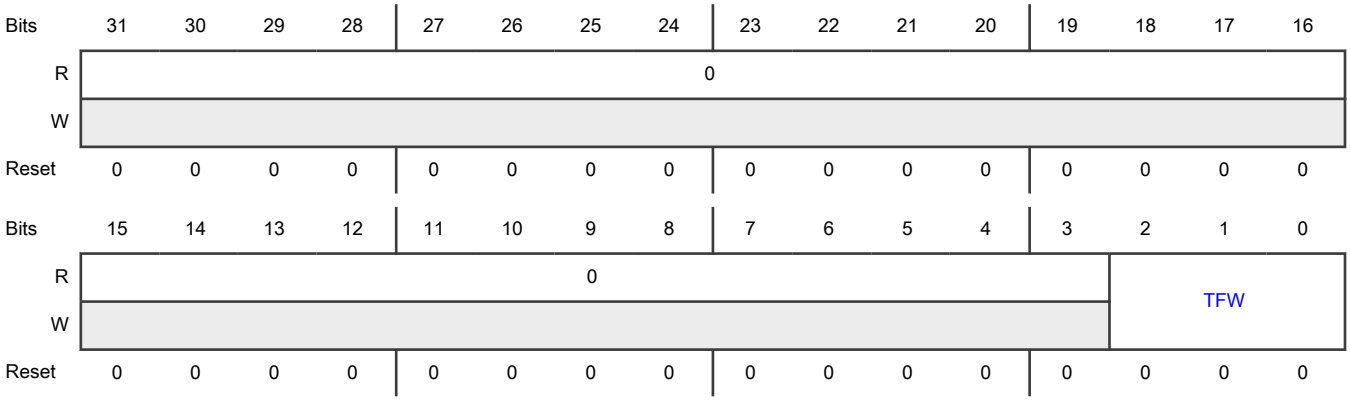
Offset

Register	Offset
TCR1	Ch

Function

Configures the watermark level for all enabled transmit channels.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 TFW	Transmit FIFO Watermark Indicates the number of 32-bit FIFO words in the watermark level of the transmit FIFO. 000b - 0 001b - 1 010b-110b - (TFW value) 111b - 7

58.6.1.6 Transmit Configuration 2 (TCR2)

Offset

Register	Offset
TCR2	10h

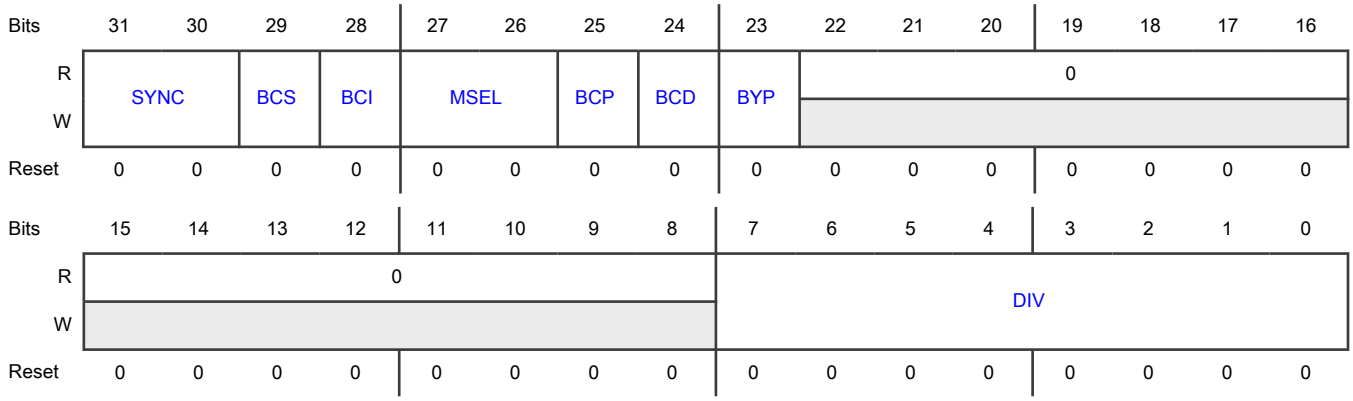
Function

Contains the SYNC mode and clock setting fields.

NOTE

Do not alter this register when TCSR[TE] is 1.

Diagram



Fields

Field	Function
31-30 SYNC	Synchronous Mode Configures Asynchronous and Synchronous modes of operation. When configured for Synchronous mode, you must configure the receiver for asynchronous operations.

NOTE

This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI0	TCR2[30]	TCR2[31]
SAI1	TCR2	—
SAI2	TCR2	—
SAI3	TCR2[30]	TCR2[31]

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
SAI0 SAI3	0b - Asynchronous mode 1b - Synchronous with receiver
SAI1 SAI2	00b - Asynchronous mode 01b - Synchronous with receiver 10b - Synchronous with another SAI transmitter 11b - Synchronous with another SAI receiver

Table continued from the previous page...

Field	Function
29 BCS	Bit Clock Swap Swaps the bit clock used by the transmitter. When you configure the transmitter in Asynchronous mode and this field is 1, the receiver bit clock (RX_BCLK) clocks the transmitter. This configuration allows the transmitter and receiver to share a bit clock, while the transmitter continues to use the transmit frame sync (TX_SYNC). When you configure the transmitter in Synchronous mode, you must write the same value to the transmitter BCS field and the receiver BCS field. When both are 1, the transmitter and receiver are both clocked by the transmitter bit clock (TX_BCLK) but both use the receiver frame sync (RX_SYNC). 0b - Use the normal bit clock source 1b - Swap the bit clock source
28 BCI	Bit Clock Input Enables the internal logic to be clocked as if the bit clock is generated externally. When this field is 1 while using an internally generated bit clock in Synchronous or Asynchronous mode, the bit clock used by the transmitter is delayed by the pad output delay. (The pad input clocks the transmitter as if the clock is externally generated.) This setting decreases the data input setup time, but increases the data output valid time. Use the Target mode timing from the data sheet for the transmitter when this field is 1. In Synchronous mode, this field allows the transmitter to use the Target mode timing from the data sheet, while the receiver uses the Controller mode timing. This field has no effect when configured for an externally generated bit clock. **NOTE** When BCI = 1, both the input buffer and output buffer must be enabled for the BCLK pad. 0b - Disable 1b - Enable
27-26 MSEL	MCLK Select Selects the audio clock option used to generate an internally generated bit clock. This field has no effect when SAI is configured for an externally generated bit clock. **NOTE** Depending on the chip, some audio clock options might not be available. See the chip-specific information for the meaning of each option. 00b - Bus clock 01b - Audio clock (MCLK) option 1 10b - Audio clock (MCLK) option 2 11b - Audio clock (MCLK) option 3
25 BCP	Bit Clock Polarity

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Configures the polarity of the bit clock. If you write 0 to this field, the bit clock becomes active high with drive outputs on rising edge and sample inputs on falling edge. If you write 1 to this field, the bit clock becomes active low with drive outputs on falling edge and sample inputs on rising edge. 0b - Active high 1b - Active low
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0b - Generate externally in Target mode 1b - Generate internally in Controller mode
23 BYP	Bit Clock Bypass Bypasses the bit clock divider. When bypassed, the internal bit clock is divide-by-one of the audio clock. When not bypassed, the internal bit clock is generated from the bit clock divider. 0b - Disable 1b - Enable
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Determines the value by which SAI divides down the audio clock to generate the bit clock (when configured for an internal bit clock). The division value is (DIV + 1) × 2.

58.6.1.7 Transmit Configuration 3 (TCR3)

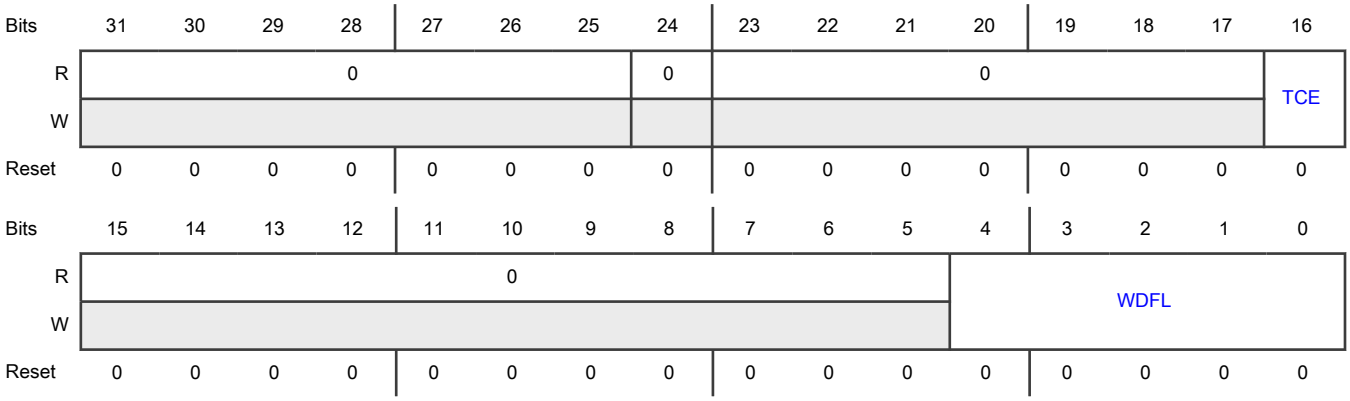
Offset

Register	Offset
TCR3	14h

Function

Contains the transmit channel settings.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-17 —	Reserved
16 TCE	Transmit Channel Enable Enables the corresponding data channel for transmit operations. Changing this field takes effect immediately for generating the FIFO request and warning flags. It takes effect at the end of each frame for transmit operations. The width of this field = the number of transmit channels (call it *n*). For example, if this field is two bits wide, bit position 16 refers to transmit channel 1 and bit position 17 refers to transmit channel 2. Writing 1 to bit 16 enables transmit channel 1, and setting bit 17 enables transmit channel 2. Writing 1 to bit *n* enables transmit channel *n*. For each transmit data channel: • 0b – Disable • 1b – Enable
15-5 —	Reserved
4-0 WDFL	Word Flag Configuration Selects the word that sets the word start flag (TCSR[WSF]). The value must be one less than the word number. For example, writing 0 selects the first word in the frame. When you configure this field with a value greater than TCR4[FRSZ], TCSR[WSF] is never set.

58.6.1.8 Transmit Configuration 4 (TCR4)

Offset

Register	Offset
TCR4	18h

Function

Contains the transmit fields for FIFO Combine mode, FIFO Packing mode, and frame sync settings.

NOTE

Do not alter this register when TCSR[TE] is 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FCONT	0		FPACK		0				FRSZ			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0		CHMOD	MF	FSE	ONDEM	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when SAI must continue transmitting after a FIFO error is detected. When this field is 0 and a FIFO error occurs, SAI continues from the start of the next frame after the FIFO error flag clears. When this field is 1 and a FIFO error occurs, SAI continues from the same word that caused the FIFO error after the FIFO warning flag clears. 0b - Continue from the start of the next frame 1b - Continue from the same word that caused the FIFO error
27-26 —	Reserved
25-24	FIFO Packing Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
FPACK	Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8 bits or 16 bits, only the first 8 bits or 16 bits are loaded from the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO write pointer only increments when you write the full 32-bit FIFO word. 00b - Disable FIFO packing 01b - Reserved 10b - Enable 8-bit FIFO packing 11b - Enable 16-bit FIFO packing
23-21 —	Reserved
20-16 FRSZ	Frame Size Configures the number of words in each frame. The value must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit-clock cycles. The value must be one less than the number of bit-clock cycles. For example, write 0 for the frame sync to assert for one bit-clock cycle only. You cannot configure the sync width to be longer than the first word of the frame.
7-6 —	Reserved
5 CHMOD	Channel Mode Specifies the mode of transmit data pins. In TDM mode, transmit data pins are 3-stated when slots are masked or channels are disabled. In Output mode, transmit data pins are never 3-stated, and they output zero when slots are masked or channels are disabled. 0b - TDM mode 1b - Output mode
4 MF	MSB First Configures what is transmitted first: LSB or MSB. 0b - LSB 1b - MSB
3 FSE	Frame Sync Early Determines when frame sync is asserted.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - First bit of the frame 1b - One bit before the first bit of the frame
2 ONDEM	On-Demand Mode Determines when the internal frame sync is generated. When this field is 1, and the frame sync is generated internally, a frame sync is only generated after the FIFO warning flag is cleared. 0b - Generated continuously 1b - Generated after the FIFO warning flag is cleared
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0b - Active high 1b - Active low
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0b - Generated externally in Target mode 1b - Generated internally in Controller mode

58.6.1.9 Transmit Configuration 5 (TCR5)

Offset

Register	Offset
TCR5	1Ch

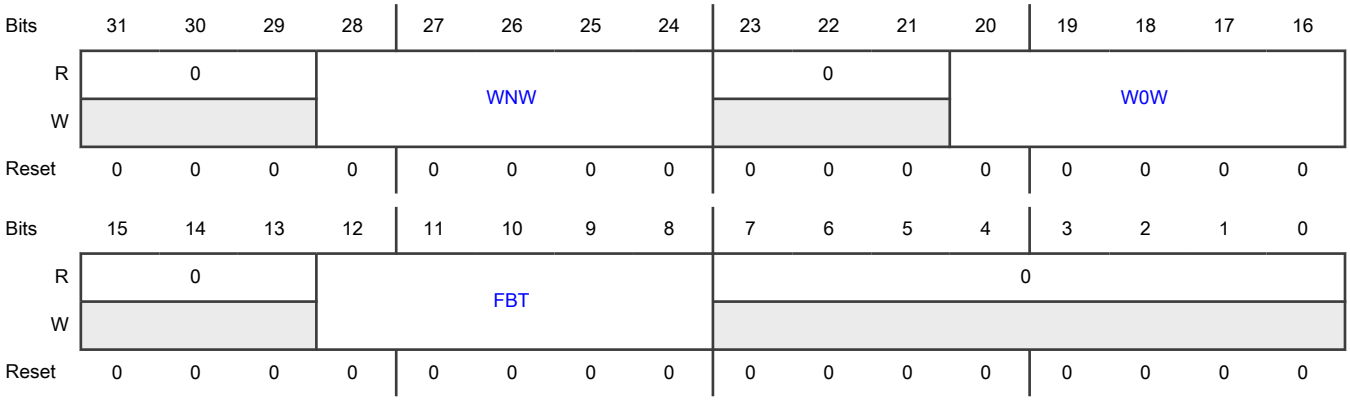
Function

Contains transmit word width and bit index settings.

NOTE

Do not alter this register when TCSR[TE] is 1.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, except for the first one in the frame. The value written must be one less than the number of bits per word. SAI does not support word widths of less than 8 bits. SAI does not support bit settings 0–6 (0b–0_0110b). 0_0111b - 8 0_1000b - 9 0_1001b-1_1110b - (WNW value + 1) 1_1111b - 32
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word of each frame. The value written must be one less than the number of bits in the first word. SAI does not support word widths of less than 8 bits when there is only one word per frame. SAI does not support bit settings 0–6 (0b–0_0110b). 0_0111b - 8 0_1000b - 9 0_1001b-1_1110b - (W0W value + 1) 1_1111b - 32
15-13 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
12-8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB-first, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB-first, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB-first. The value written must be less than or equal to 31-word width when configured for LSB-first. See Data alignment for details. 0_0000b - 0 0_0001b-1_1110b - FBT 1_1111b - 31
7-0 —	Reserved

58.6.1.10 Transmit Data (TDR0)

Offset

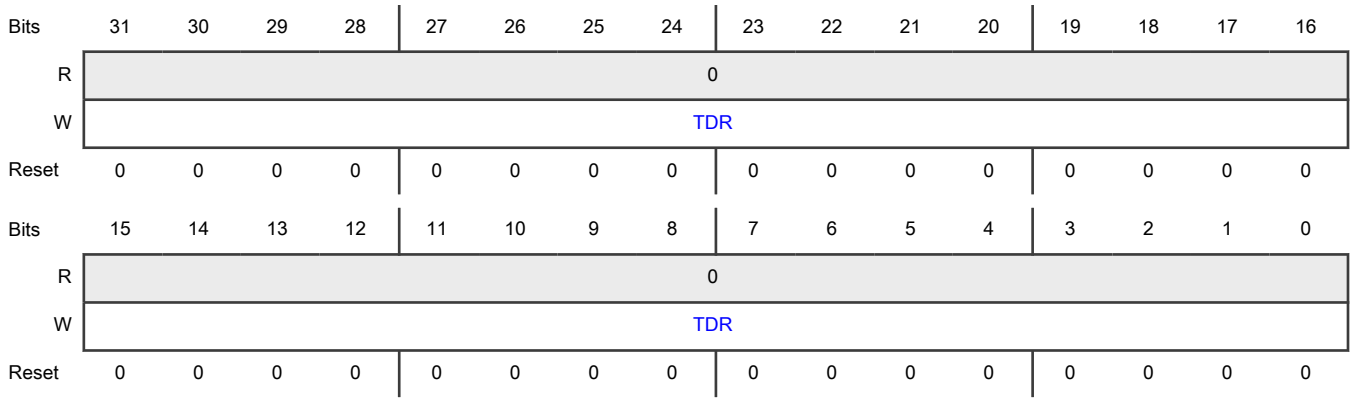
Register	Offset
TDR0	20h

Function

Contains transmit data.

When the transmit FIFO is not full, writes to this register push the data written into the transmit data FIFO. When the transmit FIFO is full, SAI ignores writes to this register.

Diagram



Fields

Field	Function
31-0 TDR	Transmit Data

58.6.1.11 Transmit FIFO (TFR0)

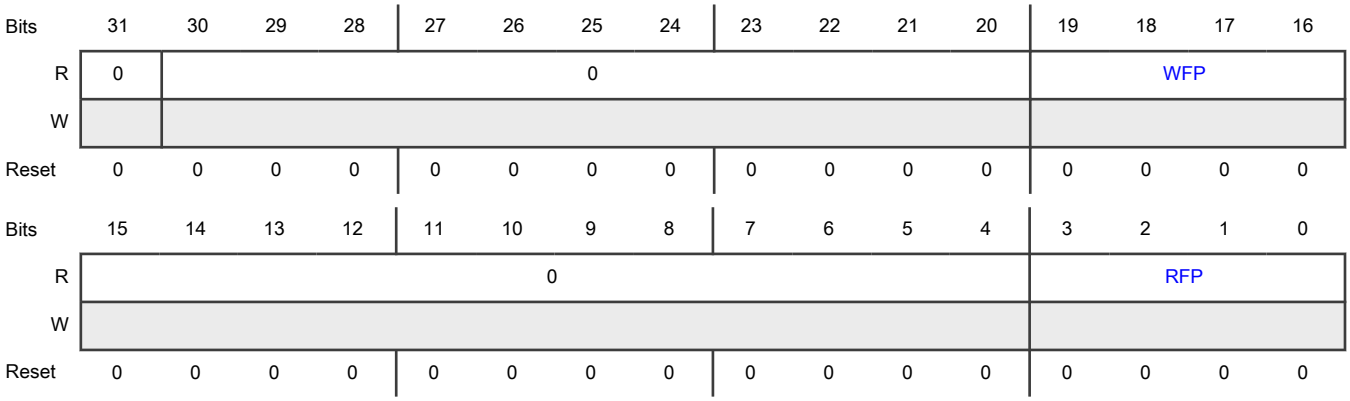
Offset

Register	Offset
TFR0	40h

Function

Contains the transmit FIFO pointers. The MSB of the read and write pointers distinguishes between FIFO full and empty conditions. If the read and write pointers are identical, the FIFO is empty. If the read and write pointers are identical except for the MSB, the FIFO is full.

Diagram



Fields

Field	Function
31 —	Reserved
30-20 —	Reserved
19-16 WFP	Write FIFO Pointer Indicates the FIFO write pointer for the transmit data channel.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-4 —	Reserved
3-0 RFP	Read FIFO Pointer Indicates the FIFO read pointer for the transmit data channel.

58.6.1.12 Transmit Mask (TMR)

Offset

Register	Offset
TMR	60h

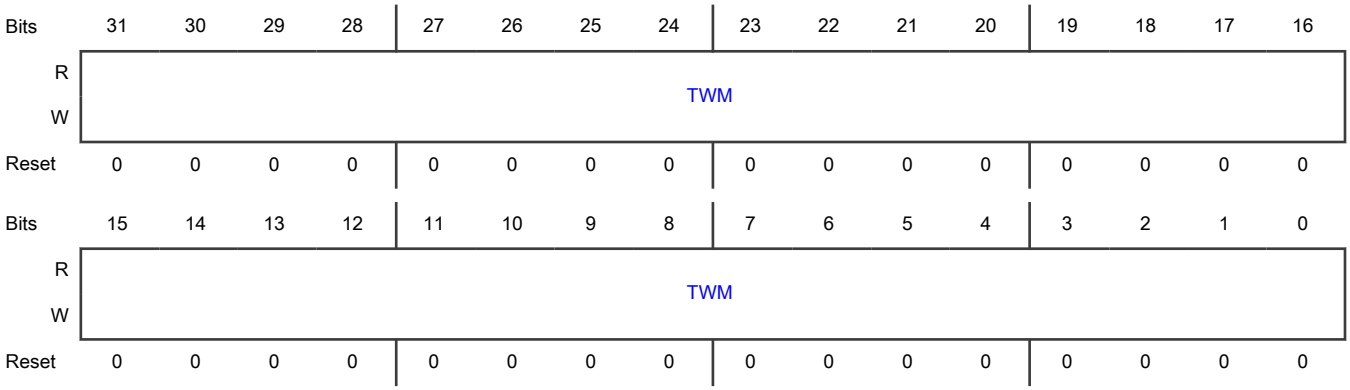
Function

Contains the mask for the transmit word. This register is double-buffered and updates:

- When TCSR[TE] first becomes 1.
- At the end of each frame.

This setup allows the masked words in each frame to change from frame to frame.

Diagram



Fields

Field	Function
31-0 TWM	Transmit Word Mask Configures whether the transmit word *n* is masked for the corresponding word in the frame. In this case, being masked means that transmit data pins are 3-stated or driven zero and transmit data is not read from the FIFO.

Table continues on the next page...

Field	Function
	0000_0000_0000_0000_0000_0000_0000_0000b - Enable 0000_0000_0000_0000_0000_0000_0000_0001b - Mask

58.6.1.13 Transmit Timestamp Control (TTCR)

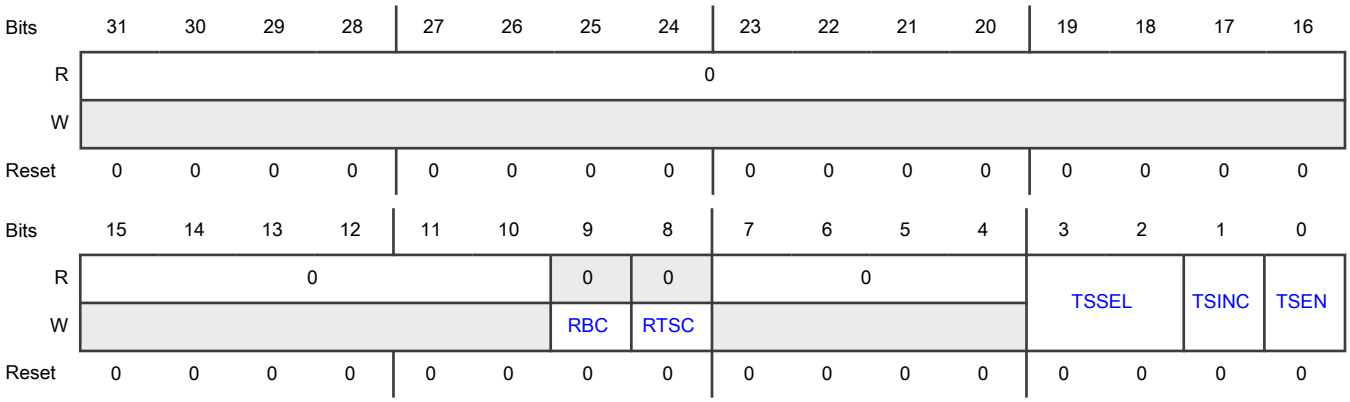
Offset

Register	Offset
TTCR	70h

Function

Controls the transmit timestamp counter and bit clock counter.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RBC	Reset Bit Counter Resets the bit counter. 0b - No effect 1b - Reset
8 RTSC	Reset Timestamp Counter Resets the timestamp counter. 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Reset
7-4 —	Reserved
3-2 TSSEL	Timestamp Select Configures whether the timestamp counter is synchronized with another timestamp counter. 00b - Increment when enabled 01b - Increment when the receive timestamp counter is enabled 10b - Increment when the transmit timestamp counter on another instance is enabled 11b - Increment when the receive timestamp counter on another instance is enabled
1 TSINC	Timestamp Increment Configures when the timestamp counter starts incrementing. 0b - When enabled and after the bit counter has incremented 1b - When enabled
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0b - Disable 1b - Enable

58.6.1.14 Transmit Timestamp (TTSR)

Offset

Register	Offset
TTSR	74h

Function

Indicates the current value of the transmit timestamp counter.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TSC	Timestamp Counter Indicates the value of the timestamp counter.

58.6.1.15 Transmit Bit Count (TBCR)

Offset

Register	Offset
TBCR	78h

Function

Indicates the current value of the transmit bit counter. The bit counter increments for each bit in a frame when the SAI transmitter is enabled. Reading this register causes Transmit Bit Count Timestamp (TBCTR) to update with the value of the timestamp counter that corresponds to the bit count register value.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 BCNT	Bit Counter Indicates the current value of the bit counter.

58.6.1.16 Transmit Bit Count Timestamp (TBCTR)

Offset

Register	Offset
TBCTR	7Ch

Function
Indicates the value of the timestamp register that corresponds to the last value read from Transmit Bit Count (TBCR).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 BCTS	Bit Timestamp Indicates the timestamp counter value that matches the last read from Transmit Bit Count (TBCR).

58.6.1.17 Receive Control (RCSR)

Offset

Register	Offset
RCSR	88h

Function

Contains receiver enable fields including resets, error and interrupt enable fields, and error flag fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 RE	Receiver Enable Enables the receiver. When you write 0 to this field, the receiver remains enabled and the field remains 1 until the end of the current frame. 0b - Disable 1b - Enable (or receiver disabled and not yet reached end of frame)
30 STOPE	Stop Enable Enables receiver operation in Stop mode. 0b - Disable 1b - Enable
29 DBGE	Debug Enable Enables receiver operation in Debug mode, which does not affect the receive bit clock. 0b - Disable after completing the current frame 1b - Enable
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RCSR[RE]. This field becomes 1 automatically when RCSR[RE] becomes 1. When you write 0 to this field, the receive bit clock remains enabled, and this field remains 1 until the end of the current frame. 0b - Disable 1b - Enable
27-26 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which can be zero. Reading this field always returns zero. **NOTE** The FIFO reset is asserted for one cycle only. You must reset FIFO pointers only when the receiver is disabled or the FIFO error flag is 1. 0b - No effect 1b - Reset
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. **NOTE** The software reset remains asserted until you clear it. 0b - No effect 1b - Software reset
23-21 —	Reserved
20 WSF	Word Start Flag Indicates whether SAI has detected the start of the configured word. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
19 SEF	Sync Error Flag Indicates whether SAI has detected an error in the externally generated frame sync. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
18 FEF	FIFO Error Flag Indicates whether an enabled receive FIFO has overflowed. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Receive overflow detected When writing 0b - No effect 1b - Clear the flag
17 FWF	FIFO Warning Flag Indicates whether an enabled receive FIFO is full. 0b - Not full 1b - Full
16 FRF	FIFO Request Flag Indicates whether the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0b - Watermark not reached 1b - Watermark reached
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables word start interrupts. 0b - Disable 1b - Enable
11 SEIE	Sync Error Interrupt Enable Enables sync error interrupts.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
10 FEIE	FIFO Error Interrupt Enable Enables FIFO error interrupts. 0b - Disable 1b - Enable
9 FWIE	FIFO Warning Interrupt Enable Enables FIFO warning interrupts. 0b - Disable 1b - Enable
8 FRIE	FIFO Request Interrupt Enable Enables FIFO request interrupts. 0b - Disable 1b - Enable
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables DMA warnings. 0b - Disable 1b - Enable
0 FRDE	FIFO Request DMA Enable Enables DMA requests. 0b - Disable 1b - Enable

58.6.1.18 Receive Configuration 1 (RCR1)

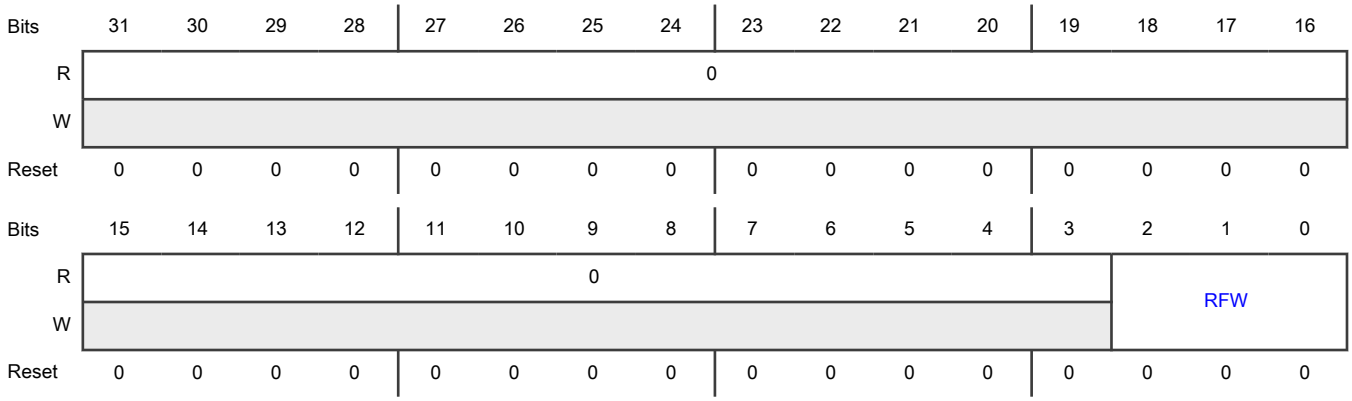
Offset

Register	Offset
RCR1	8Ch

Function

Configures the watermark level for all enabled receiver channels.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 RFW	Receive FIFO Watermark Configures the level of the receive FIFO watermark, in 32-bit FIFO words. 000b - 1 001b - 2 010b-110b - (RFW value + 1) 111b - 8

58.6.1.19 Receive Configuration 2 (RCR2)

Offset

Register	Offset
RCR2	90h

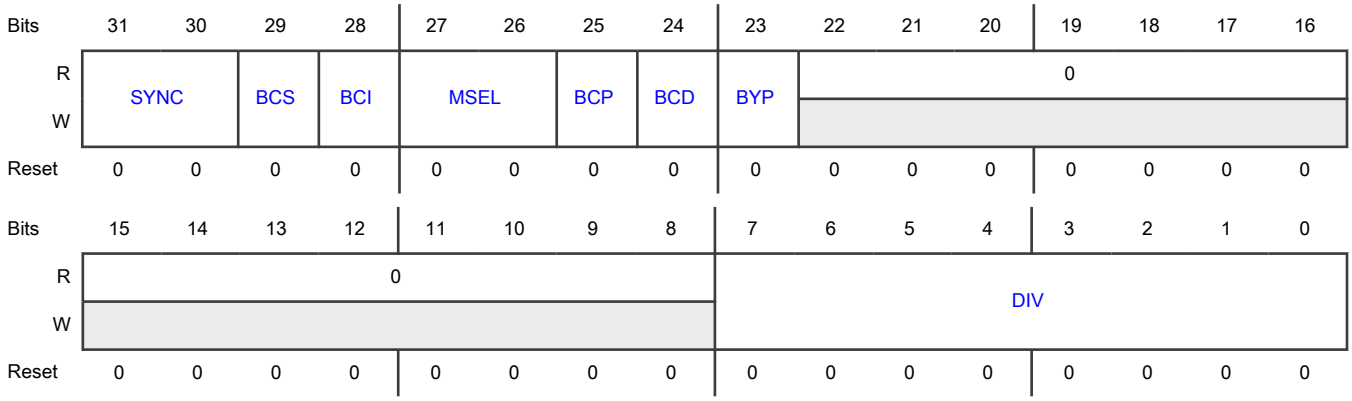
Function

Contains the SYNC mode and clock setting fields.

NOTE

Do not alter this register when RCSR[RE] is 1.

Diagram



Fields

Field	Function
31-30 SYNC	Synchronous Mode Configures Asynchronous and Synchronous modes of operation. When configured for a Synchronous mode of operation, you must configure the transmitter for asynchronous operation. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI0	RCR2[30]	RCR2[31]
SAI1	RCR2	—
SAI2	RCR2	—
SAI3	RCR2[30]	RCR2[31]

NOTE
The descriptions of the field settings vary by module instance.

Instance	Field value and description
SAI0 SAI3	0b - Asynchronous mode 1b - Synchronous with transmitter
SAI1 SAI2	00b - Asynchronous mode 01b - Synchronous with transmitter 10b - Synchronous with another SAI receiver 11b - Synchronous with another SAI transmitter

Table continued from the previous page...

Field	Function
29 BCS	Bit Clock Swap Swaps the bit clock used by the receiver. When the receiver is configured in Asynchronous mode and this field is 1, the transmitter bit clock (TX_BCLK) clocks the receiver. This setting allows the transmitter and receiver to share one bit clock, while the receiver continues to use the receiver frame sync (RX_SYNC). When you configure the receiver in Synchronous mode, you must write the same value to the transmitter BCS field and receiver BCS field. When both are 1, the transmitter and receiver are both clocked by the receiver bit clock (RX_BCLK) but both use the transmitter frame sync (TX_SYNC). 0b - Use the normal bit clock source 1b - Swap the bit clock source
28 BCI	Bit Clock Input Enables the internal logic to be clocked as if the bit clock is generated externally. When this field is 1 and when using an internally generated bit clock in Synchronous or Asynchronous mode, the bit clock used by the receiver is delayed by the pad output delay. (The pad input clocks the receiver as if the clock is externally generated.) This setting decreases the data input setup time, but increases the data output valid time. Use the Target mode timing from the data sheet for the receiver when this field is 1. In Synchronous mode, this field allows the receiver to use the Target mode timing from the data sheet, while the transmitter uses the Controller mode timing. This field has no effect when configured for an externally generated bit clock. **NOTE** When this field is 1, both the input buffer and output buffer must be enabled for the BCLK pad. 0b - Disable 1b - Enable
27-26 MSEL	MCLK Select Selects the audio clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE** Some audio clock options are not available for some chips. See the chip-specific information for the availability and chip-specific meaning of each option. 00b - Bus clock 01b - Audio clock (MCLK) option 1 10b - Audio clock (MCLK) option 2 11b - Audio clock (MCLK) option 3
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. If you write 0 to this field, the bit clock becomes active high with drive outputs on the rising edge and sample inputs on the falling edge. If you write 1 to this field, the bit clock becomes active low with drive outputs on the falling edge and sample inputs on the rising edge.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Active high 1b - Active low
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0b - Generated externally in Target mode 1b - Generated internally in Controller mode
23 BYP	Bit Clock Bypass Enables the bypass of the bit clock divider. When enabled, the internal bit clock is divide-by-one of the audio clock. When disabled, the internal bit clock is generated from the bit clock divider. 0b - Disable 1b - Enable
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Determines the value by which the audio clock is divided to generate the bit clock, when configured for an internal bit clock. The division value is (DIV + 1) × 2.

58.6.1.20 Receive Configuration 3 (RCR3)

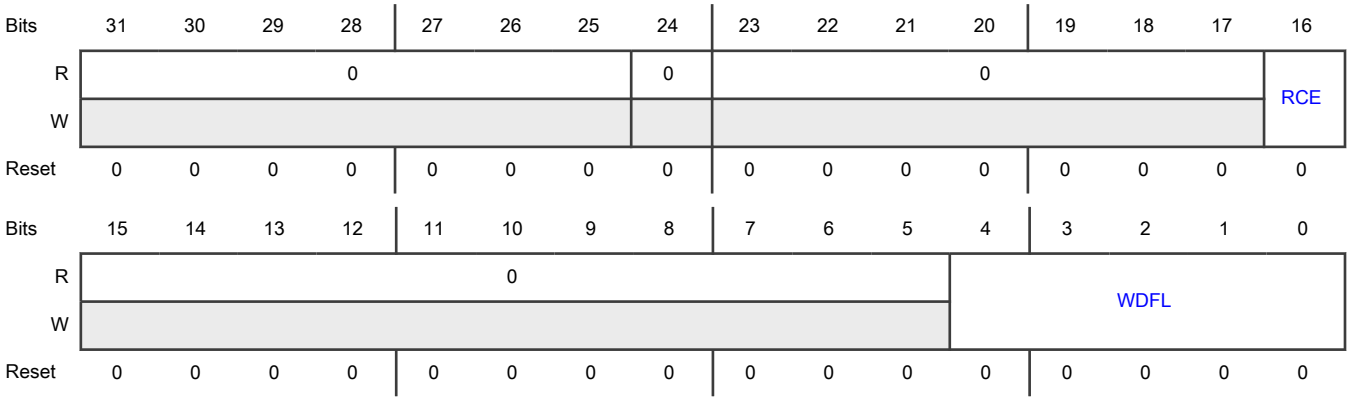
Offset

Register	Offset
RCR3	94h

Function

Contains the receive channel settings.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-17 —	Reserved
16 RCE	Receive Channel Enable Enables the corresponding data channel for receive operation. Changing this field takes effect immediately for generating the FIFO request and warning flags. It takes effect at the end of each frame for receive operations. The width of RCE = the number of receive channels (call it *n*). For example, if RCE is two bits wide, then bit position 16 refers to receive channel 1 and bit position 17 refers to receive channel 2. Writing 1 to bit 16 enables receive channel 1, and writing 1 to bit 17 enables receive channel 2. Writing 1 to bit *n* enables receive channel *n*. **NOTE** When there is only a single channel, you do not need an individual channel FIFO reset. In that case, use the global FIFO reset (RCSR[FR]). • 0b – Disable receive data channel *n* • 1b – Enable receive data channel *n*
15-5 —	Reserved
4-0 WDFL	Word Flag Configuration

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Configures which word sets the word start flag (RCSR[WSF]). The value must be one less than the word number (for example, write zero to select the first word in the frame). If you configure this field to a value greater than RCR4[FRSZ], RCSR[WSF] is never set. 0_0000b - Word 1 0_0001b - Word 2 0_0010b-1_1110b - Word (WDFL value + 1) 1_1111b - Word 32

58.6.1.21 Receive Configuration 4 (RCR4)

Offset

Register	Offset
RCR4	98h

Function

Contains the receive fields for FIFO Combine mode, FIFO Packing mode, and frame sync settings.

NOTE

Do not alter this register when RCSR[RE] is 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FCONT	0		FPACK		0				FRSZ			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0		Reserved	MF	FSE	ONDEM	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
28 FCONT	FIFO Continue on Error Configures when SAI continues receiving data after a FIFO error is detected. 0b - From the start of the next frame after the FIFO error flag is cleared 1b - From the same word that caused the FIFO error to become 1 after the FIFO warning flag is cleared
27-26 —	Reserved
25-24 FPACK	FIFO Packing Mode Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8 bits or 16 bits, only the first 8 bits or 16 bits are stored in the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO read pointer only increments when software reads the full 32-bit FIFO word. 00b - Disable 01b - Reserved 10b - Enable 8-bit FIFO packing 11b - Enable 16-bit FIFO packing
23-21 —	Reserved
20-16 FRSZ	Frame Size Configures the number of words in each frame. The value must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words. 0_0000b - 1 0_0001b - 2 0_0010b-1_1110b - (FRSZ value + 1) 1_1111b - 32
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit-clock cycles. The value must be one less than the number of bit-clock cycles. For example, write 0 for the frame sync to assert for one bit-clock cycle only. You cannot configure this field to be longer than the first word of the frame. 0_0000b - 1 0_0001b - 2

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0_0010b-1_1110b - (SYWD value + 1) 1_1111b - 32
7-6 —	Reserved
5 —	Reserved Write only zero to this field.
4 MF	MSB First Configures what is received first: LSB or MSB. 0b - LSB 1b - MSB
3 FSE	Frame Sync Early Determines when frame sync is asserted. 0b - First bit of the frame 1b - One bit before the first bit of the frame
2 ONDEM	On-Demand Mode Determines when the internal frame sync is generated. When this field is 1, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is 0. 0b - Generated continuously 1b - Generated when the FIFO warning flag is 0
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0b - Active high 1b - Active low
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0b - Generated externally in Target mode 1b - Generated internally in Controller mode

58.6.1.22 Receive Configuration 5 (RCR5)

Offset

Register	Offset
RCR5	9Ch

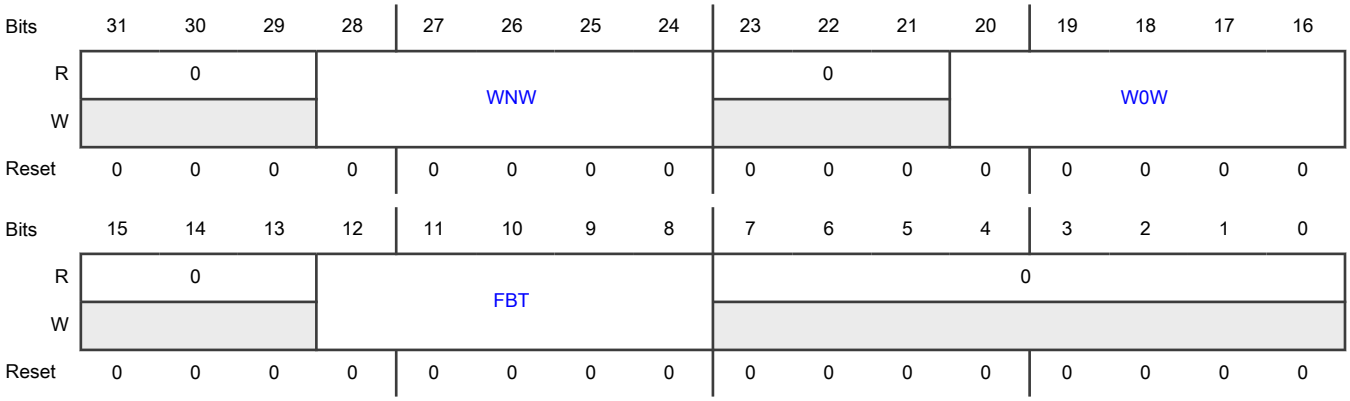
Function

Contains receive word and bit index settings.

NOTE

Do not alter this register when RCSR[RE] is 1.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word except for the first one in the frame. The value must be one less than the number of bits per word. SAI does not support word widths less than 8 bits. SAI does not support bit settings 0–6 (0b–0_0110b). 0_0111b - 8 0_1000b - 9 0_1001b-1_1110b - (WNW value + 1) 1_1111b - 32
23-21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
20-16 W0W	Word 0 Width Configures the number of bits in the first word of each frame. The value must be one less than the number of bits in the first word. SAI does not support word widths less than 8 bits when there is only one word per frame. 0_0000b - 1 0_0001b - 2 0_0010b-1_1110b - (W0W value + 1) 1_1111b - 32
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB-first, the index of the next bit received is one less than the current bit received. If configured for LSB-first, the index of the next bit received is one more than the current bit received. The value must be greater than or equal to the word width when configured for MSB-first. The value must be less than or equal to 31-word width when configured for LSB-first. See Data alignment for details. 0_0000b - 0 0_0001b-1_1110b - FBT value 1_1111b - 31
7-0 —	Reserved

58.6.1.23 Receive Data (RDR0)

Offset

Register	Offset
RDR0	A0h

Function

Contains receive data.

When the receive FIFO is not empty, reads from this register return the data from the top of the receive FIFO. When the receive FIFO is empty, SAI ignores reads from this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 RDR	Receive Data

58.6.1.24 Receive FIFO (RFR0)

Offset

Register	Offset
RFR0	C0h

Function

Contains the receive FIFO pointers. The MSB of the read and write pointers distinguishes between FIFO full and empty conditions. If the read and write pointers are identical, the FIFO is empty. If the read and write pointers are identical except for the MSB, the FIFO is full.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												WFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0											RFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-16 WFP	Write FIFO Pointer Indicates the FIFO write pointer for the receive data channel.
15 —	Reserved
14-4 —	Reserved
3-0 RFP	Read FIFO Pointer Indicates the FIFO read pointer for the receive data channel.

58.6.1.25 Receive Mask (RMR)

Offset

Register	Offset
RMR	E0h

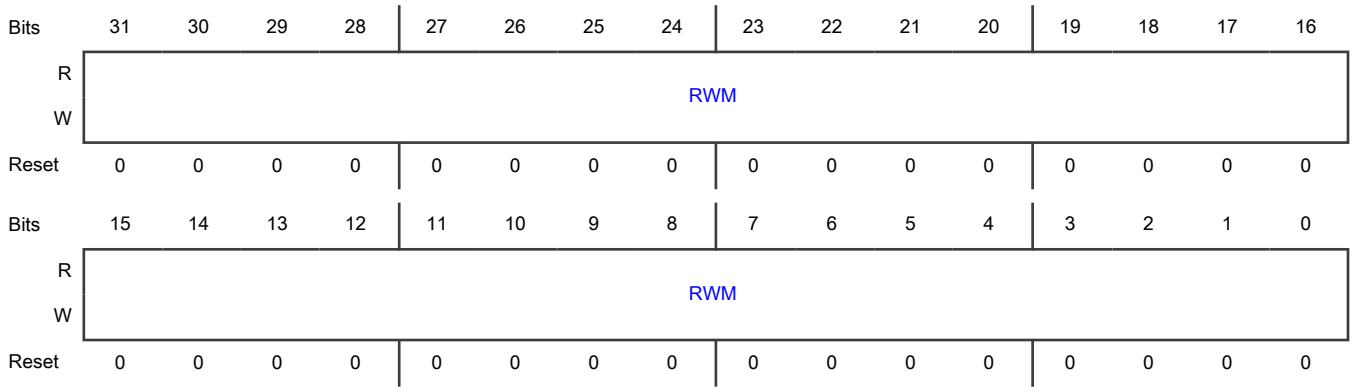
Function

Contains the mask for the receive word. This register is double-buffered and updates:

- When RCSR[RE] first becomes 1.
- At the end of each frame.

This setup allows the masked words in each frame to change from frame to frame.

Diagram



Fields

Field	Function
31-0 RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 0000_0000_0000_0000_0000_0000_0000_0000b - Enable 0000_0000_0000_0000_0000_0000_0000_0001b - Mask

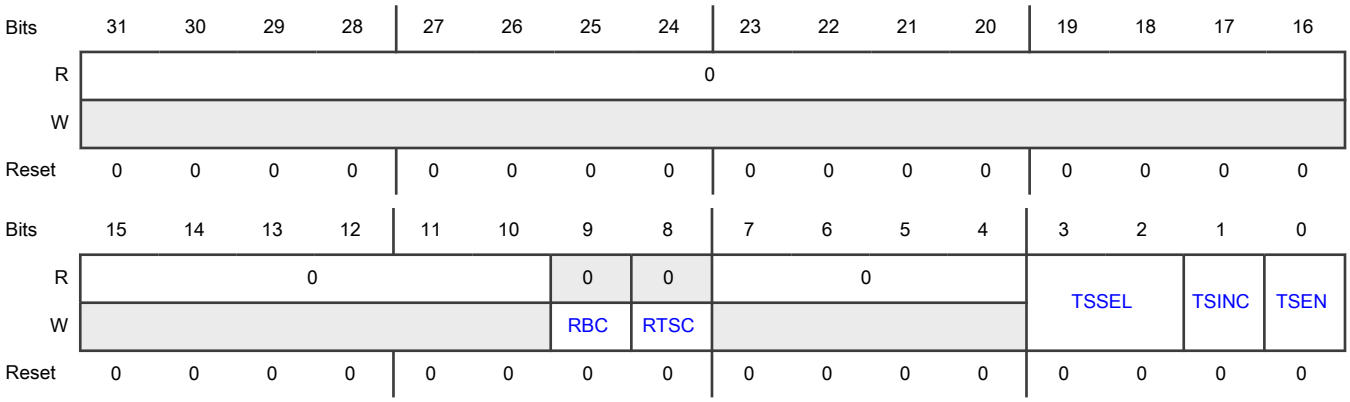
58.6.1.26 Receive Timestamp Control (RTCR)

Offset

Register	Offset
RTCR	F0h

Function
Controls the receive timestamp counter and bit clock counter.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RBC	Reset Bit Counter Resets the receive bit counter. 0b - No effect 1b - Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 RTSC	Reset Timestamp Counter Resets the receive timestamp counter. 0b - No effect 1b - Reset
7-4 —	Reserved
3-2 TSSEL	Timestamp Select Configures whether the timestamp counter is synchronized with another timestamp counter. 00b - Increment when enabled 01b - Increment when the transmit timestamp counter is enabled 10b - Increment when the receive timestamp counter on another instance is enabled 11b - Increment when the transmit timestamp counter on another instance is enabled
1 TSINC	Timestamp Increment Configures when the timestamp counter starts incrementing. 0b - When enabled and after the bit counter has incremented 1b - When enabled
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0b - Disable 1b - Enable

58.6.1.27 Receive Timestamp (RTSR)

Offset

Register	Offset
RTSR	F4h

Function
Indicates the current value of the receive timestamp counter.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TSC	Timestamp Counter Indicates the current value of the timestamp counter.

58.6.1.28 Receive Bit Count (RBCR)

Offset

Register	Offset
RBCR	F8h

Function

Indicates the current value of the receive bit counter. The bit counter increments for each bit in a frame when the SAI receiver is enabled. Reading this register causes Receive Bit Count Timestamp (RBCTR) to update with the value of the timestamp counter that corresponds to the bit count register value.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 BCNT	Bit Counter Indicates the current value of the bit counter.

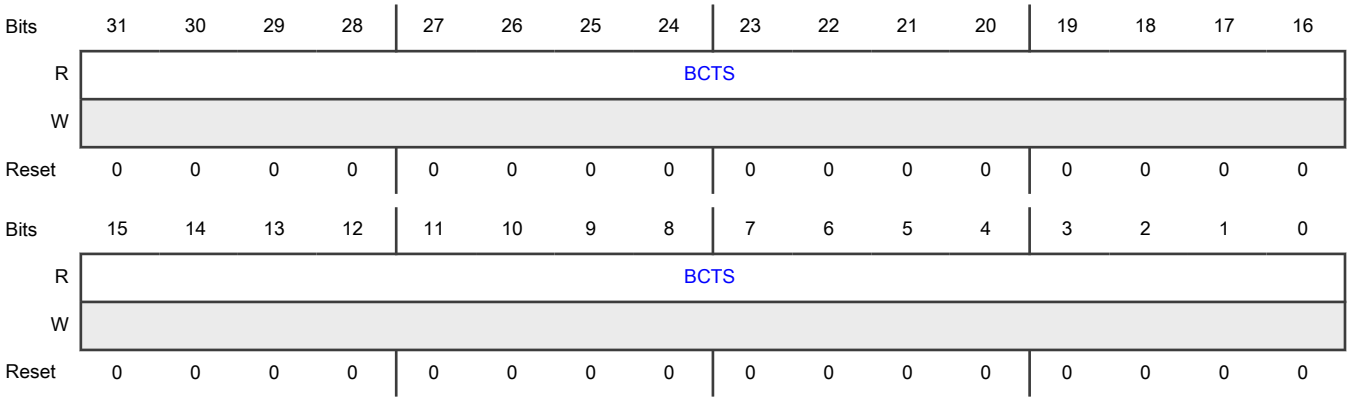
58.6.1.29 Receive Bit Count Timestamp (RBCTR)

Offset

Register	Offset
RBCTR	FCh

Function
Indicates the value of the timestamp register that corresponds to the value last read from Receive Bit Count (RBCR).

Diagram



Fields

Field	Function
31-0 BCTS	Bit Timestamp Indicates the timestamp counter value that matches the last value read from Receive Bit Count (RBCR).

58.6.1.30 MCLK Control (MCR)

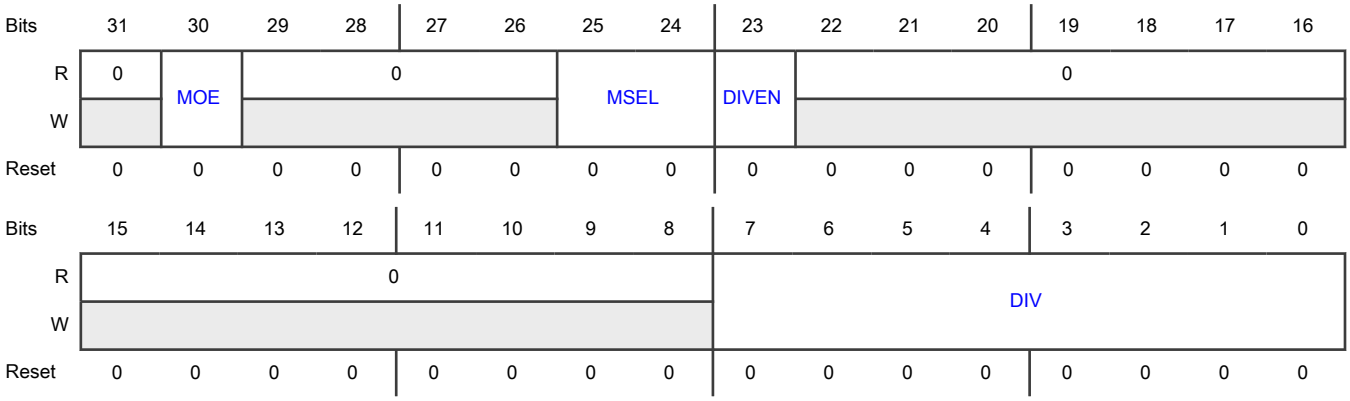
Offset

Register	Offset
MCR	100h

Function

Controls the audio clock.

Diagram



Fields

Field	Function
31 —	Reserved
30 MOE	MCLK Output Enable Configures the MCLK signal pin as an input or output. 0b - Input 1b - Output
29-26 —	Reserved
25-24 MSEL	MCLK Select Selects the clock used by the MCLK postdivider. You cannot change this field when the MCLK postdivider is enabled. See the chip-specific information for the connections to these inputs. 00b - Audio clock (MCLK) option 1 01b - Reserved 10b - Audio clock (MCLK) option 2 11b - Audio clock (MCLK) option 3
23 DIVEN	MCLK Post Divide Enable Enables the postdivider that is output on the MCLK signal pin. When enabled, the output on the MCLK signal pin is a postdivided version of the audio clock. When disabled, the output is the audio clock. The output divider does not impact the audio clock used by the transmitter or receiver. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
22-8 —	Reserved
7-0 DIV	MCLK Post Divide Specifies the postdivide value for the audio clock that only divides the output to the MCLK signal pin. The division value is (DIV + 1) × 2.

Chapter 59
HiFi 1 DSP Platform (HiFi1)

59.1 Chip-specific HiFi1 information

Table 577. Reference links to related information

Topic	Related module	Reference
Full description	HiFi1	HiFi1
System memory map		Memory map overview
Audio overview		Audio overview
Power management		Power overview PMC
Clocking		Clock overview Clock controller
HiFi1 IRQ assignments		Interrupt and DMA events assignments
System Controller	SYSCON	SYSCON

59.1.1 Module instances

This device has one instance of the HiFi1 module.

Table 578. HiFi1 instance

	HiFi1
Located in	VDD1_SENSE

59.1.2 SYSCON controls on HiFi1

Table 579. SYSCON controls on HiFi1

Control	SYSCON register
Start HiFi1 running	SYSCON1.DSPSTALL[DSPSTALL]
Force HiFi1 to enter OCD halt mode at reset	SYSCON1.OCDHALTONRESET[OCDHALTONRESET]
Indicate double exception error	SYSCON1.HIFI1_GPR0[EXCEPTIONERR]
Indicate fatal error	SYSCON1.HIFI1_GPR0[PFATALERR]
Indicate PFaultInfo received	SYSCON1.HIFI1_GPR0[PFAULTINFOVLD]
Indicate fault information	SYSCON1.HIFI1_GPR1[PFAULTINFO]
Indicate GPIO32 option TIE output state	SYSCON1.HIFI1_GPR2[EXPSTATE]
Configure HiFi1 static vector remapping to shared SRAMs.	SYSCON1.DSP_VECT_REMAP

59.1.3 Initialization

Table 580. HiFi1 initialization procedure

No.	Action	Programming
1	Reset the HiFi1	If the chip is not in debug mode • Write 1b to RSTCTL1.PRSTCTL0_SET[HiFi1] If the chip is in debug mode • Write 1b to RSTCTL1.PRSTCTL0_SET[HiFi1_DEBUG]
2	Configure the HiFi1 to Stall mode.	Write 1b to SYSCON1.DSPSTALL[DSPSTALL]
3	Disable the HiFi1 reset	If the chip is not in debug mode • Write 1b to RSTCTL1.PRSTCTL0_CLR[HiFi1] If the chip is in debug mode • Write 1b to RSTCTL1.PRSTCTL0_CLR[HiFi1_DEBUG]
4	Enable clock.	
4.1	Enable Sense domain clock (if not enabled by default).	Write 1b to SLEEPCON0.RUNCFG_CLR[SENSEP_MAINCLK_SHUTOFF].
4.2	Select HiFi1 clock source.	Configure CLKCTL1.SENSEDSPCPUCLKSEL[SEL] .
4.3	Enable HiFi1 clock mux output.	Write 1b to CLKCTL1.SENSEDSPCPUCLKSEL[SEL_EN].
4.4	Enable HiFi1 access SRAM0-17 clock (if needed).	Write 1b to CLKCTL1.PSCCTL1[SENSE_ACCESS_RAM_ARBITER0].
4.5	Enable SYSCON1 clock (if not enabled by default).	Write 1b to CLKCTL1.PSCCTL0[SYSCON1].
4.6	Enable SLEEPCON1 clock (if not enabled by default).	Write 1b to CLKCTL1.PSCCTL0[SLEEPCON1].
4.7	Enable HiFi1 clock.	Write 1b to CLKCTL1.PSCCTL1[HiFi1].
5	Initialize the memory with instructions and data.	
6	Configure the HiFi1 to Run mode	Write 0b to SYSCON1.DSPSTALL[DSPSTALL]

59.2 Overview

Cadence® Xtensa® HiFi 1 Audio DSP (HiFi1) runs at frequencies up to 250 MHz. It is a 2-slot machine with two-way single-instruction multiple-data (SIMD) arithmetic logic unit (ALU) or multiplier–accumulator (MAC) operations. All HiFi1 operations can be used as intrinsics in low-level applications.

59.2.1 Block diagram

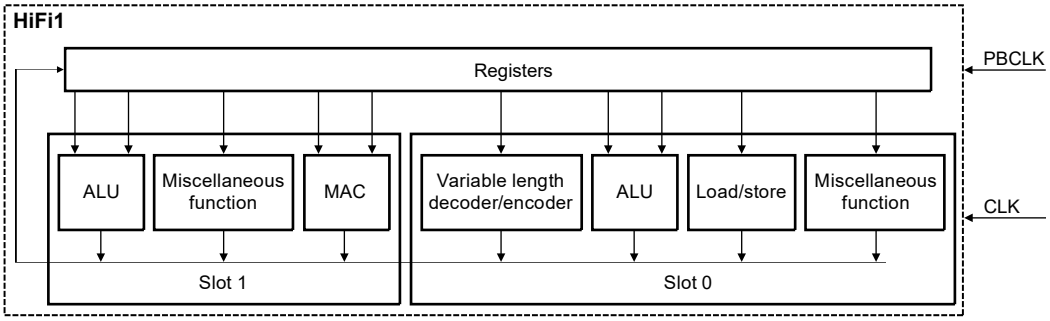


Figure 254. HiFi1 block diagram

59.2.2 Features

HiFi1 supports the following features:

- Vector floating-point unit (VFPU)
- Low-latency multiplier–accumulator (MAC)
- 256 bytes L0 loop buffer
- Memory protection: Region protection
- 32-bit integer divider
- Take exception on unaligned load and store address
- Normal L32R hardware support
- 5-stage pipeline
- 1 load and store unit

59.3 Functional description

59.3.1 Arithmetic logic unit (ALU) and multiplier–accumulator (MAC)

In HiFi1 DSP, MAC is a common process that computes the product of two numbers and adds the results to an accumulator. ALU performs arithmetic and logical operations on the operands. HiFi1 supports quad 16 x 16-bit, dual 32 x 16-bit, and single 32-bit MACs. 24 x 24-bit operations are emulated using 32 x 32 operations. It also supports CLAMPS operations and 32-bit integer divider.

59.3.2 Variable-length decoder and encoder

The HiFi1 bitstream engine supports both fixed-length and variable-length decoding and encoding. Variable-length (Huffman) decode instructions are specialized instructions, in which the elements with variable bit-widths are decoded.

59.3.3 Load and store operations

HiFi1 supports loading and storing scalars or vectors of 8, 16, 32, and 64 bits. Each scalar of load and store accesses 8, 16, 32, or 64 bits. Each vector accesses 64 bits.

59.3.4 Low-latency MAC

The low-latency MAC or FMA function reduces the latency fused multiply add (FMA) hardware and fixed-point MAC hardware to 1 cycle, and therefore improving MAC and FMA performance respectively.

The latencies of the below multi-cycle operations are reduced:

- Integer and fixed-point MAC: from 2 to 1 cycle
- Single-precision floating-point FMA: from 3 to 1 cycle
- Single-precision floating-point conversions: from 2 to 1 cycle

59.3.5 L0 loop buffer

L0 loop buffer captures instructions from the body of a Zero Overhead Loop, and then executes subsequent iterations of the loop from this buffer. This allows the instruction memories to be disabled during most of the loop execution. Power savings depend on what portion of code is being executed as Zero Overhead Loops.

59.3.6 Miscellaneous function

Miscellaneous functions in HiFi1 include:

- NSA/NSAU: Returns the number of contiguous sign bits in the most significant bits of a 32-bit signed/unsigned value
- MinMax: Performs a signed or unsigned integral minimum or maximum operation
- SEXT: Sign extend to 32 bits
- DEPBITS: Inferred automatically by the compiler from standard C code
- Density: Enable use of density (16-bit) instructions

59.3.7 PFaultInfo and Fault Information Register (FIR)

FIR provides information on the error condition and the PFaultInfo output signal mirrors this internal register.

Bit	Description	Notes
31:21	Deasserted	
20	Multiple Faults 0b - Only one fault has occurred so far 1b - Multiple faults have occurred*	* In this case, the FIR register has information only on the most severe fault.
19:16	User Bits These bits are zero by default. Software running on Xtensa can write to the user bits.	These bits are available only in configurations with both debug features (i.e. OCD at a minimum) and parity/ECC protection on the AXI bus interface.
15**:8	Fault Severity Level 0000b - Fault severity is not reset. The system can recover without resetting 0001b - Fault severity is core reset. Minimally, reset the Xtensa processor to recover from the fault 0010b - Fault severity is system reset. The whole system may need to be reset to recover from the fault	** CSR parity error sets the PFaultInfo[15] output signal. **NOTE** The suggested severity level and the recovery mechanism are recommendations and may not always work.
7:0	AXI Channel and AXI Port AXI channel [7:4]:	

Table continued from the previous page...

Bit	Description	Notes
	0000b - Read address AXI channel 0001b - Write address AXI channel 0010b - Write data AXI channel 0011b - Read data AXI channel 0100b - Write response AXI channel AXI port [3:0]: 0000b - Main AXI initiator port 0001b - Secondary (iDMA) AXI initiator port 0010b - AXI target port	

59.3.8 Clocking

Clock	Description
CLK	Main clock
PBCLK	APB interface clock

59.3.9 Interrupts

HiFi1 provides 32 interrupts with 4 priority levels, 2 timers, and 2 EXCM (exception mode) priority levels with highest priority of efficiently callable handlers.

59.4 Signals

Signal name	Description
OCDHaltOnReset	On-Chip Debug (OCD) halt on reset Tied low for normal operation. Alternatively, this signal can be tied high to force the processor to enable OCD and enter Stopped (halt) state at processor reset.
XOCDMode	Xtensa On-Chip Debug (XOCD) mode Indicates that the core is in Stopped (halt) or Stepping state.
DoubleExceptionError	Double exception error Indicates a double exception fault occurs.
PFatalError	Processor fatal error Indicates a fatal error condition occurs (e.g., CSR parity error).
PFaultInfo	Processor fault information PFaultInfo[31:0] mirrors the processor's internal FIR, using the same bit encoding to provide the source and severity of the fault. See PFaultInfo and Fault Information Register (FIR).
PFaultInfoValid	PFaultInfo valid

Table continues on the next page...

Table continued from the previous page...

Signal name	Description
	Asserted when PFaultInfo signal changes its value. The system can use its high level as an enabler to capture PFaultInfo.
TIE_EXPSTATE	Tensilica Instruction Extension (TIE) output state 32-bit set by instructions which can be connected to any system peripherals, independent of main system bus.

Chapter 60
HiFi 4 DSP Platform (HiFi4)

60.1 Chip-specific HiFi4 information

Table 581. Reference links to related information

Topic	Related module	Reference
Full description	HiFi4	HiFi4
System memory map		Memory map overview
Audio overview		Audio overview
Power management		Power overview PMC
Clocking		Clock overview Clock controller
HiFi4 IRQ assignments		Interrupt and DMA events assignments
System Controller	SYSCON	SYSCON

60.1.1 Module instances

This device has one instance of the HiFi4 module.

Table 582. HiFi4 instance

	HiFi4
Located in	VDD2_COMP

60.1.2 SYSCON controls on HiFi4

Table 583. SYSCON controls on HiFi4

Control	SYSCON register
Start HiFi4 running	SYSCON0.DSPSTALL[DSPSTALL]
Force HiFi4 to enter OCD halt mode at reset	SYSCON0.OCDHALTONRESET[OCDHALTONRESET]
Indicate double exception error	SYSCON0.HIFI4_GPR0[EXCEPTIONERR]
Indicate fatal error	SYSCON0.HIFI4_GPR0[PFATALERR]
Indicate PFaultInfo received	SYSCON0.HIFI4_GPR0[PFAULTINFOVLD]
Indicate fault information	SYSCON0.HIFI4_GPR1[PFAULTINFO]
Indicate GPIO32 option TIE output state	SYSCON0.HIFI4_GPR2[TIE_EXPSTATE]
Configure HiFi4 remap function	SYSCON0.DSP_VECT_REMAP

60.1.3 Initialization

Table 584. HiFi4 initialization procedure

No.	Action	Programming
1	Reset the HiFi4	If the chip is not in debug mode • Write 1b to RSTCTL0.PRSTCTL5_SET[HiFi4] If the chip is in debug mode • Write 1b to RSTCTL0.PRSTCTL5_SET[HiFi4_DEBUG]
2	Configure the HiFi4 to Stall mode.	Write 1b to SYSCON0.DSPSTALL[DSPSTALL]
3	Configure PMC.	
3.1	Power on HiFi4	Write 0b to PMC0.PDRUNCFG0[V2DSP_PD]
3.2	Power on HiFi4 TCM and cache RAMs array	Write 0000b to PMC0.PDRUNCFG4[12:15]
3.3	Power on HiFi4 TCM and cache RAMs periphery	Write 0000b to PMC0.PDRUNCFG5[12:15].
3.4	Apply PMC configuration changes for HiFi4	Write 1b to PMC0.CTRL[APPLYCFG].
3.5	Confirm the PMC configuration changes have completed	Read PMC0.STATUS[BUSY]. if it is 0, the changes have completed, or else, wait and read it again.
4	Disable the HiFi4 reset	If the chip is not in debug mode • Write 1b to RSTCTL0.PRSTCTL5_CLR[HiFi4] If the chip is in debug mode • Write 1b to RSTCTL0.PRSTCTL5_CLR[HiFi4_DEBUG]
5	Enable clock.	
5.1	Select HiFi4 clock source.	Configure CLKCTL0.DSPCPUCLKSEL[SEL] .
5.2	Enable HiFi4 clock mux output.	Write 1b to CLKCTL0.DSPCPUCLKSEL[SEL_EN].
5.3	Enable HiFi4 access SRAM18-29 clock (if needed).	Write 1b to CLKCTL0.PSCCTL5[HiFi4_ACCESS_RAM_ARBITER1].
5.4	Enable SYSCON0 clock (if not enabled by default).	Write 1b to CLKCTL0.PSCCTL0[SYSCON0].
5.5	Enable SLEEPCON0 clock (if not enabled by default).	Write 1b to CLKCTL0.PSCCTL0[SLEEPCON0].
5.6	Enable HiFi4 clock.	Write 1b to CLKCTL0.PSCCTL4[HiFi4].
6	Initialize the memory with instructions and data.	
7	Configure the HiFi4 to Run mode	Write 0b to SYSCON0.DSPSTALL[DSPSTALL]

60.2 Overview

Cadence® Xtensa® HiFi 4 Audio DSP (HiFi4) is a highly optimized audio processor running at frequencies up to 325 MHz, for efficient execution of audio and voice codecs, and pre- and post-processing modules. There is a floating-point unit (FPU) included, providing up to four single-precision IEEE floating-point multiplier–accumulators (MAC) per cycle. All HiFi4 audio engine operations can be used as intrinsics in low-level applications.

60.2.1 Block diagram

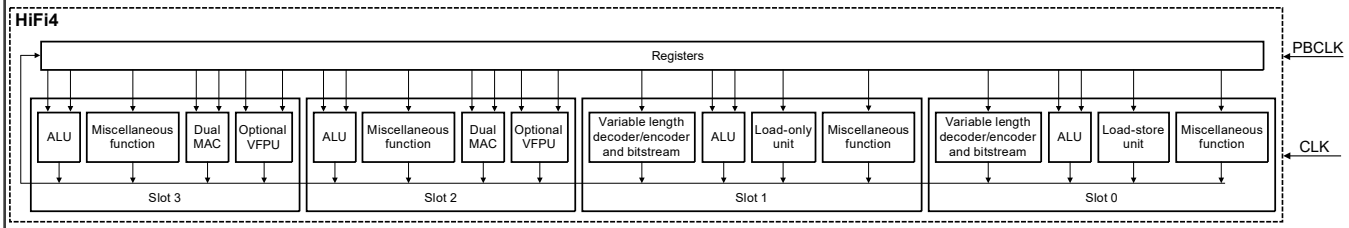


Figure 255. HiFi4 block diagram

60.2.2 Features

HiFi4 supports the following features:

- Vector floating-point unit (VFPU)
- 256 bytes L0 loop buffer
- Memory protection: Region protection
- Pipelined + UH/SH Multiplication (MUL) 32 implementation
- MUL 16
- 16-bit MAC with 40-bit accumulator
- 32-bit integer divider
- Non-IEEE double-precision FP accelerator
- Take exception on unaligned load and store address
- Normal L32R hardware support
- 7-stage pipeline
- 16 cache prefetch entries
- Instruction cache features:
  - 4-way instruction cache
  - Cache size: 32768
  - Line size: 256
  - Line locking
- Data cache features:
  - 4-way data cache
  - Cache size: 65536
  - Line size: 256
  - Write back
  - Line locking
  - 4 data banks
- Instruction RAM features:
  - Supports inbound processor interface
  - Size: 64 K

— Banks: 1

NOTE

The instruction RAM must be accessed in a 32-bit word unit. Each 32-bit word must have the contiguous four bytes all enabled or all disabled.

- Data RAM features:
  - Supports inbound processor interface
  - Size: 64 K
  - Banks: 4
- 2 load and store units

60.3 Functional description

60.3.1 Arithmetic logic unit (ALU) and multiplier–accumulator (MAC)

In HiFi4 DSP, MAC is a common process that computes the product of two numbers and adds the results to an accumulator. ALU performs arithmetic and logical operations on the operands. HiFi4 supports the following ALU and MAC functionalities:

- Four 32 x 32-bit MACs
- Limited support for 72-bit accumulators
- Ability to support eight 32 x 16-bit MACs
- 16-bit multiply and accumulate with 40-bit accumulator
- CLAMPS operations
- 32-bit integer divider

60.3.2 Vector floating-point unit (VFPU)

VFPU performs operations on floating-point numbers. HiFi4 provides up to four single-precision IEEE floating-point MACs per cycle.

60.3.3 Variable-length decoder and encoder

The HiFi4 bitstream engine supports fixed length and variable length decoding and encoding. Variable length (Huffman) decode instructions are specialized instructions, in which the elements with variable bit-widths are decoded.

60.3.4 Load and store operations

HiFi4 supports loading and storing scalars or vectors of 16, 24, 32, and 64 bits. Each scalar of load and store accesses 16, 24, 32, or 64 bits. Each vector accesses 64 bits or 48 bits for packed 24-bit data.

60.3.5 Non-IEEE double-precision floating-point (FP) accelerator

The double-precision FP accelerator adds a small set of TIE instructions and states to speed existing double-precision software execution of IEEE 754 double precision FP calculations. This implements IEEE 754 compliant 64-bit double-precision add, subtract, multiply, square root, and divide operations with the round to-nearest rounding mode. It also implements a complete set of comparison operations that allow for IEEE-compliant comparisons.

60.3.6 L0 loop buffer

L0 loop buffer captures instructions from the body of a Zero Overhead Loop, and then executes subsequent iterations of the loop from this buffer. This allows the instruction memories to be disabled during most of the loop execution. Power savings depend on what portion of code is being executed as Zero Overhead Loops.

60.3.7 Miscellaneous function

Miscellaneous functions in HiFi4 include:

- NSA/NSAU: Returns the number of contiguous sign bits in the most significant bits of a 32-bit signed/unsigned value
- MinMax: Performs a signed or unsigned integral minimum or maximum operation
- SEXT: Sign extend to 32 bits
- Density: Enable use of density (16-bit) instructions

60.3.8 PFaultInfo and Fault Information Register (FIR)

FIR provides information on the error condition and the PFaultInfo output signal mirrors this internal register.

Bit	Description	Notes
31:21	Deasserted	
20	Multiple Faults 0b - Only one fault has occurred so far 1b - Multiple faults have occurred*	* In this case, the FIR register has information only on the most severe fault.
19:16	User Bits These bits are zero by default. Software running on Xtensa can write to the user bits.	These bits are available only in configurations with both debug features (i.e. OCD at a minimum) and parity/ECC protection on the AXI bus interface.
15**:8	Fault Severity Level 0000b - Fault severity is not reset. The system can recover without resetting 0001b - Fault severity is core reset. Minimally, reset the Xtensa processor to recover from the fault 0010b - Fault severity is system reset. The whole system may need to be reset to recover from the fault	** CSR parity error sets the PFaultInfo[15] output signal. **NOTE** The suggested severity level and the recovery mechanism are recommendations and may not always work.
7:0	AXI Channel and AXI Port AXI channel [7:4]: 0000b - Read address AXI channel 0001b - Write address AXI channel 0010b - Write data AXI channel 0011b - Read data AXI channel 0100b - Write response AXI channel AXI port [3:0]: 0000b - Main AXI initiator port 0001b - Secondary (iDMA) AXI initiator port 0010b - AXI target port	

60.3.9 Clocking

Clock	Description
CLK	Main clock
PBCLK	APB interface clock
ATCLK	ATB clock

60.3.10 Interrupts

HiFi4 provides 32 interrupts with 4 priority levels, 2 timers, and 2 EXCM (exception mode) priority levels with highest priority of efficiently callable handlers.

60.4 Signals

Signal name	Description
OCDHaltOnReset	On-Chip Debug (OCD) halt on reset Tied low for normal operation. Alternatively, this signal can be tied high to force the processor to enable OCD and enter Stopped (halt) state at processor reset.
XOCDMode	Xtensa On-Chip Debug (XOCD) mode Indicates that the core is in Stopped (halt) or Stepping state.
DoubleExceptionError	Double exception error Indicates a double exception fault occurs.
PFatalError	Processor fatal error Indicates a fatal error condition occurs (e.g., parity error on AXI handshake signal or CSR parity error).
PFaultInfo	Processor fault information PFaultInfo[31:0] mirrors the processor's internal FIR, using the same bit encoding to provide the source and severity of the fault. See PFaultInfo and Fault Information Register (FIR).
PFaultInfoValid	PFaultInfo valid Asserted when PFaultInfo signal changes its value. The system can use its high level as an enabler to capture PFaultInfo.
ARADDR	AXI read address.
TIE_EXPSTATE	Tensilica Instruction Extension (TIE) output state 32-bit set by instructions which can be connected to any system peripherals, independent of main system bus.

Chapter 61
Micro-Tick Timer (UTICK)

61.1 Chip-specific UTICK information

Table 585. Reference links to related information

Topic	Related module	Reference
Full description	UTICK	UTICK
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

61.1.1 Module instances

This device has two instances of the UTICK module, UTICK0 and UTICK1.

Table 586. UTICK instances

	UTICK0	UTICK1
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

61.1.2 UTICK clock source

The table below shows the UTICK clock source, see Clock Controller (CLKCTL) for details.

Table 587. UTICK clock source

UTICK clock	UTICK0	UTICK1
timer_clk	utick0_fclk	utick1_fclk
pclk	VDD2_COMP domain main clock	VDD1_SENSE domain main clock

61.1.3 Wakeup chip from low power mode

UTICK can run and wakeup this chip from the following low power modes:

- Sleep mode
- Deep sleep mode

61.2 Overview

UTICK is a 31-bit timer that counts down to 0 and then generates an interrupt. Thus, it provides a fixed time interval between interrupts. This is a simple, ultra-low-power timer that can run and wake up the chip from Low-power mode. See chip specific information for low power modes UTICK can wakeup.

61.2.1 Block diagram

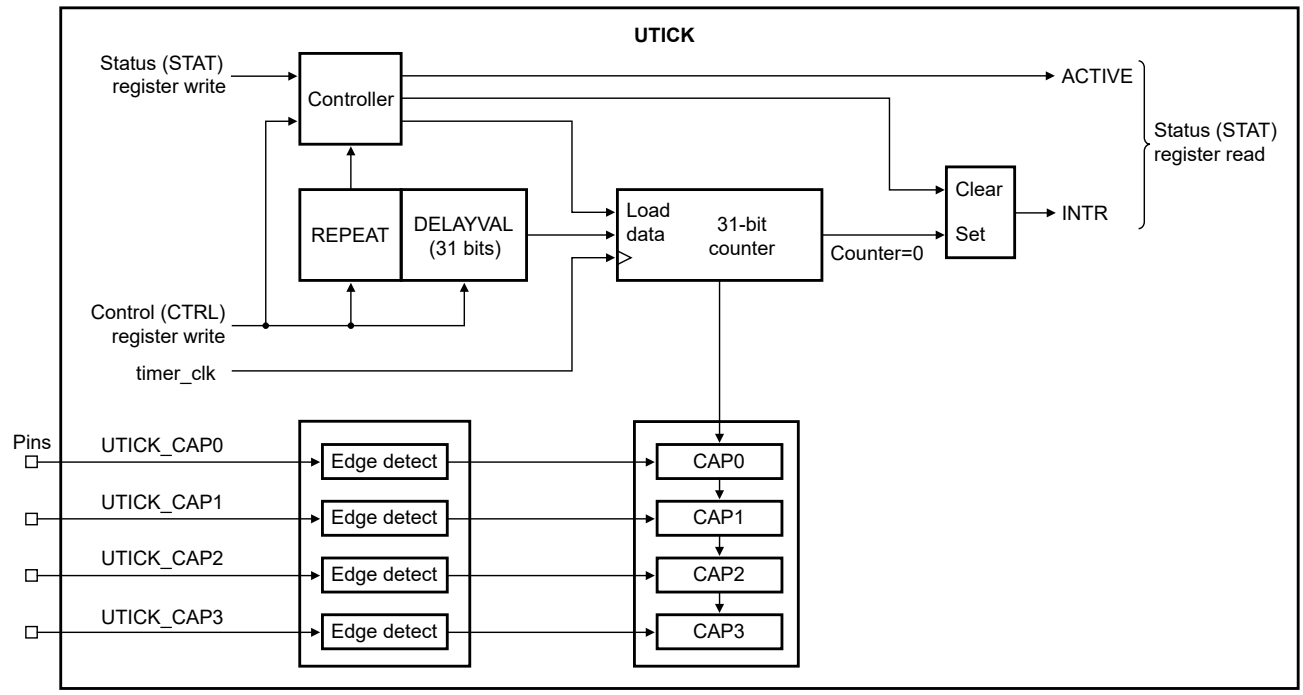


Figure 256. Block diagram

61.2.2 Features

- Wakes up the chip from Low-power mode.
- Starts with a nonzero value written to CTRL[DELAYVAL].
- Operates using interrupt or software polling mode.
- Includes Capture (CAP0 - CAP3) registers that use external pin transitions as a trigger.

61.3 Functional description

61.3.1 Operations

Perform the following procedure before using UTICK:

1. Configure CTRL[REPEAT] to either perform a one-time delay or continuous delay.
2. Configure CTRL[DELAYVAL] to specify the delay time value in terms of the timer clock.
3. Configure CFG[CAPEN*n*] to decide the number of events to capture (the maximum is four).
4. Configure CFG[CAPPOL*n*] to capture the polarity (whether to capture on a positive or negative edge of a signal).

61.3.2 Clocking

The UTICK clock is selectable from the following clock inputs:

Table 588. Clocking

Clock	Description
Global functional clock (timer_clk)	This clock is supposed to be on during normal operation. This clock is treated as asynchronous to pclk. **NOTE** See the Chip specific section for more details on UTICK clock.
System bus clock (pclk)	This clock is used only for register reads and writes.

61.3.3 Interrupts

The interrupt is generated when the tick_count reaches 1h so that on the next rising edge of timer_clk, it doesn't trigger again automatically. Its associated status field is STAT[INTR].

61.4 External signals

Signal	I/O	Description
UTICK_CAP0, UTICK_CAP1, UTICK_CAP2, UTICK_CAP3	I	Capture inputs. Configure the selected transition on a capture pin to load Capture (CAP0 - CAP3) with the counter value.

61.5 Initialization

To initialize UTICK:

1. Enable the clock to the UTICK register interface.
2. Enable UTICK interrupts for waking up from Low-power mode, configure Control (CTRL) register.
3. Configure Capture Configuration (CFG) register to enable UTICK capture functions and selects the polarity of capture triggers.
4. Enable the oscillator that provides the UTICK clock, after writing values corresponding registers UTICK timer works normally.

61.6 UTICK register descriptions

61.6.1 UTICK memory map

UTICK0 base address: 4000_F000h

UTICK1 base address: 4005_2000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (CTRL)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Status (STAT)	32	RW	See section
8h	Capture Configuration (CFG)	32	RW	See section
Ch	Capture Clear (CAPCLR)	32	W	See section
10h - 1Ch	Capture (CAP0 - CAP3)	32	R	0000_0000h

61.6.2 Control (CTRL)

Offset

Register	Offset
CTRL	0h

Function

Resets the counter, which means a new interval is measured if one is in progress.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	REPEAT	DELAYVAL														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DELAYVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 REPEAT	Repeat Delay Specifies the frequency of repeat delay. 0b - One-time delay 1b - Delay repeats continuously
30-0	Tick Interval

Table continues on the next page...

Table continued from the previous page...

Field	Function
DELAYVAL	Specifies the tick interval. The resulting delay is (DELAYVAL + 1) periods of the timer clock. The minimum usable tick interval is one clock cycle, for a delay of two timer clocks. The timer will be stopped when the tick interval is equal to zero. 000_0000_0000_0000_0000_0000_0000_0000b - Reserved All other values - Clock cycles as defined in the description

61.6.3 Status (STAT)

Offset

Register	Offset
STAT	4h

Function

Indicate the micro timer is working or stopping and any event on output compare channels. The bits are write one to clear.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R															ACTIVE	INTR
W	Reserved															W1C
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0	0

Fields

Field	Function
31-2 —	Reserved
1 ACTIVE	Timer Active Flag Indicates whether UTICK is currently active. 0b - Inactive (stopped)

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Active
0 INTR	Interrupt Flag Indicates whether an interrupt is pending. 0b - Not pending 1b - Pending

61.6.4 Capture Configuration (CFG)

Offset

Register	Offset
CFG	8h

Function

Enables micro-tick capture functions and selects the polarity of capture triggers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					CAPPOL3	CAPPOL2	CAPPOL1	CAPPOL0					CAPEN3	CAPEN2	CAPEN1	CAPEN0
W	Reserved								Reserved							
Reset	u	u	u	u	0	0	0	0	u	u	u	u	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11 CAPPOL3	Capture Polarity 3 Specifies the capture edge. 0b - Positive

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Negative
10 CAPPOL2	Capture Polarity 2 Specifies the capture edge. 0b - Positive 1b - Negative
9 CAPPOL1	Capture-Polarity 1 Specifies the capture edge. 0b - Positive 1b - Negative
8 CAPPOL0	Capture Polarity 0 Specifies the capture edge. 0b - Positive 1b - Negative
7-4 —	Reserved
3 CAPEN3	Enable Capture 3 0b - Disable 1b - Enable
2 CAPEN2	Enable Capture 2 0b - Disable 1b - Enable
1 CAPEN1	Enable Capture 1 0b - Disable 1b - Enable
0 CAPEN0	Enable Capture 0 0b - Disable 1b - Enable

61.6.5 Capture Clear (CAPCLR)

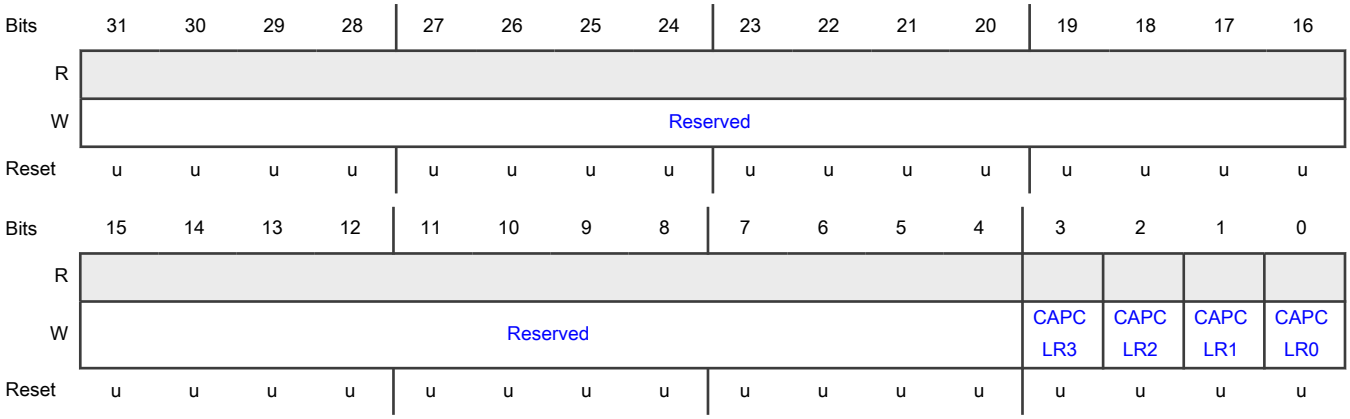
Offset

Register	Offset
CAPCLR	Ch

Function

Allows you to clear previous capture values, enabling new captures to take place.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 CAPCLR3	Clear Capture 3 0b - Does nothing 1b - Clears the CAP3 register value
2 CAPCLR2	Clear Capture 2 0b - Does nothing 1b - Clears the CAP2 register value
1 CAPCLR1	Clear Capture 1 0b - Does nothing 1b - Clears the CAP1 register value
0 CAPCLR0	Clear Capture 0 0b - Does nothing 1b - Clears the CAP0 register value

61.6.6 Capture (CAP0 - CAP3)

Offset

Register	Offset
CAP0	10h
CAP1	14h
CAP2	18h
CAP3	1Ch

Function

Contains the micro-tick time value based on any previously captured events. Each capture register is associated with one of the capture trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALID	CAP_VALUE														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAP_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VALID	Captured Value Valid Flag Specifies whether a valid value is captured, based on a transition of the related UTICK_CAP*n* pin. Write 1 to the related field in Capture Clear (CAPCLR) to clear this flag. 0b - Valid value not captured 1b - Valid value captured
30-0 CAP_VALUE	Captured Value for the Related Capture Event Specifies the captured value for the related capture event. The captured value is 1 lower than the actual value of UTICK at the moment of the capture event.

Chapter 62
Standard Counter/Timers (CTIMER)

62.1 Chip-specific CTIMER information

Table 589. Reference links to related information

Topic	Related module	Reference
Full description	CTIMER	CTIMER
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Input multiplexing	INPUTMUX	See CTIMERn_TRIG and CTIMERn_CAPm (n=0-7, m=0-3) registers in INPUTMUX

62.1.1 Module instances

This device has eight instances of the CTIMER module, CTIMER0, CTIMER1, CTIMER2, CTIMER3, CTIMER4, CTIMER5, CTIMER6 and CTIMER7.

Table 590. CTIMER instances

	CTIMER0-CTIMER4	CTIMER5-CTIMER7
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

62.1.2 Signals

The tables below give a summary of each of the Timer/Counter related pins and the recommended PORT settings. Also note that different part number and package variations may provide different CTIMER related pin functions.

Table 591. Timer/Counter pin description

Pin	Type	Description
CTIMER0_CAP3:0 CTIMER1_CAP3:0 CTIMER2_CAP3:0 CTIMER3_CAP3:0	Input	Capture Signals- A transition on a capture pin can be configured to load one of the Capture Registers with the value in the Timer Counter and optionally generate an interrupt. Capture functionality can be selected from a number of pins.

Table continues on the next page...

Table 591. Timer/Counter pin description (continued)

Pin	Type	Description
CTIMER4_CAP3:0 CTIMER5_CAP3:0 CTIMER6_CAP3:0 CTIMER7_CAP3:0		Timer/Counter block can select a capture signal as a clock source instead of the APB bus clock. For more details, see CTCR register.
CTIMER0_MAT3:0 CTIMER1_MAT3:0 CTIMER2_MAT3:0 CTIMER3_MAT3:0 CTIMER4_MAT3:0 CTIMER5_MAT3:0 CTIMER6_MAT3:0 CTIMER7_MAT3:0	Output	External Match Output - When a match register (MR3:0) equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.

62.1.3 Initialization

- Enable the clock to the CTIMER. This enables the register interface and the peripheral function clock.
  - CTIMER*n* in compute domain via the CLKCTL0_PSCCTL2[CTIMER*n*] (*n* = 0-4)
  - CTIMER*n* in sense domain via the CLKCTL1_PSCCTL1[CTIMER*n*] (*n* = 5-7)
- Select a clock source for the CTIMER.
  - CTIMER*n* in compute domain via the CLKCTL0_CT32BIT*n*FCLKSEL (*n* = 0-4) registers
  - CTIMER*n* in sense domain via the CLKCTL1_CT32BIT*n*FCLKSEL (*n* = 5-7) registers
- Each CTIMER provides interrupts to the NVIC. See MCR and CCR registers in the CTIMER register section for match and capture events. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register. For interrupt connections, see NVIC Interrupt Assignments (See Interrupts chapter).
- Select timer pins and pin modes as needed through the relevant PORT registers.
- The CTIMER DMA request lines are connected to the DMA trigger inputs via the DMAC0_ITRIG_INMUX registers. Note that timer DMA request outputs are connected to DMA trigger inputs.

62.1.4 External global enable

When the timer is enabled by the CTIMER global start enable register in SYSCON (CTIMERGLOBALSTARTEN), it is equivalent to writing 1 to the TCR[CEN] bit in CTIMER. The external global start register is used to enable multiple timers at the same time. This operation is allowed by writing a 1 to each Timer's TCR[AGCEN] bit. If the Timer control register's CEN bit is already 1, the external global start enable register has no effect.

See CTIMERGLOBALSTARTEN in SYSCON for more information about CTIMER global start enable register.

62.1.5 Peripheral input multiplexers

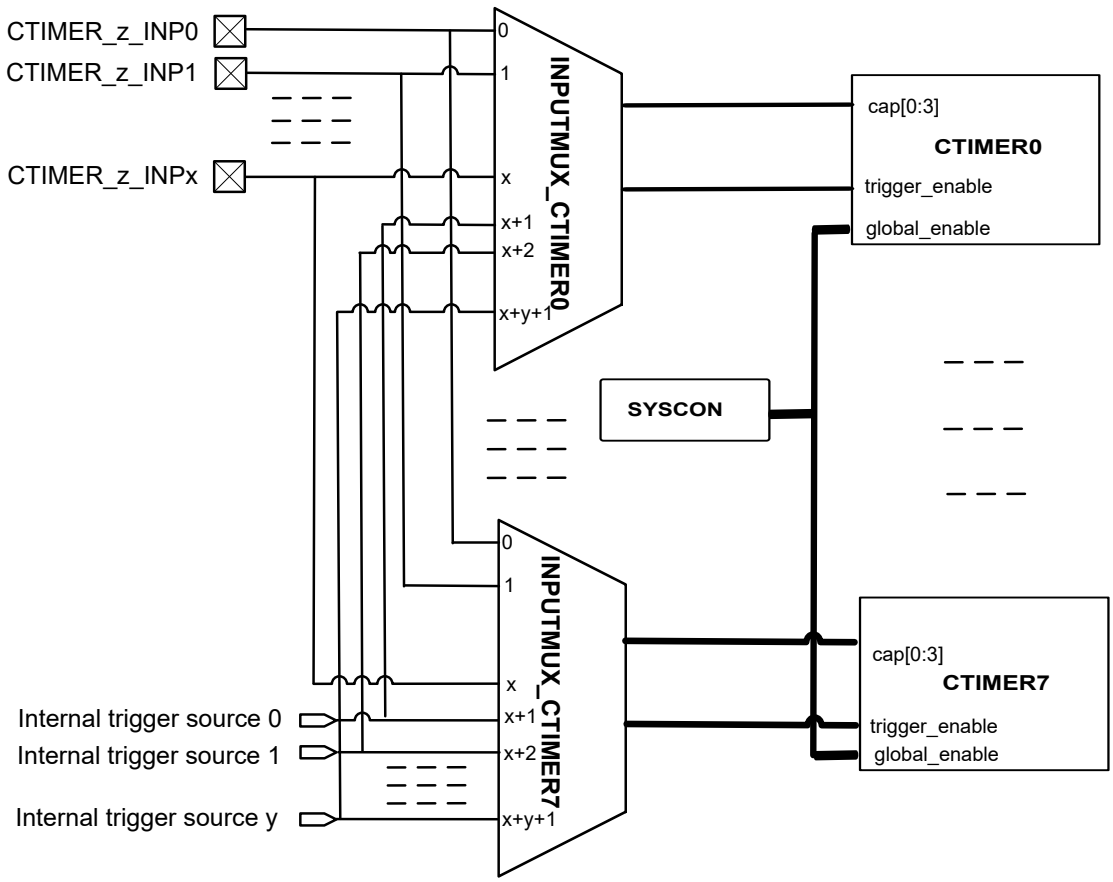


Figure 257. Peripheral input multiplexers for CTIMER

For CTIMER_z_INPx (z=C for CTIMER0 to CTIMER4, z=S for CTIMER5 to CTIMER7) and internal trigger sources, see CTIMERn_TRIG and CTIMERn_CAPm (n=0-7, m=0-3) registers in INPUTMUX for details.

62.2 Overview

Each CTIMER is designed to count cycles of the CTIMER function clock or an externally supplied clock. A CTIMER can optionally generate an interrupt or perform other actions at specified timer values based on the settings of Match (MR0 - MR3). Each CTIMER also includes capture input pins (see Capture mode) to capture the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, three match registers can be used to provide a single-edge controlled PWM output on the match output pins. One match register is used to control the PWM cycle length. All match registers can optionally be auto-reloaded from a companion shadow register (see Match Shadow (MSR0 - MSR3)) whenever the counter is reset to zero. This feature permits modifying the match values for the next counter cycle without disrupting PWM waveforms during the current cycle. When enabled, match reload occurs whenever the counter is reset, either because of a match event or by writing 1 to TCR[CRST].

62.2.1 Block diagram

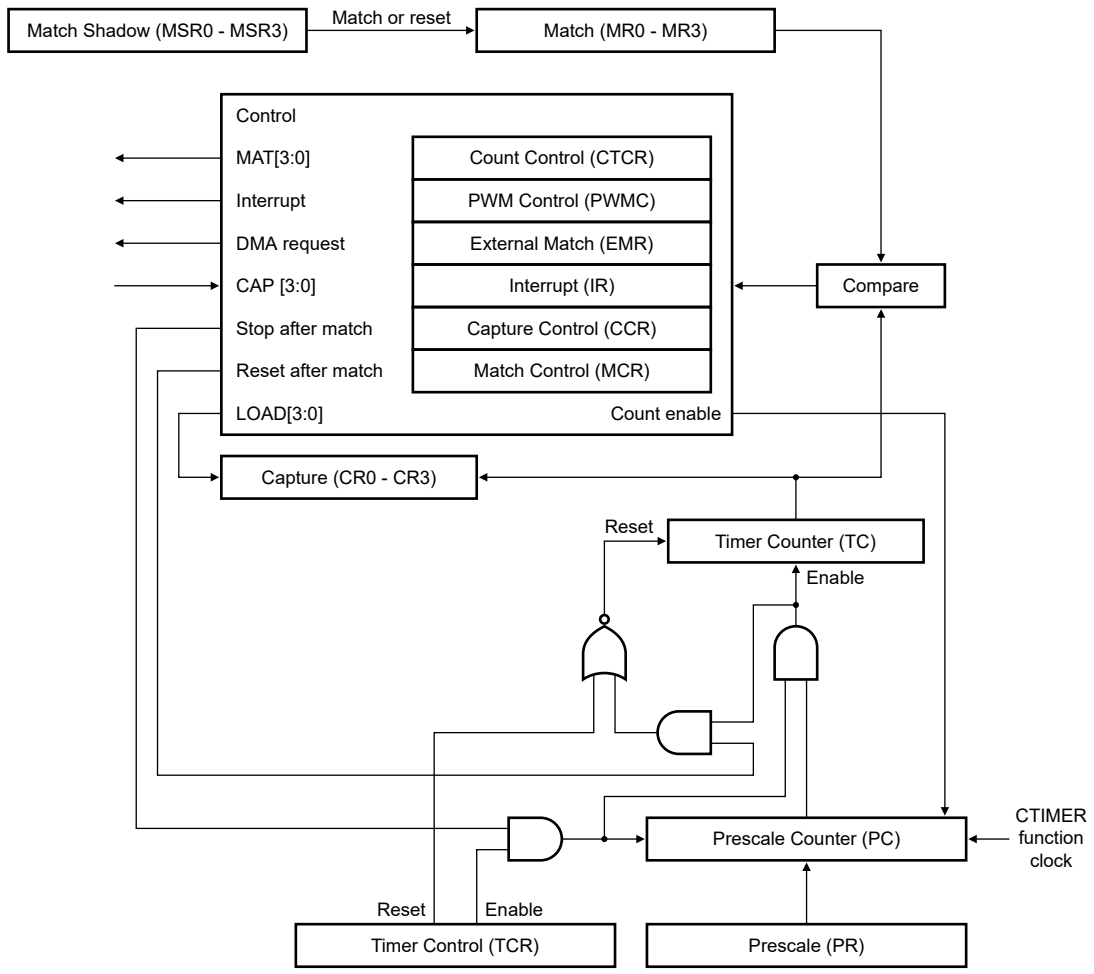


Figure 258. Block diagram

62.2.2 Features

- Contains a 32-bit CTIMER with a programmable 32-bit prescaler (the timers include external capture and match pin connections)
- Contains Timer Control (TCR) and External trigger enable, either of which you can use to enable each timer (you can also use an external global start enable register to globally enable one or more timers)
- Performs counter or timer operations
- Selects the function clocks that may be asynchronous to other system clocks
- Includes up to four 32-bit captures (see Capture mode) to take a snapshot of the timer value when an input signal transitions (a capture event may also optionally generate an interrupt, and the number of capture inputs available for each timer on a chip may vary)
- Enables you to configure the timer and prescaler to be cleared on a designated capture event (this feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse, capturing the timer value on the trailing edge)
- Includes Match (MR0 - MR3) to allow:
  - Continuous operation with optional interrupt generation after a match occurs

  - Optional auto-reload from Match Shadow (MSR0 - MSR3) when the counter is reset
  - Stop timer after a match with optional interrupt generation
  - Reset timer after a match with optional interrupt generation
- Generates up to four external outputs corresponding to Match (MR0 - MR3), for each timer, with the following capabilities (the number of match outputs available for each timer on a chip may vary):
  - Go low when matched
  - Go high when matched
  - Toggle when matched
  - Do nothing
- Enables you to configure up to four match registers for PWM operation, allowing up to three single-edged controlled PWM outputs (the number of match outputs available for each timer on a chip may vary)
- Uses up to two match registers to generate DMA requests

62.3 Functional description

Figure 259 shows a timer configured to reset the count and generate an interrupt after a match. The value of Prescale (PR) is 2 and the value of Match (MR0 - MR3) is 6. At the end of the timer cycle where the match occurs, the timer count is reset and gives a full-length cycle to the match value. The interrupt indicating that a match has occurred is generated in the next clock after the timer reaches the match value. Timer mode is selected in this case.

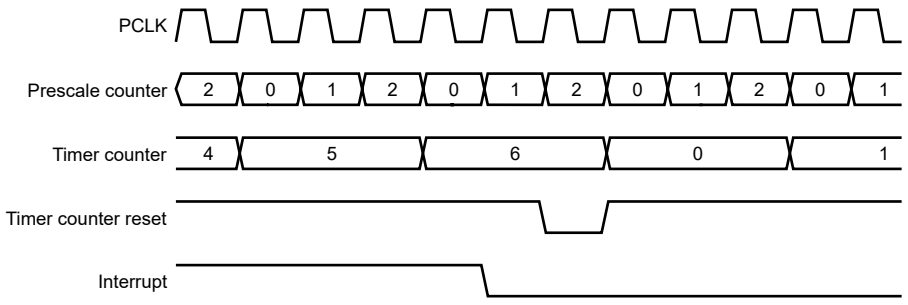


Figure 259. Timer cycle with PR = 2, MR*x* = 6, and both interrupt and reset enabled after match

Figure 260 shows a timer configured to stop and generate an interrupt after a match. The value of Prescale (PR) is 2 and the value of Match (MR0 - MR3) is 6. In the next clock after the timer reaches the match value, TCR[CEN] becomes 0 and the interrupt indicating that a match has occurred is generated.

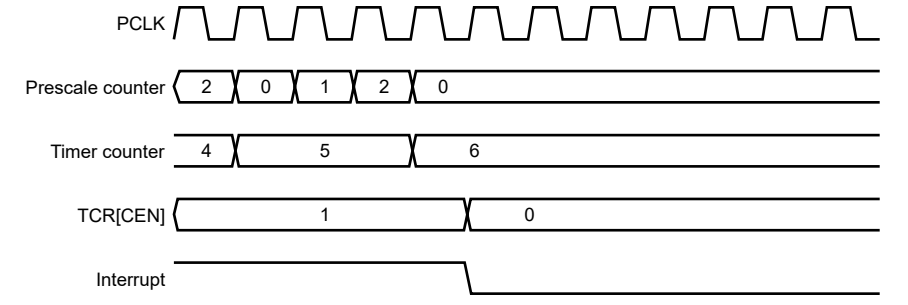


Figure 260. Timer cycle with PR = 2, MR*x* = 6, and both interrupt and stop enabled after match

62.3.1 Operation sections

All single-edge controlled PWM outputs go low at the beginning of each PWM cycle (timer is set to zero) unless their match value is zero.

Each PWM output goes high when its match value is reached. If no match occurs (that is, the match value is greater than the PWM cycle length), the PWM output remains continuously low.

If a match value larger than the PWM cycle length is written to Match (MR0 - MR3), and the PWM signal is already high, the PWM signal is cleared at the start of the next PWM cycle.

If a match register contains the same value as the timer reset value (the PWM cycle length), the PWM output is reset to low on the next clock tick after the timer reaches the match value. Therefore, the PWM output always consists of one clock tick wide positive pulse with a period that the PWM cycle length determines (that is, the timer reload value).

If Match (MR0 - MR3) = 0, the PWM output goes to high the first time the timer goes back to zero, and remains high continuously.

The following figure shows sample PWM waveforms with a PWM cycle length of 100 that MR3 specifies and PWM outputs that PWM Control (PWMC) enables. Counter mode is selected in this case.

NOTE

When you select match outputs to perform as PWM outputs, you must write 0 to MCR[MR*n*R] and MCR[MR*n*S], except to the match register setting the PWM cycle length. For this register, write 1 to MCR[MR*n*R] to enable the timer reset when the timer value matches the value of the corresponding match register.

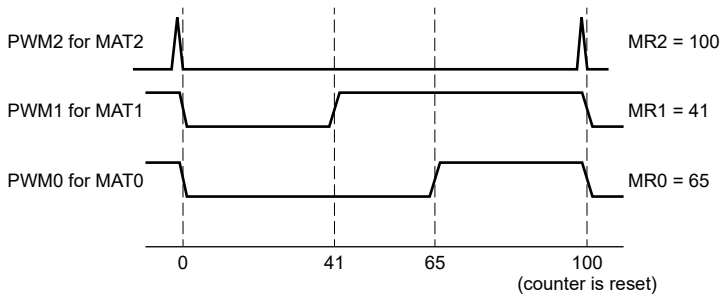


Figure 261. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWM Control register.

62.3.2 Mode sections

62.3.2.1 Capture mode

You can configure a CAP to load Capture (CR0 - CR3) with the value in the counter or timer and optionally generate an interrupt. One of the pins generates the capture signal with a capture function. Each CAP is connected to one capture channel of the timer.

CTIMER can select a CAP as a clock source instead of selecting the APB CTIMER function clock. See Count Control (CTCR) for more information.

62.3.2.2 Match mode

When the value of a match register (Match (MR0 - MR3)) equals that of Timer Counter (TC), the corresponding match output can either toggle, go low, go high, or do nothing. External Match (EMR) and PWM Control (PWMC) control the functionality of this output.

62.3.3 External trigger enable

When trigger_enable is asserted (rising edge), it is equivalent to writing 1 to TCR[CEN]. If TCR[CEN] is already 1, trigger_enable rising edge events have no effect.

62.3.4 Clocking

Table 592. CTIMER clocks

Type of clock	Description
Functional Clock(ctclk)	Asynchronous to the bus clock, provide clock to perform counter or timer operations.
Bus Clock(pclk)	The bus clock is only used for bus accesses to the control and configuration registers.

62.3.5 Reset

TCR[CRST] resets Timer Counter (TC) and Prescale Counter (PC).

62.3.6 Interrupts

This module has capture and match interrupts. See the Interrupt (IR)

62.3.7 DMA

DMA requests are generated when the value of Timer Counter (TC) either matches the value of MR0 or MR1 (see Match (MR0 - MR3) for more information). This DMA request is not connected to the operation of the match outputs (see Match mode) controlled by External Match (EMR). Each match sets a DMA request flag, which is connected to the DMA controller. You must configure the DMA controller correctly.

When a timer is initially set up to generate a DMA request, the request may already be asserted before a match condition occurs. An initial DMA request may be avoided by writing 1 to the interrupt flag location twice, as if clearing a timer interrupt. See Interrupt (IR) for more information. A DMA request is cleared automatically when the DMA controller manages it.

NOTE

Timer DMA requests are generated whenever the timer value is equal to the related match register value. DMA requests are always generated when the timer is running, unless the match register value is higher than the upper count limit of the timer. It is important not to select and enable timer DMA requests in the DMA block unless you configure the timer correctly to generate valid DMA requests.

62.4 External signals

Table 593. CTIMER pin descriptions

Pin	Type	Description
CTIMER*m*_CAP*n*	Input	Capture signals—You can configure a transition on a capture pin to load one of the capture registers (Capture (CR0 - CR3)) with the value of Timer Counter (TC) and optionally generate an interrupt. You can select the capture functionality for a number of pins. CTIMER can select a capture signal as a clock source instead of the APB bus clock. See Count Control (CTCR) for more information.
CTIMER*m*_MAT*n*	Output	External match output—When the value of a match register (Match (MR0 - MR3)) equals the value of Timer Counter (TC), this output can either toggle, go low, go high, or do nothing. External Match (EMR) controls the functionality of this output. You can select the match output functionality for a number of pins in parallel.

62.4.1 Multiple capture (CAP) and match (MAT) pins

You can configure the pins of the chip to be used for this module. These pins are generally reused, and you can connect several internal module pins selectively. However, you can connect only one internal pin at a time.

NOTE

Match conditions may be used internally without the use of a chip's pin.

62.5 Initialization

See the Chip-specific CTIMER information for the initialization section for more details or steps.

62.6 Application information

CTIMER register descriptions discusses the following:

- Interval timer for counting internal events
- PWM outputs
- Pulse width demodulator via capture inputs
- Free running timer

62.7 CTIMER register descriptions

62.7.1 CTIMER memory map

CTIMER0 base address: 4002_8000h

CTIMER1 base address: 4002_9000h

CTIMER2 base address: 4002_A000h

CTIMER3 base address: 4002_B000h

CTIMER4 base address: 4002_C000h

CTIMER5 base address: 4004_8000h

CTIMER6 base address: 4004_9000h

CTIMER7 base address: 4004_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Interrupt (IR)	32	RW	See section
4h	Timer Control (TCR)	32	RW	See section
8h	Timer Counter (TC)	32	RW	0000_0000h
Ch	Prescale (PR)	32	RW	0000_0000h
10h	Prescale Counter (PC)	32	RW	0000_0000h
14h	Match Control (MCR)	32	RW	See section
18h - 24h	Match (MR0 - MR3)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
28h	Capture Control (CCR)	32	RW	See section
2Ch - 38h	Capture (CR0 - CR3)	32	R	0000_0000h
3Ch	External Match (EMR)	32	RW	See section
70h	Count Control (CTCR)	32	RW	See section
74h	PWM Control (PWMC)	32	RW	See section
78h - 84h	Match Shadow (MSR0 - MSR3)	32	RW	0000_0000h

62.7.2 Interrupt (IR)

Offset

Register	Offset
IR	0h

Function

Provides flags for match interrupts and capture interrupts. If an interrupt is generated, it sets the corresponding flag in this register. Otherwise, the flag remains cleared. Writing 1 to the corresponding IR field resets the interrupt. Writing 0 has no effect. Clearing an interrupt for a timer match also clears any corresponding DMA request. You can write to this register to clear interrupts, and can read this register to identify which interrupt sources are pending.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CR3INT	CR2INT	CR1INT	CR0INT	MR3INT	MR2INT	MR1INT	MR0INT
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-4 CRnINT	Interrupt Flag for Capture Channel n Event Controls an interrupt flag for capture channel *n* events.
3-0 MRnINT	Interrupt Flag for Match Channel n Event Controls an interrupt flag for match channel *n* events.

62.7.3 Timer Control (TCR)

Offset

Register	Offset
TCR	4h

Function

Controls timer counter functions. You can disable or reset Timer Counter (TC) by using this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										ATCEN	AGCEN	Reserved		CRST	CEN
Reset	u	u	u	u	u	u	u	u	u	u	0	0	u	u	0	0

Fields

Field	Function
31-6 —	Reserved
5 ATCEN	Allow Trigger Count Enable Enables the input trigger. If this field is 1, it allows the input trigger_enable = 1 action to take effect (see External trigger enable). If this field is 0, the action is not allowed. 0b - Disable 1b - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 AGCEN	Allow Global Count Enable Enables the global count. If this field is 1, it allows the input global_enable = 1 action to take effect. If this field = 0, the action is not allowed. 0b - Disable 1b - Enable
3-2 —	Reserved
1 CRST	Counter Reset Enable Enables the counter reset. If this field is 1, Timer Counter (TC) and Prescale Counter (PC) are synchronously reset after the next positive edge of the CTIMER function clock. The counters remain reset until this field becomes 0. If this field is 0, no effect takes place. 0b - Disable 1b - Enable
0 CEN	Counter Enable Enables the counters. If this field is 1, Timer Counter (TC) and Prescale Counter (PC) are enabled. When an external trigger enables the timer or the external global start enable register enables it, this field becomes 1 automatically. 0b - Disable 1b - Enable

62.7.4 Timer Counter (TC)

Offset

Register	Offset
TC	8h

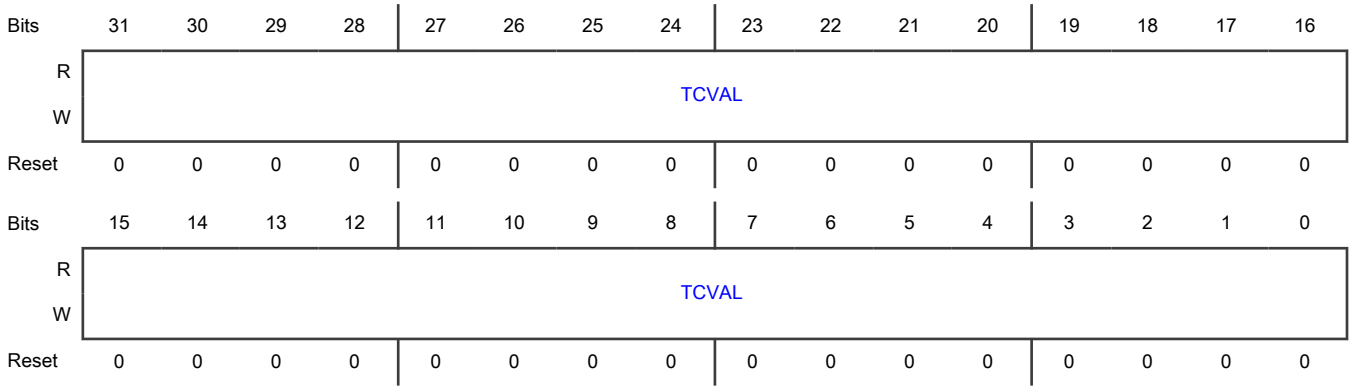
Function

Specifies the timer counter.

This register is incremented when Prescale Counter (PC) reaches its terminal count. Unless it is reset before reaching its upper limit, Timer Counter (TC) counts up through the value FFFF_FFFFh, and then wraps back to the value 0000_0000h. This event does not cause an interrupt, but you can use Match (MR0 - MR3) to detect an overflow if needed.

Timer Counter (TC) is incremented after every Prescale (PR) + 1 cycle of the CTIMER function clock. Timer Control (TCR) controls the functioning of this register.

Diagram



Fields

Field	Function
31-0 TCVAL	Timer Counter Value Specifies the value of the timer counter.

62.7.5 Prescale (PR)

Offset

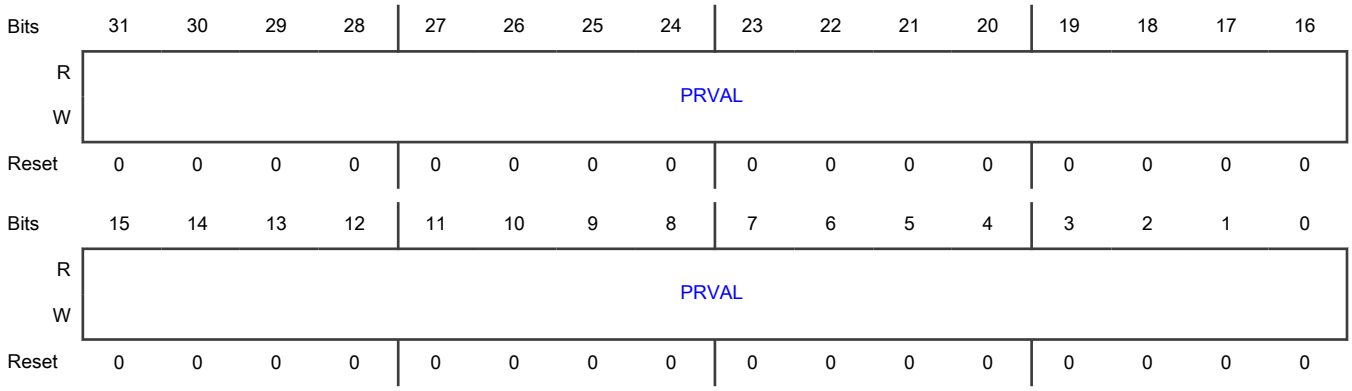
Register	Offset
PR	Ch

Function

Specifies the prescale value.

If Prescale Counter (PC) is equal to the prescale value, the next clock timer increments Timer Counter (TC) and turns the value of Prescale Counter (PC) to 0.

Diagram



Fields

Field	Function
31-0 PRVAL	Prescale Reload Value Specifies the value of the prescale reload.

62.7.6 Prescale Counter (PC)

Offset

Register	Offset
PC	10h

Function

Controls the division of the CTIMER function clock by some constant value before it is applied to Timer Counter (TC). This allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows.

Prescale Counter (PC) is incremented on every CTIMER function clock. When it reaches the value stored in Prescale (PR), Timer Counter (TC) is incremented and Prescale Counter (PC) is reset on the next CTIMER function clock. This causes Timer Counter (TC) to increment on every CTIMER function clock when Prescale (PR) = 0, every two CTIMER function clocks when Prescale (PR) = 1, and so on.

You can use the bus interface to observe and control Prescale Counter (PC).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PCVAL	Prescale Counter Value Specifies the value of the prescale counter.

62.7.7 Match Control (MCR)

Offset

Register	Offset
MCR	14h

Function

Controls the operations that are performed when one of the match registers (Match (MR0 - MR3)) matches Timer Counter (TC).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R					MR3RL	MR2RL	MR1RL	MR0RL								
W	Reserved								Reserved							
Reset	u	u	u	u	0	0	0	0	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					MR3S	MR3R	MR3I	MR2S	MR2R	MR2I	MR1S	MR1R	MR1I	MR0S	MR0R	MR0I
W	Reserved															
Reset	u	u	u	u	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 MRnRL	Reload MR Reloads with the contents of the match n shadow register (see Match Shadow (MSR0 - MSR3)) when Timer Counter (TC) is reset to zero (either via a match event or by writing 1 to TCR[CRST]). 0b - Does not reload 1b - Reloads
23-12 —	Reserved
11 MR3S	Stop on MR3 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR3 matches Timer Counter (TC). 0b - Does not stop 1b - Stops

Table continues on the next page...

Table continued from the previous page...

Field	Function
10 MR3R	Reset on MR3 Resets Timer Counter (TC) if MR3 matches its value. 0b - Does not reset 1b - Resets
9 MR3I	Interrupt on MR3 Generates an interrupt when MR3 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates
8 MR2S	Stop on MR2 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR2 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
7 MR2R	Reset on MR2 Resets Timer Counter (TC) if MR2 matches its value. 0b - Does not reset 1b - Resets
6 MR2I	Interrupt on MR2 Generates an interrupt when MR2 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates
5 MR1S	Stop on MR1 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR1 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
4 MR1R	Reset on MR1 Resets Timer Counter (TC) if MR1 matches its value. 0b - Does not reset 1b - Resets
3 MR1I	Interrupt on MR1 Generates an interrupt when MR1 matches the value in Timer Counter (TC).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Does not generate 1b - Generates
2 MR0S	Stop on MR0 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR0 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
1 MR0R	Reset on MR0 Resets Timer Counter (TC) if MR0 matches its value. 0b - Does not reset 1b - Resets
0 MR0I	Interrupt on MR0 Generates an interrupt when MR0 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates

62.7.8 Match (MR0 - MR3)

Offset

Register	Offset
MR0	18h
MR1	1Ch
MR2	20h
MR3	24h

Function

Continuously compares the value of Match (MR0 - MR3) with the value of TC[TCVAL]. When both the values are equal, actions such as the following can be triggered automatically:

- Generation of an interrupt
- Resetting of Timer Counter (TC)
- Stopping of the timer

The settings defined in Match Control (MCR) control these actions.

If MCR[MRnRL] = 1, Match (MR0 - MR3) is automatically reloaded with the current contents of its corresponding Match Shadow (MSR0 - MSR3) whenever Timer Counter (TC) becomes 0. This transfer takes place on the same clock edge that advances Timer Counter (TC) to 0.

NOTE

Timer Counter (TC) resets in response to an occurrence of a match on MATCH being used to set the cycle counter rate. A reset can also occur if you write 1 to TCR[CRST].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH	Timer Counter Match Value Triggers an action automatically when the value of this field is the same as that of TC[TCVAL].

62.7.9 Capture Control (CCR)

Offset

Register	Offset
CCR	28h

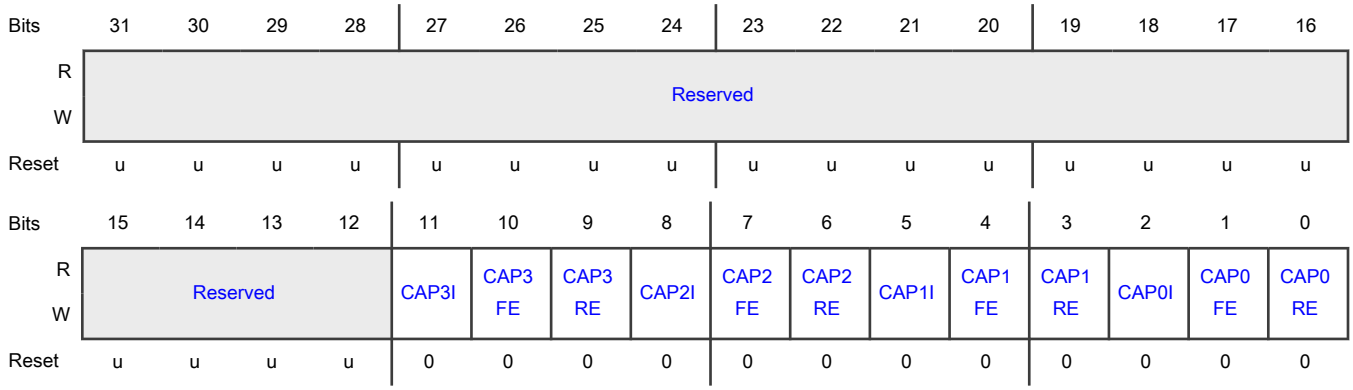
Function

Controls whether one of the capture registers (Capture (CR0 - CR3)) is loaded with the value in Timer Counter (TC) when the capture event occurs. The register also controls whether the capture event generates an interrupt. When you configure both the rising and falling bits at the same time, which is a valid configuration, it results into a capture event for both edges. In the description below, "*n*" represents the timer number, 0 or 1.

NOTE

If you select Counter mode for a particular CAP*n* input in Count Control (CTCR), you must program the fields in this register as 0. You can select both capture and interrupt for the other CAP*n* inputs.

Diagram



Fields

Field	Function
31-12 —	Reserved
11 CAP3I	Generate Interrupt on Channel 3 Capture Event Generates an interrupt on channel 3 capture event using a CR3 load. 0b - Does not generate 1b - Generates
10 CAP3FE	Falling Edge of Capture Channel 3 Loads CR3 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
9 CAP3RE	Rising Edge of Capture Channel 3 Loads CR3 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
8 CAP2I	Generate Interrupt on Channel 2 Capture Event Generates an interrupt on channel 2 capture event using a CR2 load. 0b - Does not generate 1b - Generates
7 CAP2FE	Falling Edge of Capture Channel 2 Loads CR2 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads

Table continues on the next page...

Table continued from the previous page...

Field	Function
6 CAP2RE	Rising Edge of Capture Channel 2 Loads CR2 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
5 CAP1I	Generate Interrupt on Channel 1 Capture Event Generates an interrupt on channel 1 capture event using a CR1 load. 0b - Does not generates 1b - Generates
4 CAP1FE	Falling Edge of Capture Channel 1 Loads CR1 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
3 CAP1RE	Rising Edge of Capture Channel 1 Loads CR1 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
2 CAP0I	Generate Interrupt on Channel 0 Capture Event Generates an interrupt on channel 0 capture event using a CR0 load. 0b - Does not generate 1b - Generates
1 CAP0FE	Falling Edge of Capture Channel 0 Loads CR0 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
0 CAP0RE	Rising Edge of Capture Channel 0 Loads CR0 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads

62.7.10 Capture (CR0 - CR3)

Offset

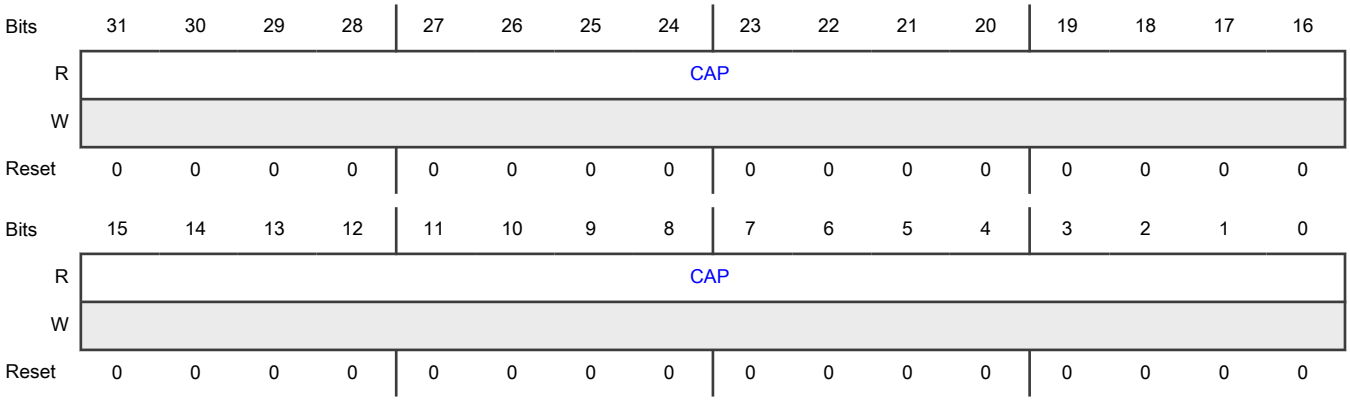
Register	Offset
CR0	2Ch
CR1	30h
CR2	34h
CR3	38h

Function

Works with a capture channel and may load with the value of Timer Counter (TC) when a specified event occurs on the signal defined for that capture channel. The signal could originate from an external pin or an internal source.

This register determines whether the capture function is enabled, and whether a capture event happens on the rising edge, falling edge, or on both edges of the associated signal. The register is loaded with the value of Timer Counter (TC) when there is an event on the CAP*n* input.

Diagram



Fields

Field	Function
31-0 CAP	Timer Counter Capture Value Determines whether the capture function is enabled and defines the edge on which the capture function occurs.

62.7.11 External Match (EMR)

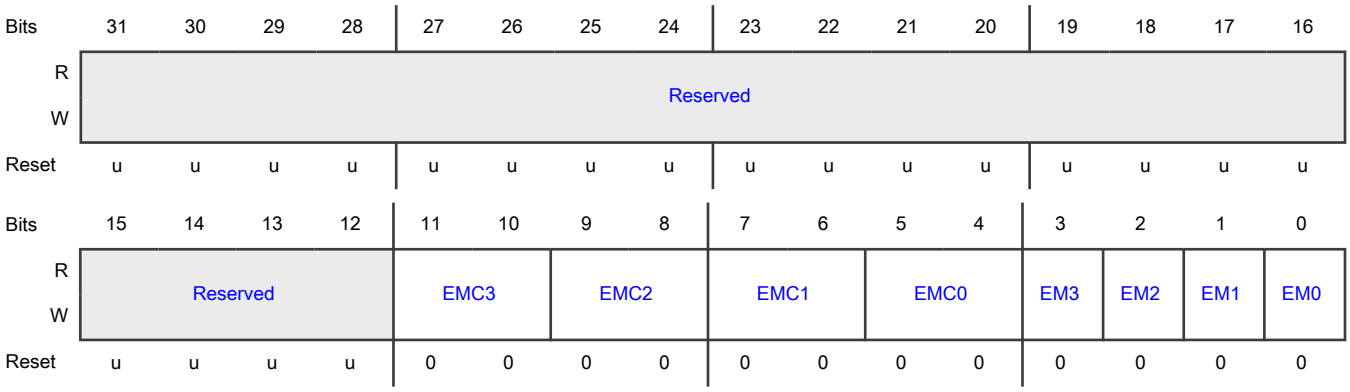
Offset

Register	Offset
EMR	3Ch

Function

Provides both control and status of the external match pins. If the match outputs (see Match mode) are configured as PWM outputs, PWM rules determine the function of this register.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-10 EMC3	External Match Control 3 Determines the functionality of EM3: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT3 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT3 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
9-8 EMC2	External Match Control 2 Determines the functionality of EM2: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT2 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT2 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01b - Goes low 10b - Goes high 11b - Toggles
7-6 EMC1	External Match Control 1 Determines the functionality of EM1: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT1 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT1 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
5-4 EMC0	External Match Control 0 Determines the functionality of EM0: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT0 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT0 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
3 EM3	External Match 3 Reflects the state of output MAT3—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR3 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC3. This field is driven to the MAT pins if the match function is selected via IOPCTL. 0b - Low 1b - High
2 EM2	External Match 2 Reflects the state of output MAT2—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR2 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or

Table continues on the next page...

Table continued from the previous page...

Field	Function
	do nothing, as selected by EMC2. This field is driven to the MAT pins if the match function is selected via IOPCTL. 0b - Low 1b - High
1 EM1	External Match 1 Reflects the state of output MAT1—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR1 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC1. This field is driven to the MAT pins if the match function is selected via IOPCTL. 0b - Low 1b - High
0 EM0	External Match 0 Reflects the state of output MAT0—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR0 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC0. This field is driven to the MAT pins if the match function is selected via IOPCTL. 0b - Low 1b - High

62.7.12 Count Control (CTCR)

Offset

Register	Offset
CTCR	70h

Function

Selects between Timer and Counter mode, and in Counter mode, selects the signal and edges for counting.

When Counter mode is selected as the mode of operation, the CAP input (selected by CINSEL) is sampled on every rising edge of the CTIMER function clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized:

- Rising edge
- Falling edge
- Either of the edges
- No change in the selected CAP input level

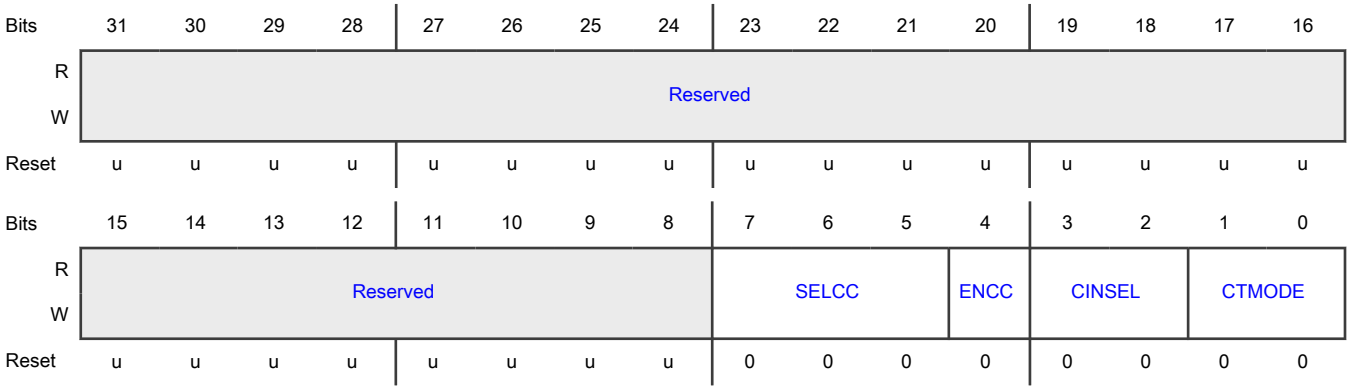
Only if the identified event occurs and the event corresponds to the one selected by CTMODE, Timer Counter (TC) is incremented.

Effective processing of the externally supplied clock to the counter has some limitations:

- Because two successive rising edges of the CTIMER function clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the CTIMER function clock.
- Consequently, the duration of the high or low levels on the same CAP input in this case cannot be shorter than the CTIMER function clock duration.

You can also use ENCC and SELCC to enable and configure the capture-clears-timer feature. This feature allows for a designated edge on a particular CAP input to reset the timer to all zeros. Using this mechanism to clear the timer on the leading edge of an input pulse and perform a capture on the trailing edge permits direct pulse-width measurement using a single capture input without the need to perform a subtraction operation in software.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-5 SELCC	Edge Select Selects which input edge (rising or falling) causes the timer and prescaler to be cleared, when ENCC is 1. These fields have no effect when ENCC is 0. 000b - Capture channel 0 rising edge 001b - Capture channel 0 falling edge 010b - Capture channel 1 rising edge 011b - Capture channel 1 falling edge 100b - Capture channel 2 rising edge 101b - Capture channel 2 falling edge 110b - Capture channel 3 rising edge 111b - Capture channel 3 falling edge
4 ENCC	Capture Channel Enable Enables capture channel. Writing 1 to this field enables clearing of the timer and the prescaler when the capture-edge event specified in SELCC occurs.

Table continues on the next page...

Table continued from the previous page...

Field	Function
3-2 CINSEL	Count Input Select Selects which CAP pin is sampled for clocking when CTMODE is not 0. **NOTE** If Counter mode is selected for a particular CAP*n* input in Count Control (CTCR), you must program the corresponding fields for that input in Capture Control (CCR) as 0. However, you can select both capture and interrupt for the other three CAP*n* inputs in the same timer. 00b - Channel 0, CAPn[0] for CTIMERn 01b - Channel 1, CAPn[1] for CTIMERn 10b - Channel 2, CAPn[2] for CTIMERn 11b - Channel 3, CAPn[3] for CTIMERn
1-0 CTMODE	Counter Timer Mode Selects between Timer and Counter mode, and in Counter mode, selects the signal and edges for counting: • If this field is 0, every rising CTIMER function clock edge is incremented. • If this field is 1, Timer Counter (TC) is incremented on the rising edges of the CAP input selected by CINSEL. • If this field is 10, Timer Counter (TC) is incremented on the falling edges of the CAP input selected by CINSEL. • If this field is 11, Timer Counter (TC) is incremented on both edges of the CAP input selected by CINSEL. 00b - Timer mode 01b - Counter mode rising edge 10b - Counter mode falling edge 11b - Counter mode dual edge

62.7.13 PWM Control (PWMC)

Offset

Register	Offset
PWMC	74h

Function

Configures match outputs (see Match mode) as PWM outputs. You can independently set each match output to perform either as a PWM output or match output controlled by External Match (EMR).

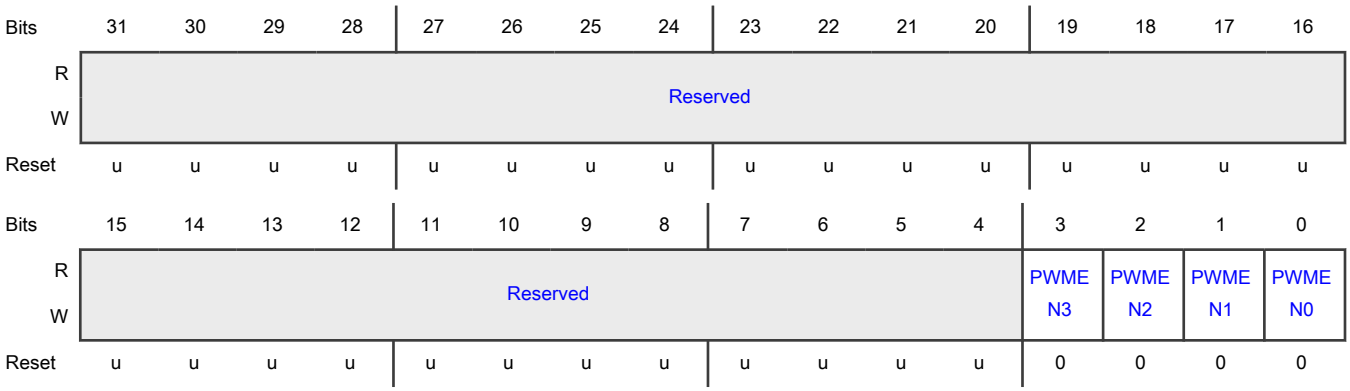
NOTE

Different part numbers and package variations may provide different match output pin functions.

For each timer, you can select a maximum of three single-edge controlled PWM outputs on the MAT*n*[2:0] outputs. One additional match register (see Match (MR0 - MR3)) determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to high. The match register resets the timer that you can configure to set the PWM cycle length. When the timer is reset to zero, all currently high match outputs configured as PWM outputs are cleared.

This register enables PWM mode for the external match pins.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 PWMEN3	PWM Mode Enable for Channel 3 Enables PWM mode for channel 3. Use match channel 3 to set the PWM cycle length. If this field is 0, EM3 controls CTIMER*n*_MAT3. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT3. 0b - Disable 1b - Enable
2 PWMEN2	PWM Mode Enable for Channel 2 Enables PWM mode for channel 2. If this field is 0, EM2 controls CTIMER*n*_MAT2. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT2. 0b - Disable 1b - Enable
1 PWMEN1	PWM Mode Enable for Channel 1 Enables PWM mode for channel 1. If this field is 0, EM1 controls CTIMER*n*_MAT1. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT1. 0b - Disable 1b - Enable
0	PWM Mode Enable for Channel 0

Table continues on the next page...

Table continued from the previous page...

Field	Function
PWMEN0	Enables PWM mode for channel 0. If this field is 0, EM0 controls CTIMER*n*_MAT0. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT0. 0b - Disable 1b - Enable

62.7.14 Match Shadow (MSR0 - MSR3)

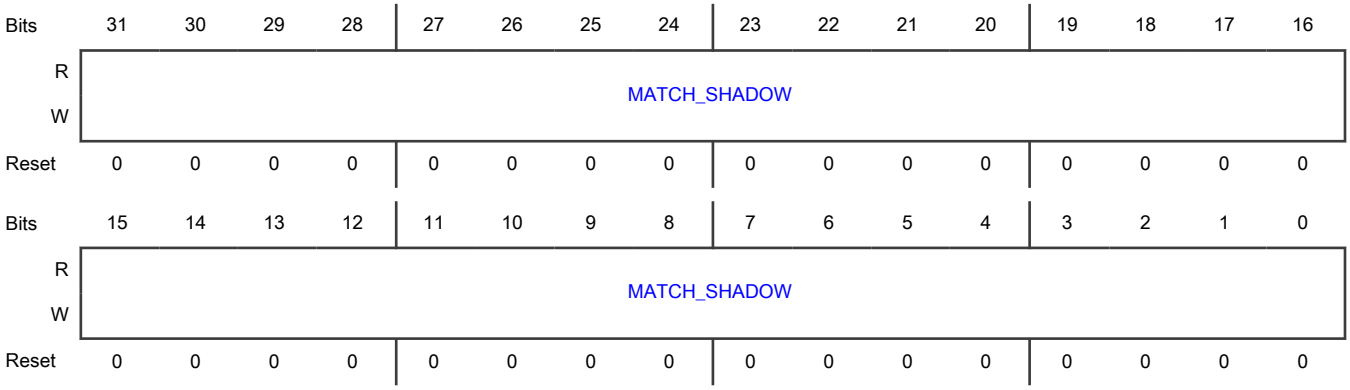
Offset

Register	Offset
MSR0	78h
MSR1	7Ch
MSR2	80h
MSR3	84h

Function

Contains the value that the corresponding Match (MR0 - MR3) are (optionally) reloaded with at the start of each new counter cycle. Typically, the match that causes the counter to be reset (and instigates the match reload) is also programmed to generate an interrupt or DMA request. Software or the DMA engine then has one full counter cycle to modify the contents of this register before the next reload occurs.

Diagram



Fields

Field	Function
31-0 MATCH_SHADOW	Timer Counter Match Shadow Value Reloads automatically with the contents of this register whenever Timer Counter (TC) is reset to zero.

Chapter 63
OS Event Timer (OSTIMER)

63.1 Chip-specific OSTIMER information

Table 594. Reference links to related information

Topic	Related module	Reference
Full description	OSTIMER	OSTIMER
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

63.1.1 Module instances

This device has one instance of the OSTIMER module, which has four interfaces to the cores:

Table 595. OSTIMER instances

	OSTIMER_CPU0	OSTIMER_HiFi4	OSTIMER_CPU1	OSTIMER_HiFi1
Located in	VDD1_SENSE	VDD1_SENSE	VDD1_SENSE	VDD1_SENSE
Access	CPU0, eDMA0, eDMA1	HiFi4, eDMA0, eDMA1	CPU1, eDMA2, eDMA3	HiFi1, eDMA2, eDMA3
Interface to	CPU0	HiFi4	CPU1	HiFi1

63.1.2 Gray to binary converter

SYSCON can configure gray to binary converter

Table 596. Gray to binary converter configuration

Gray to binary converter	SYSCON0 and SYSCON3 registers
Gray code	GRAY_CODE_LSB and GRAY_CODE_MSB
Binary code	BINARY_CODE_LSB and BINARY_CODE_MSB

63.2 Overview

OSTIMER includes a shared 64-bit timer and separate 64-bit match and capture functions for the Cortex-M33 core and DSP CPUs.

63.2.1 Block diagram

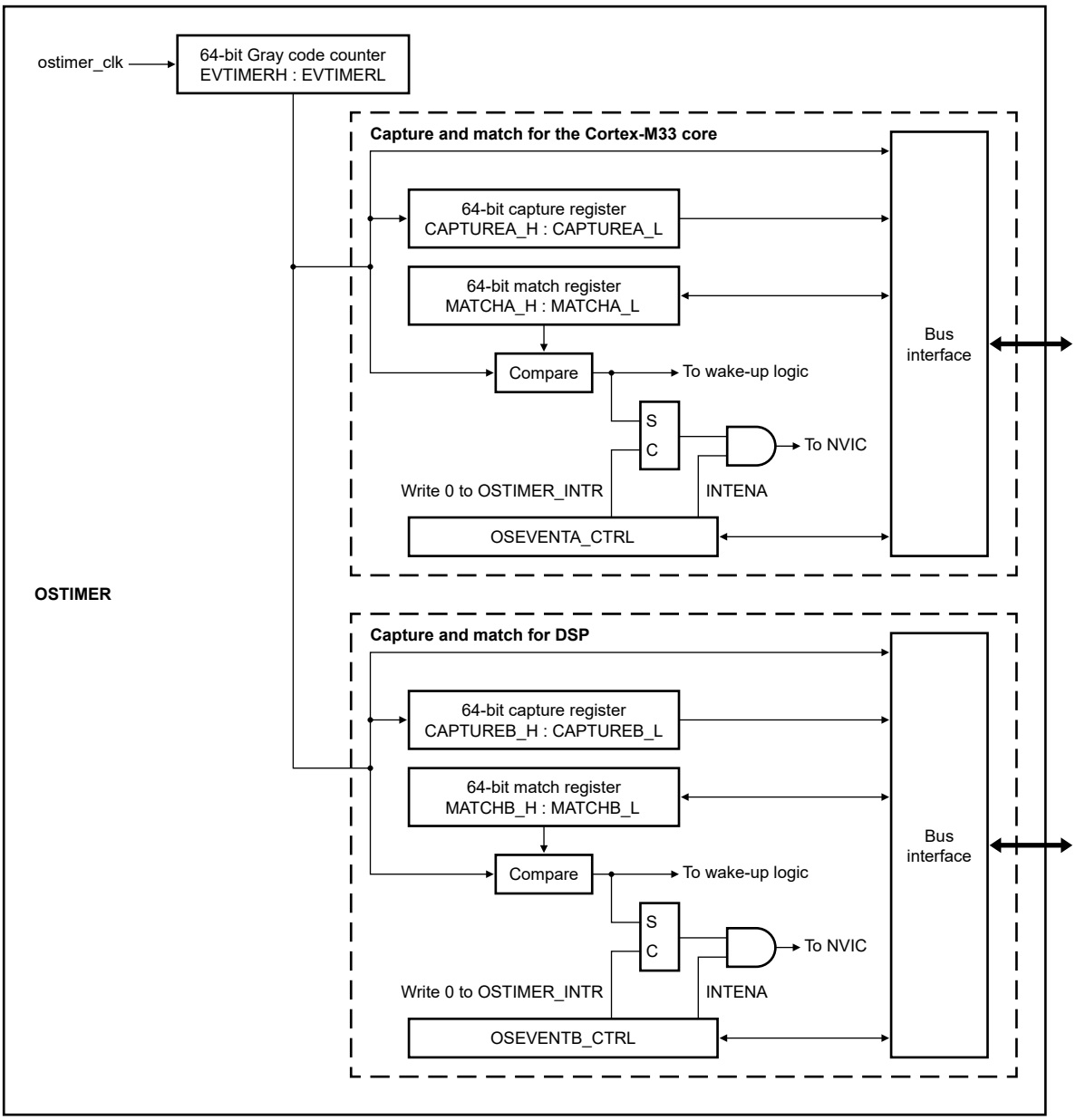


Figure 262. Block diagram

63.2.2 Features

- Provides a 64-bit Gray code counter (known as central EVTimer). Using Gray code means that the timer can run at a frequency unrelated to any CPU clock and can still be read by a CPU without a synchronization delay. Gray code is a reflected binary code that changes in a single-bit position for each increment.
- Supports separate functions for each CPU:
  - A capture register can copy the main counter value when triggered by a CPU request.
  - A match register can be compared to the main counter and can optionally generate an interrupt or wake-up event.

63.3 Functional description

The following sections describe the functional details of this module.

63.3.1 Shared event or timestamp timer

The 64-bit shared EVTimer initializes after chip power-up and then counts up continuously. The typical clock for this timer is the 1 MHz low-power oscillator.

This timer is implemented as a Gray code counter that enables the counter to be read asynchronously by any of the processing domains or captured by a capture register running on an asynchronous clock. The output of this shared EVTimer is connected to the processor-specific submodules.

63.3.2 CPU-specific match, capture, and interrupt generation

The submodules associated with each CPU include:

- A separate bus interface.
- 64 bits of capture values, available in CAPTURE_L and CAPTURE_H.
- 64 bits of match values, available in MATCH_H and MATCH_L.
- A control register, which includes an interrupt request flag and an interrupt enable field.

Capture registers	64 bits of capture values are available in CAPTURE_L and CAPTURE_H for each CPU. When the CPU issues a capture command, capture registers store the current value of the central EVTimer.
Match registers and interrupt request	64 bits of match values are available in MATCH_H and MATCH_L for each bus interface. The EVTimer output is compared against this combined value for interrupt or wake-up generation. A match to this register pair sets OSEVENT_CTRL[OSTIMER_INTRFLAG], which you can enable to generate an interrupt or wake-up request. You must write values to the match registers in Gray code. OSEVENT_CTRL[MATCH_WR_RDY] must be 0 before you write to the match registers. When you write a new value to the match registers, you must write to MATCH_L before the write to MATCH_H.

63.3.3 Clocking

The OSTIMER uses three clock signals: hclk, capture_clk, and ostimer_clk.

1. Hclk is the IPS interface clock for both the Arm and DSP sub-module, being used in all the registers and write/read operations.
2. Ostimer_clk is the counter clock, being used in the timer itself and in the intr_wakeup signals. It is expected to be running in a low frequency (~1MHz).
3. capture_clk is the Arm and DSP clock, it is used to register the capture signal “capture_load”.

63.3.4 Reset

For OSTIMER, the reset function is more complicated as compared to most other functions.

- Only the POR and software reset can reset the shared EVTimer. "Software reset" refers to a complete reset of the module.
- The capture and match registers are reset by a system reset. "System reset" refers to a reset from the RESET*n* pin, SYSRESETREQ from the CPU, a POR, or any other internal low-voltage resets.
- The control register (OSEVENT_CTL) is reset by the system reset or by software reset. However, the MATCH_WR_RDY bit can only be reset by system reset.

The following table shows a list of resets for OSTIMER registers.

Table 597. OSTIMER registers and their corresponding resets

Registers	POR	Software reset
EVTIMERL and EVTIMERH	Yes	Yes
CAPTURE_L and CAPTURE_H	Yes	No
OSEVENT_CTRL	Yes	Yes

63.3.5 Interrupts

If the interrupt request is enabled (i.e., OSEVENT_CTRL[OSTIMER_INTENA]= 1), the OSTIMER will generate interrupt when the central 64-bit EVTimer and the value programmed in MATCH_L and MATCH_H matches.

No interrupt generates if the interrupt request is disabled (i.e., OSEVENT_CTRL[OSTIMER_INTENA]= 0).

63.4 External signals

This module has no external signals.

63.5 Initialization

Perform this procedure to initialize OSTIMER:

1. Enable the OSTIMER clock. This enables the register interface and the peripheral function clock.
2. Select a clock source for OSTIMER.
3. Clear the OSTIMER peripheral reset.
4. Enable the interrupt that OSTIMER provides to the interrupt controller (set OSEVENT_CTRL[OSTIMER_INTRFLAG]= 1). This allows the chip to wake up from Deep Sleep mode. You can connect this interrupt to DSP alternatively.

63.6 Memory map and register definition

This section includes the module's memory map and detailed descriptions of all registers.

63.6.1 OSTIMER register descriptions

63.6.1.1 OSTIMER memory map

OSTIMER_CPU0 base address: 4020_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	EVTIMER Low (EVTIMERL)	32	R	0000_0000h
4h	EVTIMER High (EVTIMERH)	32	R	0000_0000h
8h	Local Capture Low for CPU (CAPTURE_L)	32	R	0000_0000h
Ch	Local Capture High for CPU (CAPTURE_H)	32	R	0000_0000h
10h	Local Match Low for CPU (MATCH_L)	32	RW	FFFF_FFFFh
14h	Local Match High for CPU (MATCH_H)	32	RW	FFFF_FFFFh
1Ch	OSTIMER Control for CPU (OSEVENT_CTRL)	32	RW	0000_0008h

63.6.1.2 EVTIMER Low (EVTIMERL)

Offset

Register	Offset
EVTIMERL	0h

Function

Contains the immediate value of the central EVTimer. This is a Gray encoded counter register that is not synchronized with the bus clock.

Because two 32-bit reads are required to retrieve the entire counter value, it is possible for the counter to roll over between the reads of EVTIMERL and EVTIMERH. Gray encoding ensures that even if this roll over occurs, the value read represents either the counter value immediately preceding or immediately following the rollover, if the bus clock rate is at least twice the module's clock rate (nominally 1 MHz), or if the bus is clocked by the same 1 MHz LP oscillator that provides the clock to the module.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 EVTIMER_COUNT_VALUE	EVTimer Count Value Specifies the current value of the lower 32 bits of the central 64-bit EVTimer when you read this register. **NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.1.3 EVTIMER High (EVTIMERH)

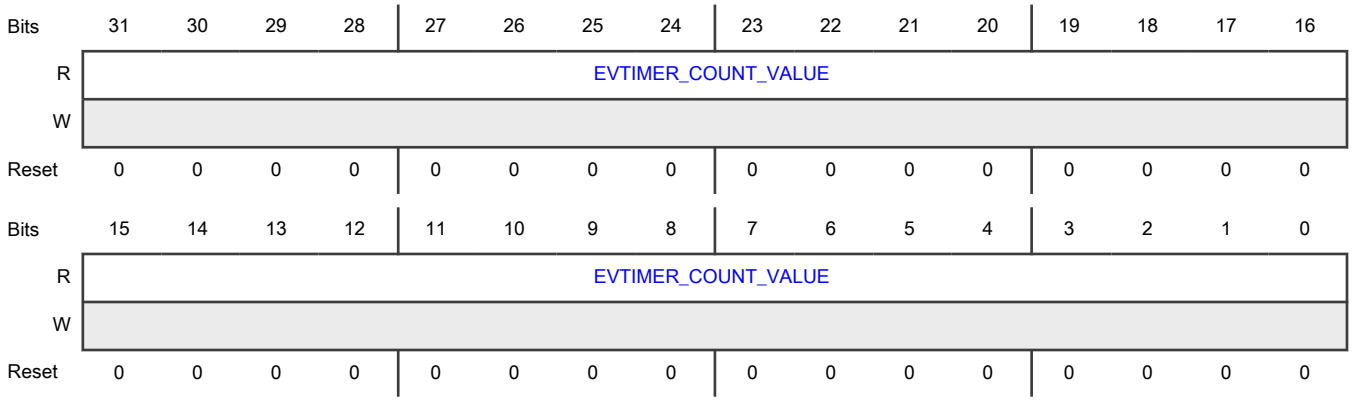
Offset

Register	Offset
EVTIMERH	4h

Function

Contains the higher value of the central EVTimer.

Diagram



Fields

Field	Function
31-0 EVTIMER_COUNT_VALUE	EVTimer Count Value Specifies the current value of the upper 32 bits of the central 64-bit EVTimer when you read this register. **NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.1.4 Local Capture Low for CPU (CAPTURE_L)

Offset

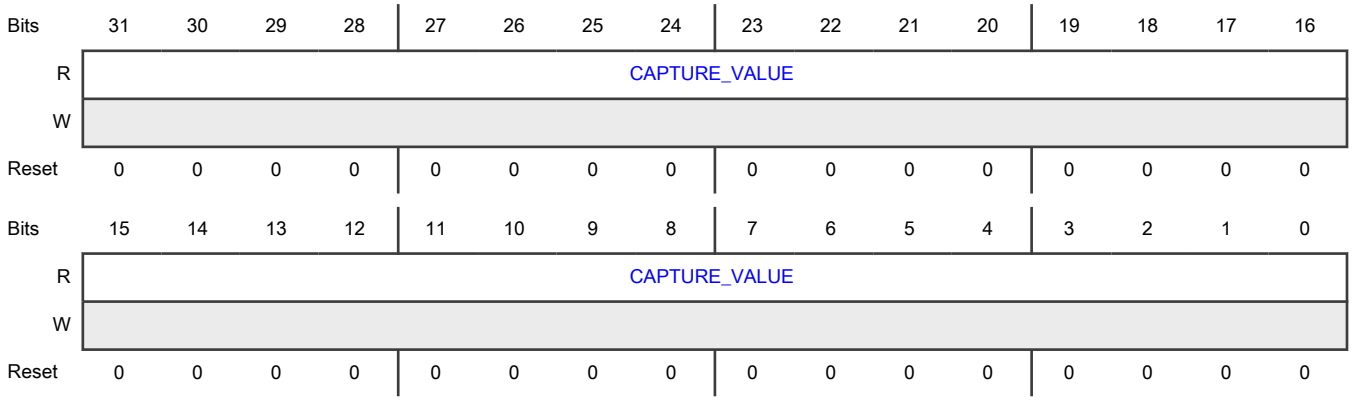
Register	Offset
CAPTURE_L	8h

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you should insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram



Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the lower 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU. Each CPU reads its own capture value at the same pair of addresses.

63.6.1.5 Local Capture High for CPU (CAPTURE_H)

Offset

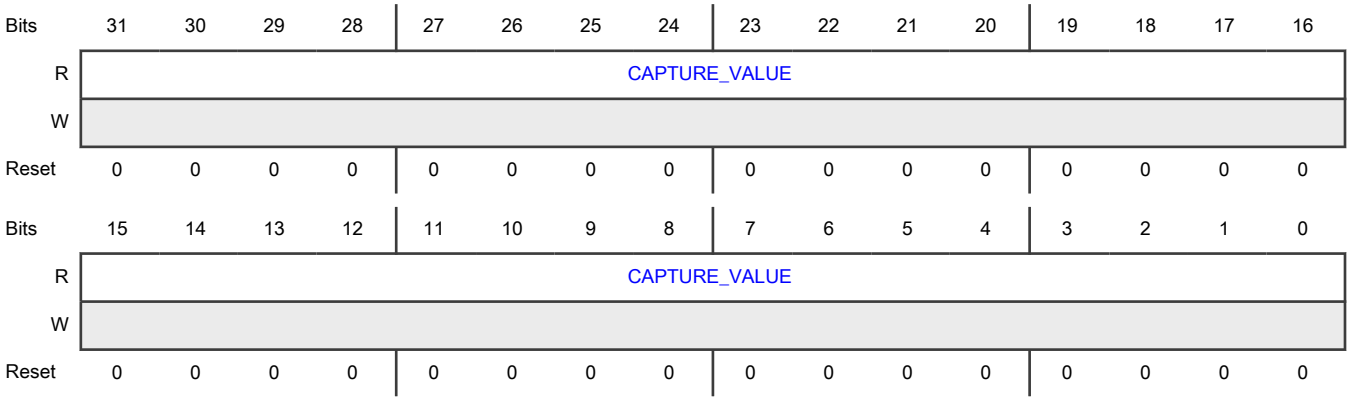
Register	Offset
CAPTURE_H	Ch

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you must insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram



Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the upper 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU reading its own capture value at the same pair of addresses.

63.6.1.6 Local Match Low for CPU (MATCH_L)

Offset

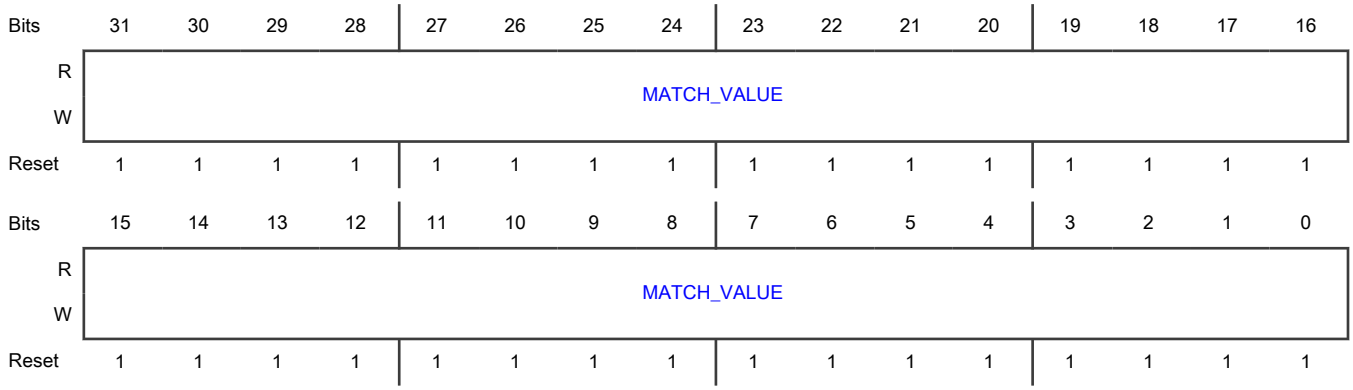
Register	Offset
MATCH_L	10h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (lower 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.1.7 Local Match High for CPU (MATCH_H)

Offset

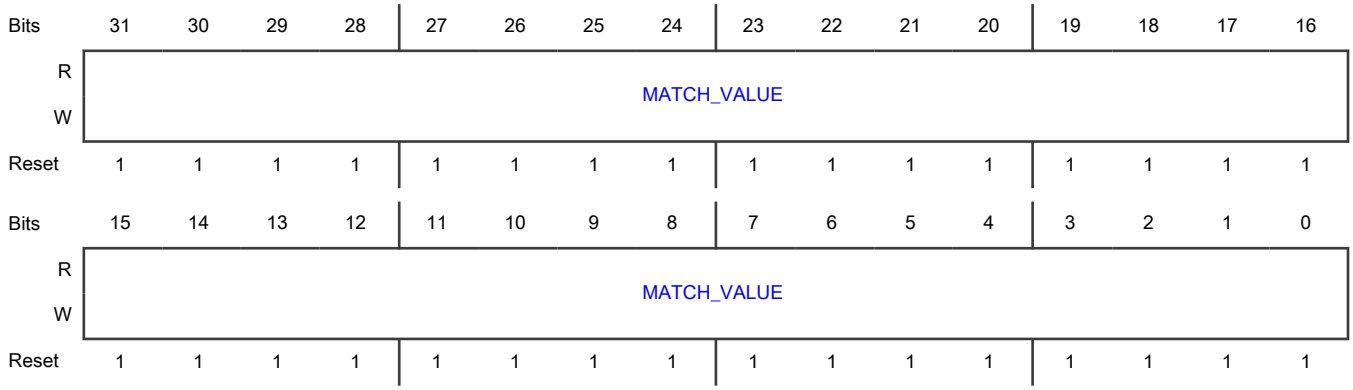
Register	Offset
MATCH_H	14h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (upper 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.1.8 OSTIMER Control for CPU (OSEVENT_CTRL)

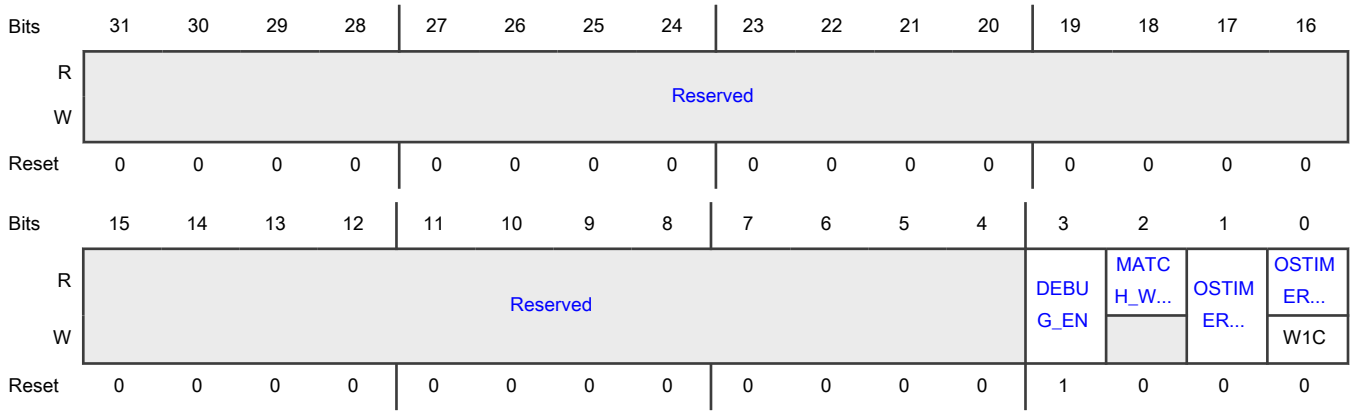
Offset

Register	Offset
OSEVENT_CTRL	1Ch

Function

Provides the interrupt flag and interrupt enable signals for each CPU. A separate OSEVENT_CTRL register is implemented for each CPU. Each CPU reads its own local value at the same address.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 DEBUG_EN	Debug Enable Enables OSTIMER to operate when the chip is in Debug mode. 0b - Disables 1b - Enables
2 MATCH_WR_RDY	EVTimer Match Write Ready Specifies that when this field is low, it is safe to reload (write) to the match registers. In typical applications, it is not necessary to read this field. The 64-bit match register value is transferred from a pair of shadow registers to the active match registers after you write to MATCH_H. You must not initiate a second write to MATCH_L and MATCH_H until after this transfer completes. This field becomes 0 after the transfer completes. It is not necessary to read this status field, if an interrupt has already occurred because of the first match value, or if it is certain (via some other means) that the required period of time (3 bus clocks) has elapsed.
1 OSTIMER_INTENA	Interrupt or Wake-Up Request This bit is to enable/disable interrupt request. If the value is 1, then this field asserts an interrupt or wake-up request to the domain processor. Otherwise, the interrupt or wake-up requests are blocked. 0b - Interrupts blocked 1b - Interrupts enabled
0 OSTIMER_INTRFLAG	Interrupt Flag Sets an interrupt flag when a match occurs between the central 64-bit EVTimer and the value programmed in the MATCH_L and MATCH_H register pair for the associated CPU. Writes to clear this field are asynchronous. You must write 1 to this field before writing a new match value into MATCH_L and MATCH_H.

63.6.2 OSTIMER register descriptions

63.6.2.1 OSTIMER memory map

OSTIMER_HiFi4 base address: 4020_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	EVTIMER Low (EVTIMERL)	32	R	0000_0000h
4h	EVTIMER High (EVTIMERH)	32	R	0000_0000h
8h	Local Capture Low for CPU (CAPTURE_L)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Local Capture High for CPU (CAPTURE_H)	32	R	0000_0000h
10h	Local Match Low for CPU (MATCH_L)	32	RW	FFFF_FFFFh
14h	Local Match High for CPU (MATCH_H)	32	RW	FFFF_FFFFh
1Ch	OSTIMER Control for CPU (OSEVENT_CTRL)	32	RW	0000_0008h

63.6.2.2 EVTIMER Low (EVTIMERL)

Offset

Register	Offset
EVTIMERL	0h

Function

Contains the immediate value of the central EVTimer. This is a Gray encoded counter register that is not synchronized with the bus clock.

Because two 32-bit reads are required to retrieve the entire counter value, it is possible for the counter to roll over between the reads of EVTIMERL and EVTIMERH. Gray encoding ensures that even if this roll over occurs, the value read represents either the counter value immediately preceding or immediately following the rollover, if the bus clock rate is at least twice the module's clock rate (nominally 1 MHz), or if the bus is clocked by the same 1 MHz LP oscillator that provides the clock to the module.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0	EVTimer Count Value Specifies the current value of the lower 32 bits of the central 64-bit EVTimer when you read this register.

Table continues on the next page...

Field	Function
EVTIMER_COUNT_VALUE	**NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.2.3 EVTIMER High (EVTIMERH)

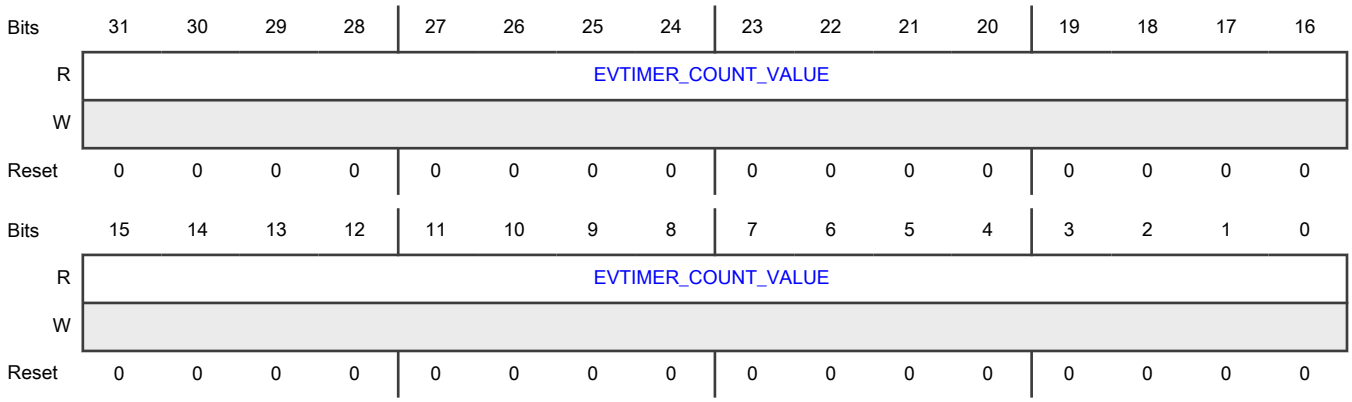
Offset

Register	Offset
EVTIMERH	4h

Function

Contains the higher value of the central EVTimer.

Diagram



Fields

Field	Function
31-0 EVTIMER_COUNT_VALUE	EVTimer Count Value Specifies the current value of the upper 32 bits of the central 64-bit EVTimer when you read this register. **NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.2.4 Local Capture Low for CPU (CAPTURE_L)

Offset

Register	Offset
CAPTURE_L	8h

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you should insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the lower 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU. Each CPU reads its own capture value at the same pair of addresses.

63.6.2.5 Local Capture High for CPU (CAPTURE_H)

Offset

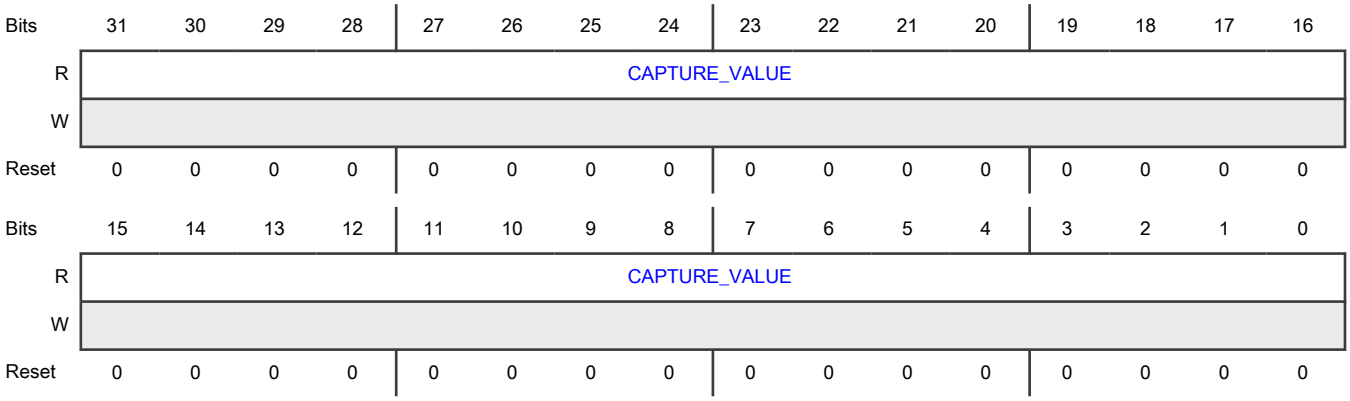
Register	Offset
CAPTURE_H	Ch

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you must insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram



Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the upper 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU reading its own capture value at the same pair of addresses.

63.6.2.6 Local Match Low for CPU (MATCH_L)

Offset

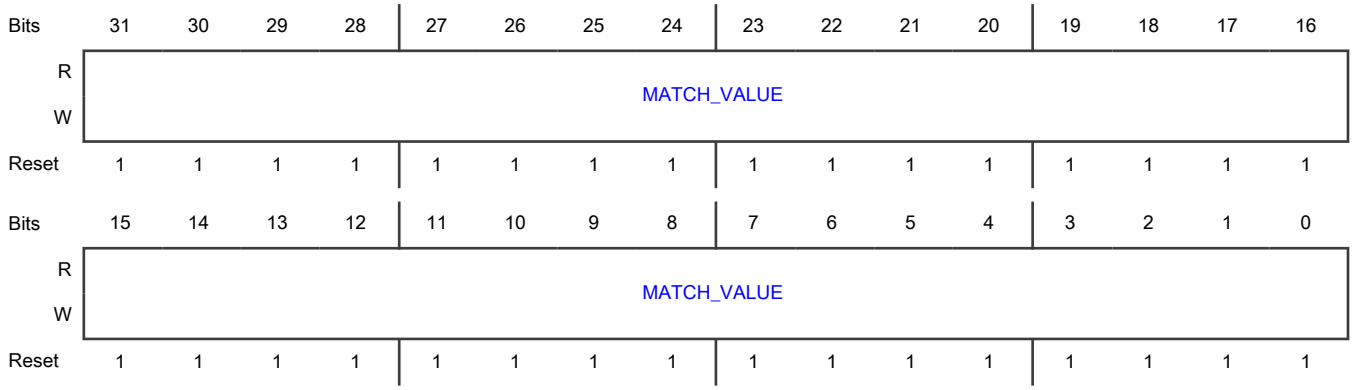
Register	Offset
MATCH_L	10h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (lower 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.2.7 Local Match High for CPU (MATCH_H)

Offset

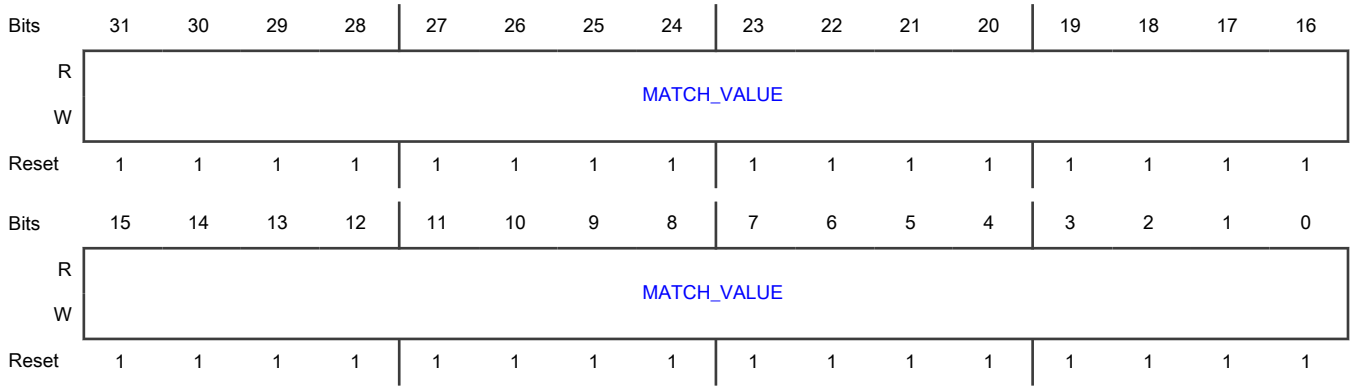
Register	Offset
MATCH_H	14h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (upper 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.2.8 OSTIMER Control for CPU (OSEVENT_CTRL)

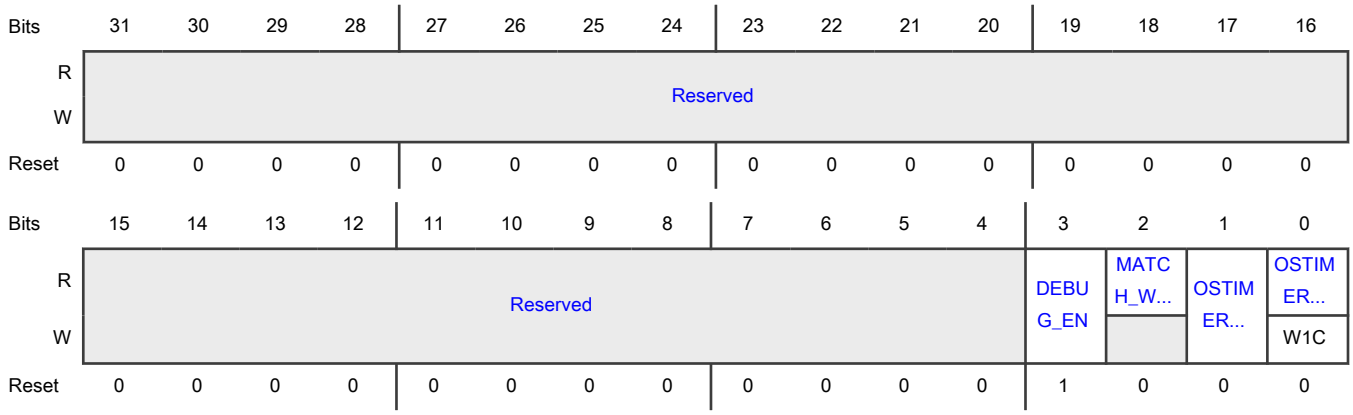
Offset

Register	Offset
OSEVENT_CTRL	1Ch

Function

Provides the interrupt flag and interrupt enable signals for each CPU. A separate OSEVENT_CTRL register is implemented for each CPU. Each CPU reads its own local value at the same address.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 DEBUG_EN	Debug Enable Enables OSTIMER to operate when the chip is in Debug mode. 0b - Disables 1b - Enables
2 MATCH_WR_RDY	EVTimer Match Write Ready Specifies that when this field is low, it is safe to reload (write) to the match registers. In typical applications, it is not necessary to read this field. The 64-bit match register value is transferred from a pair of shadow registers to the active match registers after you write to MATCH_H. You must not initiate a second write to MATCH_L and MATCH_H until after this transfer completes. This field becomes 0 after the transfer completes. It is not necessary to read this status field, if an interrupt has already occurred because of the first match value, or if it is certain (via some other means) that the required period of time (3 bus clocks) has elapsed.
1 OSTIMER_INTENA	Interrupt or Wake-Up Request This bit is to enable/disable interrupt request. If the value is 1, then this field asserts an interrupt or wake-up request to the domain processor. Otherwise, the interrupt or wake-up requests are blocked. 0b - Interrupts blocked 1b - Interrupts enabled
0 OSTIMER_INTRFLAG	Interrupt Flag Sets an interrupt flag when a match occurs between the central 64-bit EVTimer and the value programmed in the MATCH_L and MATCH_H register pair for the associated CPU. Writes to clear this field are asynchronous. You must write 1 to this field before writing a new match value into MATCH_L and MATCH_H.

63.6.3 OSTIMER register descriptions

63.6.3.1 OSTIMER memory map

OSTIMER_CPU1 base address: 4020_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	EVTIMER Low (EVTIMERL)	32	R	0000_0000h
4h	EVTIMER High (EVTIMERH)	32	R	0000_0000h
8h	Local Capture Low for CPU (CAPTURE_L)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Local Capture High for CPU (CAPTURE_H)	32	R	0000_0000h
10h	Local Match Low for CPU (MATCH_L)	32	RW	FFFF_FFFFh
14h	Local Match High for CPU (MATCH_H)	32	RW	FFFF_FFFFh
1Ch	OSTIMER Control for CPU (OSEVENT_CTRL)	32	RW	0000_0008h

63.6.3.2 EVTIMER Low (EVTIMERL)

Offset

Register	Offset
EVTIMERL	0h

Function

Contains the immediate value of the central EVTimer. This is a Gray encoded counter register that is not synchronized with the bus clock.

Because two 32-bit reads are required to retrieve the entire counter value, it is possible for the counter to roll over between the reads of EVTIMERL and EVTIMERH. Gray encoding ensures that even if this roll over occurs, the value read represents either the counter value immediately preceding or immediately following the rollover, if the bus clock rate is at least twice the module's clock rate (nominally 1 MHz), or if the bus is clocked by the same 1 MHz LP oscillator that provides the clock to the module.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVTIMER_COUNT_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0	EVTimer Count Value Specifies the current value of the lower 32 bits of the central 64-bit EVTimer when you read this register.

Table continues on the next page...

Field	Function
EVTIMER_COUNT_VALUE	**NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.3.3 EVTIMER High (EVTIMERH)

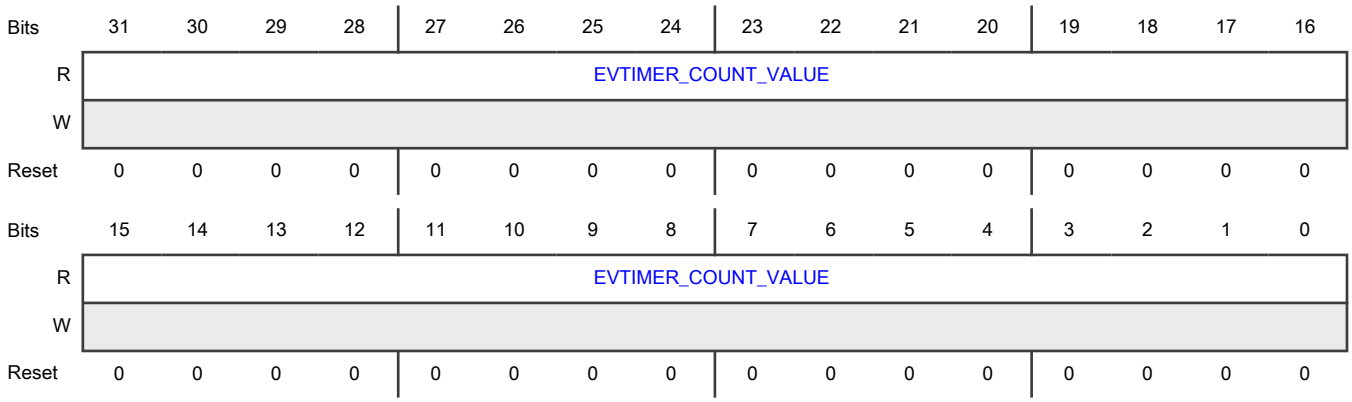
Offset

Register	Offset
EVTIMERH	4h

Function

Contains the higher value of the central EVTimer.

Diagram



Fields

Field	Function
31-0 EVTIMER_COUNT_VALUE	EVTimer Count Value Specifies the current value of the upper 32 bits of the central 64-bit EVTimer when you read this register. **NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.3.4 Local Capture Low for CPU (CAPTURE_L)

Offset

Register	Offset
CAPTURE_L	8h

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you should insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the lower 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU. Each CPU reads its own capture value at the same pair of addresses.

63.6.3.5 Local Capture High for CPU (CAPTURE_H)

Offset

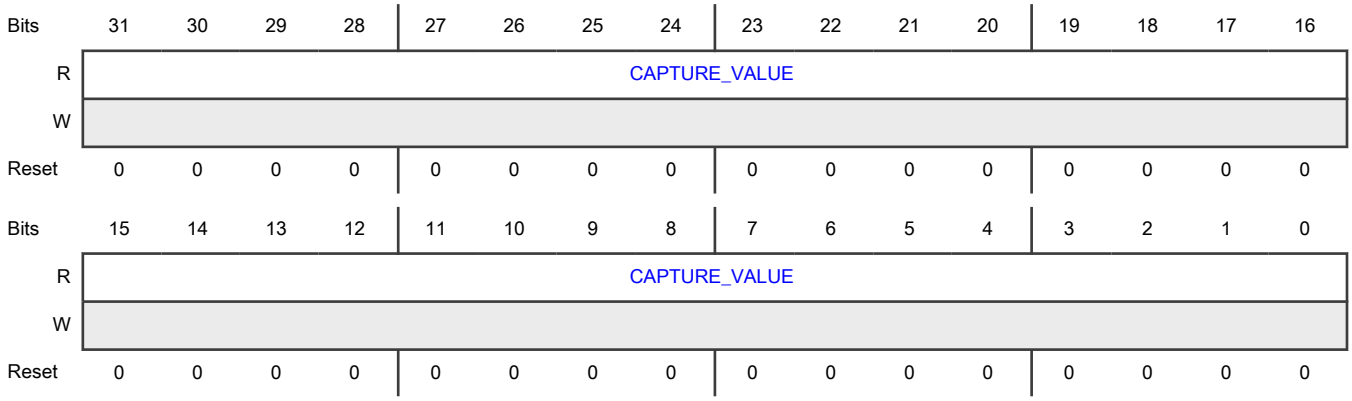
Register	Offset
CAPTURE_H	Ch

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you must insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram



Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the upper 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU reading its own capture value at the same pair of addresses.

63.6.3.6 Local Match Low for CPU (MATCH_L)

Offset

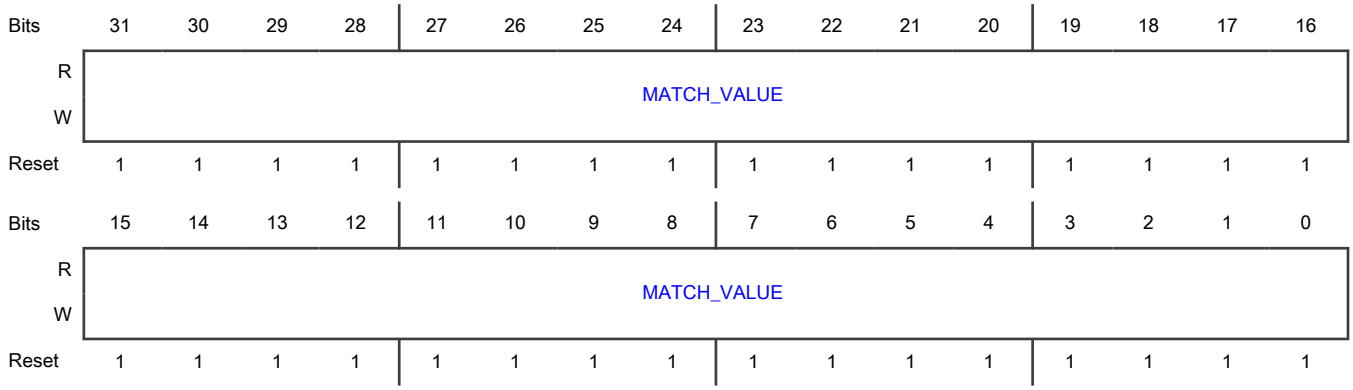
Register	Offset
MATCH_L	10h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (lower 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.3.7 Local Match High for CPU (MATCH_H)

Offset

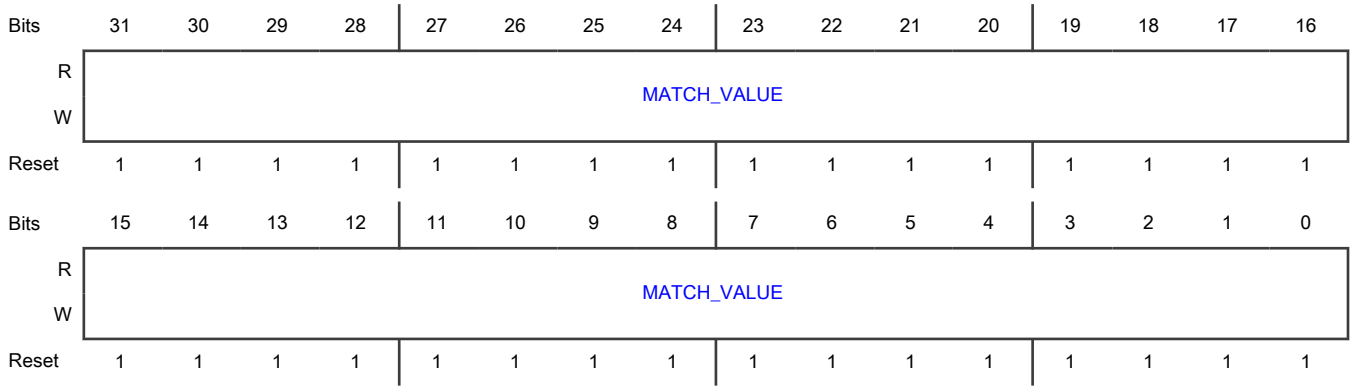
Register	Offset
MATCH_H	14h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (upper 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.3.8 OSTIMER Control for CPU (OSEVENT_CTRL)

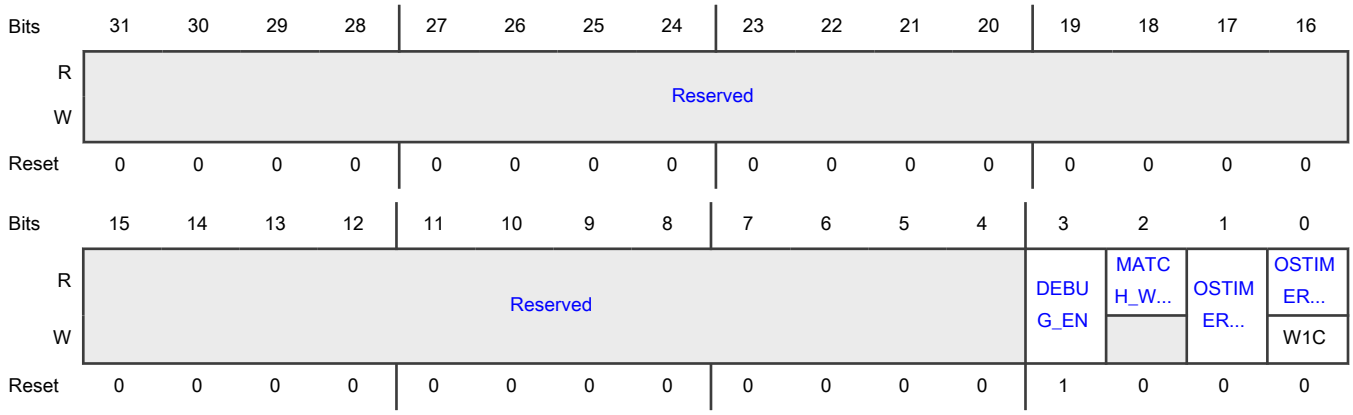
Offset

Register	Offset
OSEVENT_CTRL	1Ch

Function

Provides the interrupt flag and interrupt enable signals for each CPU. A separate OSEVENT_CTRL register is implemented for each CPU. Each CPU reads its own local value at the same address.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 DEBUG_EN	Debug Enable Enables OSTIMER to operate when the chip is in Debug mode. 0b - Disables 1b - Enables
2 MATCH_WR_RDY	EVTimer Match Write Ready Specifies that when this field is low, it is safe to reload (write) to the match registers. In typical applications, it is not necessary to read this field. The 64-bit match register value is transferred from a pair of shadow registers to the active match registers after you write to MATCH_H. You must not initiate a second write to MATCH_L and MATCH_H until after this transfer completes. This field becomes 0 after the transfer completes. It is not necessary to read this status field, if an interrupt has already occurred because of the first match value, or if it is certain (via some other means) that the required period of time (3 bus clocks) has elapsed.
1 OSTIMER_INTENA	Interrupt or Wake-Up Request This bit is to enable/disable interrupt request. If the value is 1, then this field asserts an interrupt or wake-up request to the domain processor. Otherwise, the interrupt or wake-up requests are blocked. 0b - Interrupts blocked 1b - Interrupts enabled
0 OSTIMER_INTRFLAG	Interrupt Flag Sets an interrupt flag when a match occurs between the central 64-bit EVTimer and the value programmed in the MATCH_L and MATCH_H register pair for the associated CPU. Writes to clear this field are asynchronous. You must write 1 to this field before writing a new match value into MATCH_L and MATCH_H.

63.6.4 OSTIMER register descriptions

63.6.4.1 OSTIMER memory map

OSTIMER_HiFi1 base address: 4020_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h	EVTIMER Low (EVTIMERL)	32	R	0000_0000h
4h	EVTIMER High (EVTIMERH)	32	R	0000_0000h
8h	Local Capture Low for CPU (CAPTURE_L)	32	R	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Local Capture High for CPU (CAPTURE_H)	32	R	0000_0000h
10h	Local Match Low for CPU (MATCH_L)	32	RW	FFFF_FFFFh
14h	Local Match High for CPU (MATCH_H)	32	RW	FFFF_FFFFh
1Ch	OSTIMER Control for CPU (OSEVENT_CTRL)	32	RW	0000_0008h

63.6.4.2 EVTIMER Low (EVTIMERL)

Offset

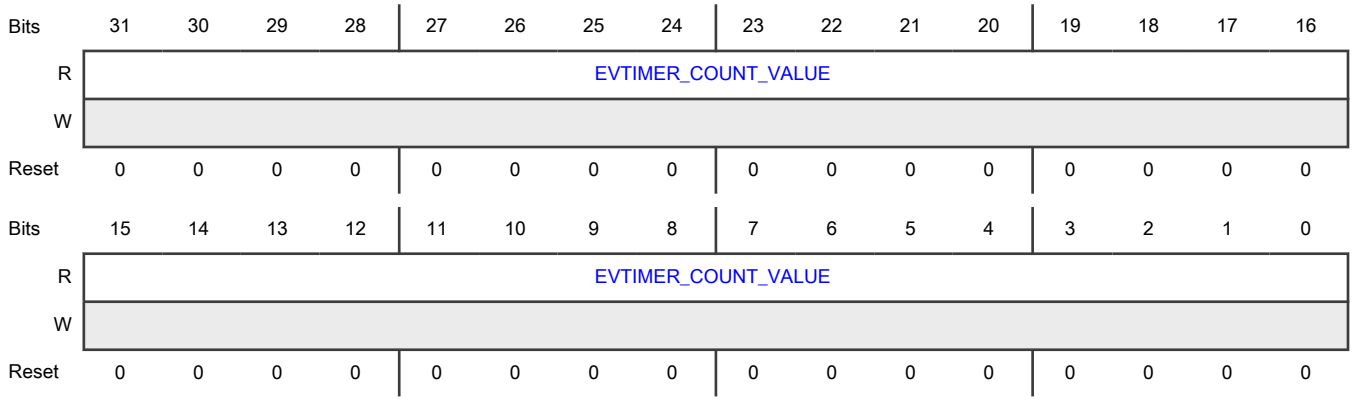
Register	Offset
EVTIMERL	0h

Function

Contains the immediate value of the central EVTimer. This is a Gray encoded counter register that is not synchronized with the bus clock.

Because two 32-bit reads are required to retrieve the entire counter value, it is possible for the counter to roll over between the reads of EVTIMERL and EVTIMERH. Gray encoding ensures that even if this roll over occurs, the value read represents either the counter value immediately preceding or immediately following the rollover, if the bus clock rate is at least twice the module's clock rate (nominally 1 MHz), or if the bus is clocked by the same 1 MHz LP oscillator that provides the clock to the module.

Diagram



Fields

Field	Function
31-0	EVTimer Count Value Specifies the current value of the lower 32 bits of the central 64-bit EVTimer when you read this register.

Table continues on the next page...

Field	Function
EVTIMER_COUNT_VALUE	**NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.4.3 EVTIMER High (EVTIMERH)

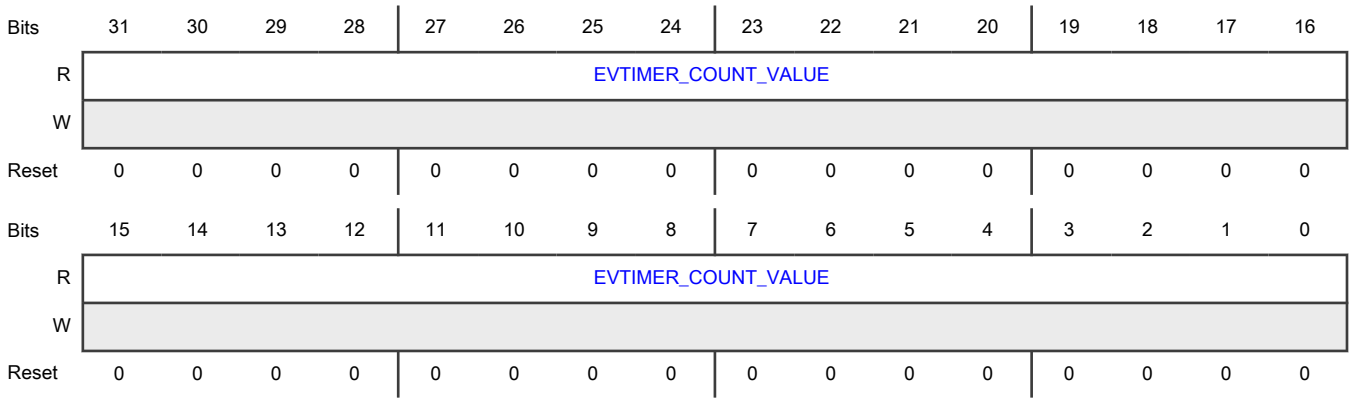
Offset

Register	Offset
EVTIMERH	4h

Function

Contains the higher value of the central EVTimer.

Diagram



Fields

Field	Function
31-0 EVTIMER_COUNT_VALUE	EVTimer Count Value Specifies the current value of the upper 32 bits of the central 64-bit EVTimer when you read this register. **NOTE** Only one physical EVTimer exists and is readable from all domains.

63.6.4.4 Local Capture Low for CPU (CAPTURE_L)

Offset

Register	Offset
CAPTURE_L	8h

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you should insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTURE_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the lower 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU. Each CPU reads its own capture value at the same pair of addresses.

63.6.4.5 Local Capture High for CPU (CAPTURE_H)

Offset

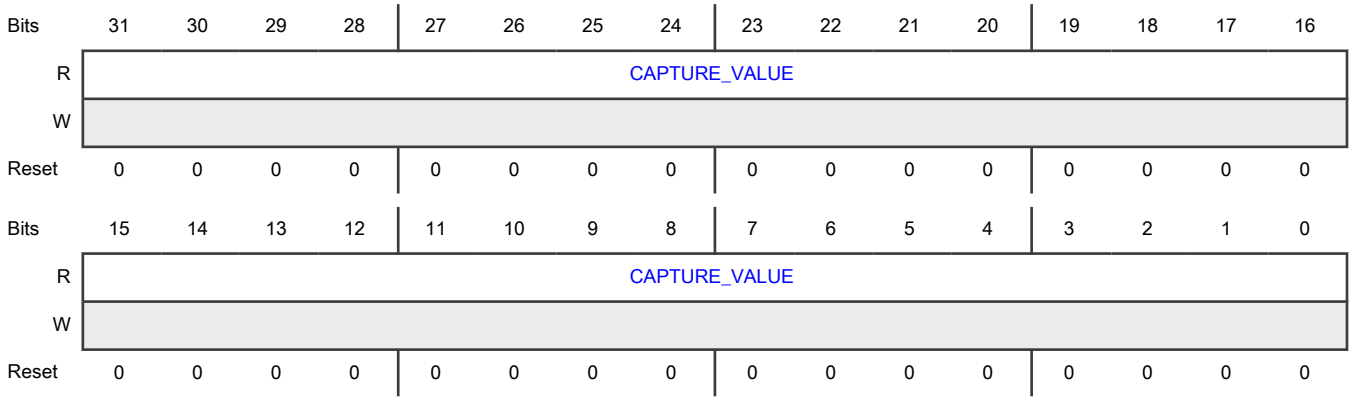
Register	Offset
CAPTURE_H	Ch

Function

Contains the value of the central EVTimer when captured by the CPU. This register is Gray encoded to match EVTimer. A load is triggered by a load signal from the CPU (for example, an "arm_txev" pulse from the Cortex-M33 core CPU or a "tie_expstate" pulse from DSP). You must not execute a new capture command between reading CAPTURE_L and CAPTURE_H, to avoid retrieving incoherent results.

For the instance associated with the DSP CPU: If a read of CAPTURE_L and CAPTURE_H closely follows the command to implement a capture, then you must insert the `#pragma no_reorder` directive in the DSP code (to direct the compiler to not initiate the read operation until the capture command has executed).

Diagram



Fields

Field	Function
31-0 CAPTURE_VALUE	EVTimer Capture Value Specifies the value of the upper 32 bits of the central 64-bit EVTimer when you read this register, at the time that the CPU generated the last capture signal. A separate pair of CAPTURE_L and CAPTURE_H is implemented for each CPU reading its own capture value at the same pair of addresses.

63.6.4.6 Local Match Low for CPU (MATCH_L)

Offset

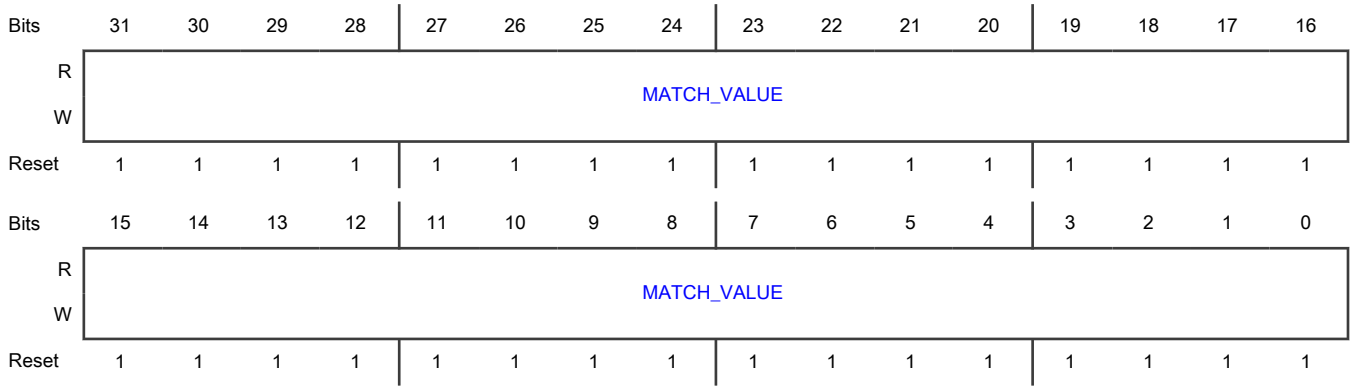
Register	Offset
MATCH_L	10h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (lower 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.4.7 Local Match High for CPU (MATCH_H)

Offset

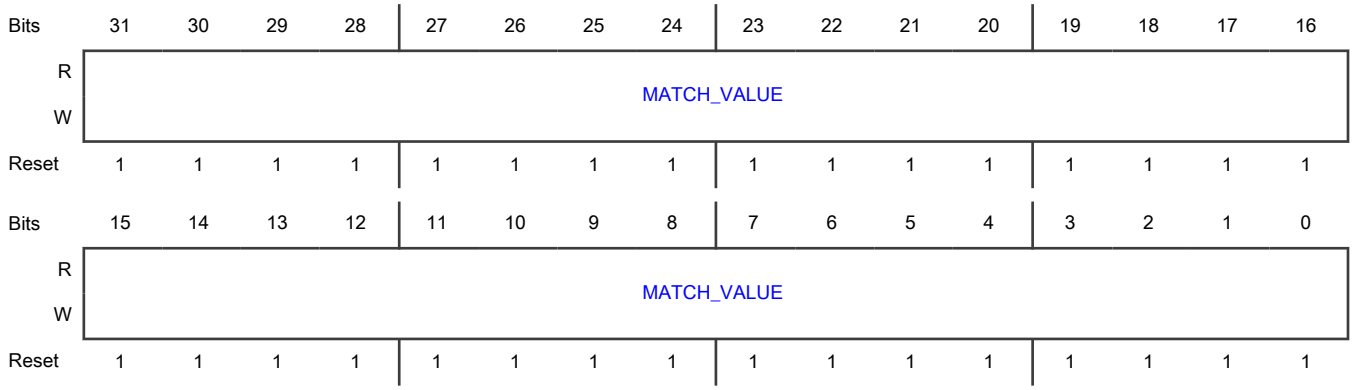
Register	Offset
MATCH_H	14h

Function

Compares the match value with the central EVTimer value. This register is Gray encoded to match EVTimer. The value in this pair of registers is stable, so there is no need for multiple reads. When writing to the match register pair, you must write to MATCH_L first, followed by the write to MATCH_H.

You must not initiate a second write to the match register pair until the first write is complete (which is a minimum of three bus clocks followed by a write to MATCH_H). OSEVENT_CTRL[MATCH_WR_RDY] indicates when it is safe to reload the match registers. You do not need to read OSEVENT_CTRL[MATCH_WR_RDY] if an interrupt has already occurred because of the previous match value, or the required time period has elapsed.

Diagram



Fields

Field	Function
31-0 MATCH_VALUE	EVTimer Match Value Compares the value (upper 32 bits) of the MATCH_L and MATCH_H register pair with the central EVTimer value. If both the values match, the CPU generates an interrupt request if the interrupt is enabled. A separate pair of MATCH_L and MATCH_H is implemented for each CPU reading its own local value at the same pair of addresses.

63.6.4.8 OSTIMER Control for CPU (OSEVENT_CTRL)

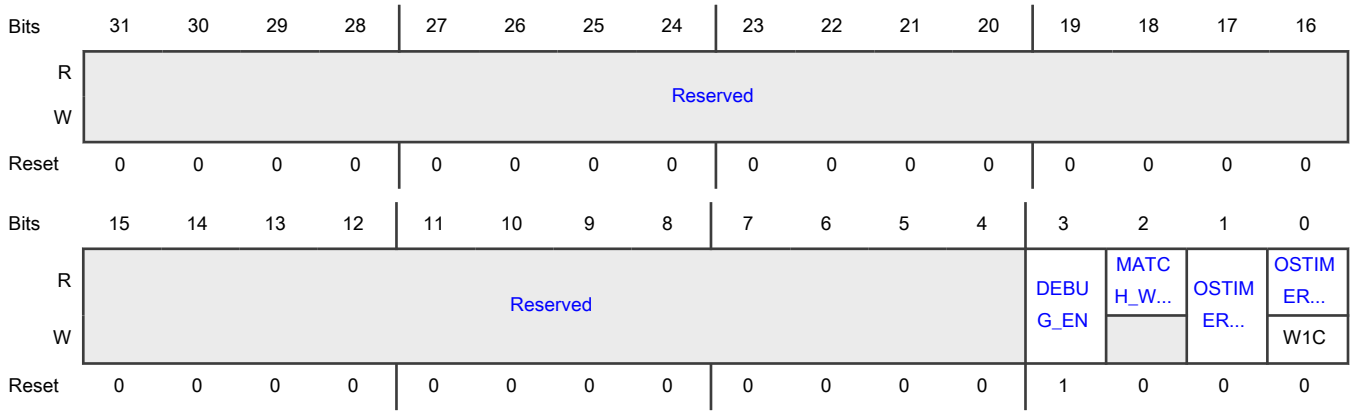
Offset

Register	Offset
OSEVENT_CTRL	1Ch

Function

Provides the interrupt flag and interrupt enable signals for each CPU. A separate OSEVENT_CTRL register is implemented for each CPU. Each CPU reads its own local value at the same address.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 DEBUG_EN	Debug Enable Enables OSTIMER to operate when the chip is in Debug mode. 0b - Disables 1b - Enables
2 MATCH_WR_RDY	EVTimer Match Write Ready Specifies that when this field is low, it is safe to reload (write) to the match registers. In typical applications, it is not necessary to read this field. The 64-bit match register value is transferred from a pair of shadow registers to the active match registers after you write to MATCH_H. You must not initiate a second write to MATCH_L and MATCH_H until after this transfer completes. This field becomes 0 after the transfer completes. It is not necessary to read this status field, if an interrupt has already occurred because of the first match value, or if it is certain (via some other means) that the required period of time (3 bus clocks) has elapsed.
1 OSTIMER_INTENA	Interrupt or Wake-Up Request This bit is to enable/disable interrupt request. If the value is 1, then this field asserts an interrupt or wake-up request to the domain processor. Otherwise, the interrupt or wake-up requests are blocked. 0b - Interrupts blocked 1b - Interrupts enabled
0 OSTIMER_INTRFLAG	Interrupt Flag Sets an interrupt flag when a match occurs between the central 64-bit EVTimer and the value programmed in the MATCH_L and MATCH_H register pair for the associated CPU. Writes to clear this field are asynchronous. You must write 1 to this field before writing a new match value into MATCH_L and MATCH_H.

Chapter 64
SCTimer (SCT)

64.1 Chip-specific SCT information

Table 598. Reference links to related information

Topic	Related module	Reference
Full description	SCT	SCT
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Interrupts	NVIC	Interrupts
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Interrupts and DMA Events	-	Interrupts and DMA events assignments INPUTMUX

64.1.1 Module instances

This device has one instance of the SCT module, SCT0.

Table 599. SCT instance

	SCT0
Located	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
System clock	clk_and_reset_hclk (COMPUTE_MAIN_CLK), see System clocks for more details.

64.1.2 SCT inputs

You can use CONFIG[CKSEL] to specify the input of SCT clock. The following table shows those inputs.

Table 600. SCT inputs

Input number	Input from
Input 0	Configured by INPUTMUX0_SCT0_TRIG0 register.
Input 1	Configured by INPUTMUX0_SCT0_TRIG1 register.
Input 2	Configured by INPUTMUX0_SCT0_TRIG2 register.

Table continues on the next page...

Table 600. SCT inputs (continued)

Input number	Input from
Input 3	Configured by INPUTMUX0_SCT0_TRIG3 register.
Input 4	Configured by INPUTMUX0_SCT0_TRIG4 register.
Input 5	Configured by INPUTMUX0_SCT0_TRIG5 register.
Input 6	Configured by INPUTMUX0_SCT0_TRIG6 register.
Input 7	clk_and_reset_pllclk (SCT_FCLK).

64.2 Overview

SCT is a powerful, flexible timer module capable of creating complex PWM waveforms and performing other advanced timing and control operations with minimal or no CPU intervention. It can operate as a single 32-bit counter or as two independent, 16-bit counters in Unidirectional or Bidirectional modes.

As with most timers, SCT supports:

- A selection of match registers against which you can compare the count value.
- Capture registers where you can record the current count value if a predefined condition is detected.

An additional feature contributing to the versatility of SCT is the concept of events. This module supports multiple separate events that you can define based on some combination of parameters, including all or some of the following:

- A match on one of the match registers
- A transition on one of the SCT inputs or outputs
- The direction of count

Every action that SCT performs occurs in direct response to the events that you define, without any software overhead. You can enable an event to:

- Start, stop, or halt the counter.
- Limit the counter, which refers to clearing the counter in Unidirectional mode, or changing the counter direction in Bidirectional mode.
- Set, clear, or toggle any SCT output.
- Force a capture of the count value into any of the capture registers.
- Generate an interrupt or DMA request.

SCT also allows you to group and filter events. A group of enabled and disabled events can be described as a state, and several states with different sets of enabled and disabled events are allowed. Changing from one state to another is event-driven and can therefore happen without your intervention. By defining these states, SCT provides the means to run state machines in hardware to accomplish complex waveform and timing tasks.

In a simple system such as a classical timer or counter with capture and match capabilities, all events that could cause the timer to capture the timer value or toggle a match output are enabled, and remain enabled, at all times while the counter is running. In this case, no events are filtered and the state that does not change describes the system. This is the default configuration for SCT.

In a more complex system, two states could be set up that allow some events in one state and not in the other. You can use an event that is enabled in both states to move from one state to the other and back while filtering out events in either of the states. In such a two-state system, you can create different waveforms at the SCT output, depending on the event history. Change of states is event-driven and happens without any CPU intervention.

Formally, you can program SCT as a state machine generator. With the unique ability to perform switching between groups of events, you can use SCT as a highly complex state machine engine. Events identify the occurrence of conditions that warrant state

changes and determine the next state to move to. This provides you with an extremely powerful control tool—particularly when SCT inputs and outputs are connected to other on-chip resources.

In addition to events and states, SCT provides other enhanced features:

- Four alternative clocking modes including a fully asynchronous mode
- Selection of any SCT input as a clock source or a clock gate
- Ability to define a fractional portion to certain match registers (fractional matching uses a “dither” approach to achieve average resolutions of up to 1/16th of a clock period when generating PWM edges)
- Ability to select a greater-than-or-equal-to match condition for the purpose of event generation

NOTE

In this chapter, the term "bus error" indicates an SCT response that makes the processor take an exception.

64.2.1 Block diagram

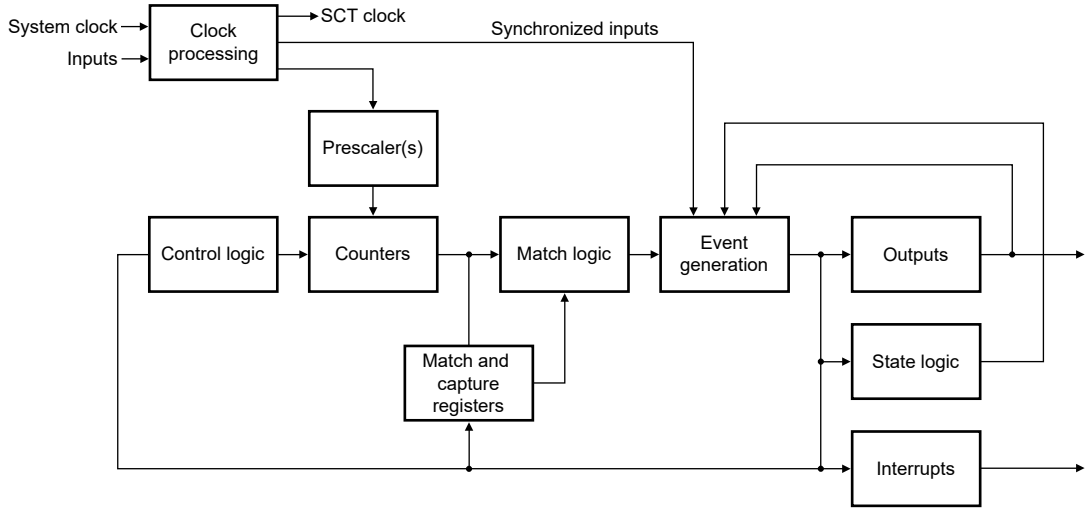


Figure 263. Block diagram

64.2.2 Features

SCT supports:

- 8 inputs
- 10 outputs
- 16 match and capture registers
- 16 events
- 32 states
- 6 dithers

Counter or timer features:

- Configurable as two 16-bit counters or one 32-bit counter (this configuration is possible for each SCT)
- Counters clocked by the system clock or a selected input
- Configurable as up counters or up-down counters
- Configurable number of match and capture registers

- Ability for the counter value to be loaded into capture registers, triggered by a match or I/O toggle
- Event creation possible for the following timer events for match and/or I/O transitions:
  - DMA request
  - Interrupt
  - Stop
  - Limit
  - Halt
  - Change counting direction
  - Toggle outputs
  - Change the state

PWM features:

- Ability to use counters with match registers to toggle outputs and create time-proportioned PWM signals (PWM waveforms can change based on the current state)
- Up to eight single-edge or four dual-edge PWM outputs with an independent duty cycle and a common PWM cycle length

Event creation features:

- Ability to enable events in Bidirectional mode, based on the count direction (the following conditions define an event):
  - A counter match condition
  - An input (or output) condition such as a rising or falling edge or level
  - A combination of match and/or I/O conditions
- Ability for selected events to limit, halt, start, or stop a counter or change direction
- Ability for events to trigger state changes, output transitions, timer captures, interrupts, and DMA transactions
- Ability to use match register 0 as an automatic limit
- Ability to hold match events until another qualifying event occurs

State control features:

- No pre-defined meaning (you determine these states entirely)
- Provision for a mechanism of context switching for SCT, including creation of complex state machines
- Defined by events that can happen in the state while the counter is running
- Ability to transition into another state as a result of an event
- Ability to be assigned to one or more states
- Ability for state variables to allow sequencing across multiple counter cycles

64.3 Functional description

64.3.1 SCT operation

In its simplest, single-state configuration, SCT operates as an event-controlled single or bidirectional counter. You can configure events as:

- Counter match events
- An input or output level
- Transitions on an input or output pin

- Combination of match and I/O behavior

In response to an event, SCT output (or outputs) can transition, or SCT can perform other actions such as creating an interrupt, and starting, stopping, or resetting the counter. You can perform multiple, simultaneous actions for each event. Furthermore, any number of events can trigger a specific SCT action.

A single or multiple SCT actions uniquely define an event. Events that you enable to trigger an SCT action or actions at any stage of the counter define a state, and the events not selected for this state are ignored.

In a multi-state configuration, states change in response to events. A state change is an additional action that SCT can perform when an event occurs. If you configure an event to change the state, the new state defines a new set of events resulting in different SCT actions. Through multiple cycles of the counter, events can change the state multiple times and thus create a large variety of event-controlled transitions on the SCT outputs and/or interrupts.

After configuration, SCT can run continuously, without your intervention, and can generate multiple output patterns entirely under the control of events. See the following sections to perform various SCT actions:

- Configuring SCT for multiple events and states to configure SCT
- Running SCT to start, run, and stop SCT
- Configuring SCT without using states to configure SCT as a simple, event-controlled counter or timer

64.3.2 Configuring SCT without using states

SCT can function as a standard counter or timer with external capture inputs and match outputs, without using the state logic. To use SCT without states, configure SCT as follows:

1. Write 0 to State Variable (STATE) (0 is the default).
2. Write 0 to EV*n*_CTRL[STATELD] and EV*n*_CTRL[STATEV] for each event.
3. Write 0x1 to EV*n*_STATE[STATEMSKn] for each event. This enables the event.

In effect, the event occurs in a single state that never changes when the counter is running.

64.3.3 Configuring SCT for multiple events and states

64.3.3.1 Configure the counter

1. Configure the low (L) and high (H) counters in SCT Configuration (CONFIG) by selecting two independent 16-bit counters (L and H) or one combined 32-bit counter in CONFIG[UNIFY].
2. Select the SCT clock source by using CONFIG[CLKMODE] and CONFIG[CKSEL] from any of the inputs or an internal clock.

64.3.3.2 Configure the match and capture registers

1. Select the number of match and capture registers that the application uses (not more than what is available on this chip), by using Match and Capture Register Mode (REGMODE). For each of the match and capture register pairs, you can select whether the register is used as a match register or capture register.
2. Define match conditions for each selected match register:
   - Each match register (Match Value (MATCH0 - MATCH15)) sets one match value, if a 32-bit counter is used, or two match values, if the L and H 16-bit counters are used.
   - Each match reload register (Match Reload Value (MATCHREL0 - MATCHREL15)) sets a reload value that is loaded into the match register when the counter reaches a limit condition or the value 0.

64.3.3.3 Configure events and event responses

State variables allow SCT control across more than one cycle of the counter. Counter matches, I/O edges, and state values are combined into a set of general-purpose events that can switch outputs, request interrupts, and change state values (see Figure 264). Perform this procedure to configure events and event responses:

1. Define what triggers each event, by using Event n Control (EV0_CTRL - EV15_CTRL) (one register per event):
   - Using EV*n*_CTRL[COMBMODE], select whether the event occurs:
     - On change of an input or output
     - On an input or output level
     - On a match condition of the counter
     - On a combination of match and I/O conditions
   - For a match condition:
     a. Select the match register that contains the match condition for the event to occur. Enter the number of the selected match register in EV*n*_CTRL[MATCHSEL].
     b. Define, by using EV*n*_CTRL[HEVENT], whether the event occurs on matching the L or the H counter (when using L and H counters).
   - For an SCT input or output level or transition:
     a. Select the input number or the output number that is associated with this event, using EV*n*_CTRL[IOSEL] and EV*n*_CTRL[OUTSEL].
     b. Define how the selected input or output triggers the event (edge- or level-sensitive) using EV*n*_CTRL[IOCOND].
2. Define the effect of each event on SCT outputs, by using Output n Set (OUT0_SET - OUT9_SET) or Output n Clear (OUT0_CLR - OUT9_CLR) (up to the maximum number of outputs on this chip, one register per output): For each SCT output, select the events that set or clear this output. More than one event can change the output, and each event can change multiple outputs.
3. Define how each event affects the counter:
   a. Write 1 to the corresponding LIMIT[LIMMSK_H] or LIMIT[LIMMSK_L] for the event to set an upper limit for the counter:
      - When a limit event occurs in Unidirectional mode, the counter is cleared to zero and begins counting up on the next clock edge.
      - When a limit event occurs in Bidirectional mode, the counter begins to count down from the current value on the next clock edge.
   b. Write 1 to the corresponding HALT[HALTMSK_H] or HALT[HALTMSK_L] for the event to halt the counter. If the counter is halted, it stops counting, and no new events can occur. You can restore the counter operation by writing 0 CTRL[HALT_H] and/or CTRL[HALT_L].
   c. Write 1 to the corresponding STOP[STOPMSK_H] or STOP[STOPMSK_L] for the event to stop the counter. If the counter is stopped, it stops counting. However, if you configure an event as a transition on an I/O, it can restart the counter.
   d. Write 1 to the corresponding START[STARTMSK_H] or START[STARTMSK_L] for the event to restart the counting. Only events that are defined by an input changing can be used to restart the counter.
4. Define the events that contribute to the SCT interrupt: write 1 to the corresponding EVEN[IEN*n*] and EVFLAG[FLAG*n*] to enable the event to contribute to the SCT interrupt.

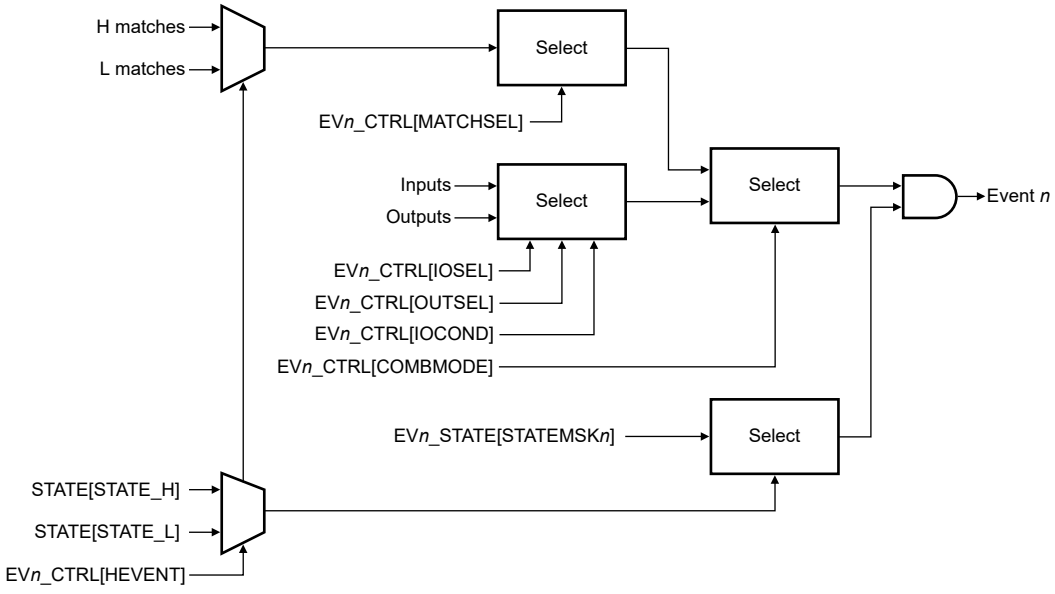


Figure 264. Event selection

64.3.3.4 Configure multiple states

1. Select the state or states (up to two), by using Event n State (EV0_STATE - EV15_STATE), for each event (up to the maximum number of events on this chip—one register per event), in which this event is allowed to occur. You can select each state for more than one event.
2. Determine how the event affects the system state: using Event n Control (EV0_CTRL - EV15_CTRL) (up to the maximum number of events on this chip—one register per event), set the new state value in EV*n*_CTRL[STATEV] for this event. If the event is the highest numbered in the current state, this value is either added to the existing state value or replaces it, depending on the configuration of EV*n*_CTRL[STATELD].

NOTE

If EV*n*_CTRL[STATEV] and EV*n*_CTRL[STATELD] are 0, the state does not change. If there are higher numbered events in the current state, this event cannot change the state.

64.3.3.5 Miscellaneous options

SCT includes a certain (selectable) number of capture registers that you can program to capture the counter contents when one or more events occur.

If the counter is in Bidirectional mode, you can configure Output Counter Direction Control (OUTPUTDIRCTRL) to determine the effect of setting and clearing of an output, based on whether the counter is counting up or down.

64.3.4 State logic

You can configure SCT as a timer or counter with multiple programmable states, which you can define through the events that can be captured in each particular state. A multi-state SCT can change from one state to another when an event that you define triggers a state change through each event's Event n Control (EV0_CTRL - EV15_CTRL) in one of the following ways:

- The event can increment the current state number by a new value.
- The event can write a new state value.

If an event increments the state number beyond the number of available states, SCT enters a locked state where all further events are ignored while the counter is still running. Software must interfere to change out of this state.

Software can capture the counter value (and potentially create an interrupt and write to all outputs) when an event that moves SCT into a locked state occurs. Later, when SCT is in the locked state, software can read the counter again to record the time passed since the locking event and can also read the state variable to obtain the current state number.

If SCT registers an event that forces an abort, putting SCT in a locked state can be a safe way to record the time that has passed since the abort event when no new events are allowed to occur. Because multiple states (any state number between the maximum implemented state and 31) are locked states, you can define multiple abort or error events, each incrementing the state number by a different value.

NOTE

When the implemented state is 32, which is equal to the maximum state number supported, there is no locked state available. In this case, ignore the locked state related description.

64.3.5 Clearing the prescaler

When enabled by a nonzero CTRL[PRE_L] and/or CTRL[PRE_H], the prescaler acts as a clock divider for the counter, like a fractional part of the counter value. The prescaler is cleared whenever the counter is cleared or loaded for any of the following reasons:

- Hardware reset
- When you write to Counter Value (COUNT)
- When you write 1 to CTRL[CLRCTR]
- An event selected by a 1 in the counter limit register when CTRL[BIDIR] = 0

When CTRL[BIDIR] = 0, a limit event caused by an I/O signal can clear a nonzero prescaler. However, a limit event caused by a match clears only a nonzero prescaler in a special case, as described in Match vs. I/O events.

A limit event, when CTRL[BIDIR] = 1, does not clear the prescaler. Instead, it writes 0 to CTRL[DOWN] and decrements the counter on the same clock if the counter is enabled in that clock.

64.3.6 Match vs. I/O events

The prescaler and clock mode 01 make the counter operation complex, in which the SCT clock is the bus clock. However, you can enable the prescaler and counter to count only when a selected edge is detected on a clock input:

- The prescaler is enabled when the clock mode is not 01, or when the input edge that CONFIG[CKSEL] selects is detected.
- The counter is enabled when the prescaler is enabled, and (PRELIM = 0 or the prescaler is equal to the value of PRELIM).

An I/O component of an event can occur in any SCT clock when its counter, CTRL[HALT], is 0. In general, a match component of an event can only occur in an SCT clock when its counter fields, CTRL[HALT] and CTRL[STOP], are both 0 and the counter is enabled.

The following table shows the conditions when various kinds of events can occur.

Table 601. Event conditions

EV*n*_CTRL[COMBMODE]	EV*n*_CTRL[IOCOND]	Event can occur on clock:
I/O	Any	Whenever CTRL[HALT] = 0.
MATCH	Any	When CTRL[HALT] = 0, CTRL[STOP] = 0, and the counter is enabled.
OR	Any	• From the I/O component, whenever CTRL[HALT] = 0.

Table continues on the next page...

Table 601. Event conditions (continued)

EV*n*_CTRL[COMBMODE]	EV*n*_CTRL[IOCOND]	Event can occur on clock:
		• From the match component, when CTRL[HALT] = 0, CTRL[STOP] = 0, and the counter is enabled.
AND	Low or High	When CTRL[HALT] = 0, CTRL[STOP] = 0, and the counter is enabled.
AND	Rise or Fall	Whenever CTRL[HALT] = 0.

64.3.7 Fractional matches

You can configure the first six match registers from Match Value (MATCH0 - MATCH15) to have a fractional portion to their match values. A higher average resolution is achievable on these registers, with an associated Fractional Match (FRACMAT0 - FRACMAT5), by using a dithering mechanism. The dither engine delays the assertion of a match by one counter clock every *n* (0 to 15) out of 16 counter cycles. The value of *n* is specified in Fractional Match (FRACMAT0 - FRACMAT5), associated with each of the first six match registers.

You can disable dithering on any of the match registers by loading all zeroes (the default value) into its corresponding Fractional Match (FRACMAT0 - FRACMAT5).

64.3.7.1 Dithering

At the start of each new SCT counter cycle (that is, when the counter counts down to zero in Bidirectional mode or is cleared to zero by a limit event), the dither engine determines which matches are to be delayed by one clock during the upcoming counter cycle. Delaying the match effectively adds 1 to the designated match value when up-counting or subtracts 1 when down-counting, during that particular counter cycle.

For each dither-enabled match register, the value that you program in its associated Fractional Match (FRACMAT0 - FRACMAT5) specifies for how many counter cycles, out of every 16 of them, its match is to be delayed. An algorithm applied to the FRACMAT value distributes this number as evenly as possible across the 16 counter cycles. This results in a unique dither pattern for each match register (see Table 602).

Dither Condition (DITHER) (event-mask register) provides you additional control over the dithering process. Typically, the dither engine advances though the match dither patterns at the start of every new SCT counter cycle. Dither Condition (DITHER) allows you to specify that advancement to the next element in the dither patterns occurs only if one or more designated events have happened during the previous cycle of the counter.

The dither algorithm is designed to spread out the cycles in which the matches are delayed as evenly as possible across the 16 counter cycles. The following table shows the dither pattern that is applied for each value of FRACMAT. "D" indicates the counter cycles where a match on the relevant match register is delayed.

Table 602. Dither pattern

	Counter cycle															
FRACMAT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1h	-	-	-	-	-	-	-	-	D	-	-	-	-	-	-	-
2h	-	-	-	-	D	-	-	-	-	-	-	-	D	-	-	-
3h	-	-	-	-	D	-	-	-	D	-	-	-	D	-	-	-
4h	-	-	D	-	-	-	D	-	-	-	D	-	-	-	D	-

Table continues on the next page...

Table 602. Dither pattern (continued)

	Counter cycle															
FRACMAT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5h	-	-	D	-	-	-	D	-	D	-	D	-	-	-	D	-
6h	-	-	D	-	D	-	D	-	-	-	D	-	D	-	D	-
7h	-	-	D	-	D	-	D	-	D	-	D	-	D	-	D	-
8h	-	D	-	D	-	D	-	D	-	D	-	D	-	D	-	D
9h	-	D	-	D	-	D	-	D	D	D	-	D	-	D	-	D
Ah	-	D	-	D	D	D	-	D	-	D	-	D	D	D	-	D
Bh	-	D	-	D	D	D	-	D	D	D	-	D	D	D	-	D
Ch	-	D	D	D	-	D	D	D	-	D	D	D	-	D	D	D
Dh	-	D	D	D	-	D	D	D	D	D	D	D	-	D	D	D
Eh	-	D	D	D	D	D	D	D	-	D	D	D	D	D	D	D
Fh	-	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

64.4 External signals

SCT inputs can come from numerous places and are selected by the SCT input mux.

SCT outputs can be routed to multiple places and can be connected to both a pin and an ADC trigger at the same time.

Table 603. SCT signals

Signal	I/O	Description
SCT_IN*n*	I	SCT input. Signals come from GPIO sources, each from a variety of pins selected via IOPCTL muxing.
SCT_OUT*n*	O	SCT output. Signals go to IOPCTL muxing where they can be (optionally) output onto a selection of GPIO pins.

64.5 Initialization

To initialize SCT:

1. Enable the clock to SCT.
2. Clear the SCT peripheral reset.
3. Configure the interrupts.
4. Connect the SCT outputs to external pins.
5. Configure the DMA request lines to connect to the DMA trigger inputs.

64.6 Application information

64.6.1 Running SCT

1. Initialize SCT (see Initialization).
2. Write to State Variable (STATE) to define the initial state. By default, the initial state is 0.

3. To start SCT, write to SCT Control (CTRL):
    a. Clear the counters.
    b. Write 0 or 1 to CTRL[STOP_L] and/or CTRL[STOP_H].

    NOTE

    The counter starts counting after CTRL[STOP] becomes 0 as well. If CTRL[STOP] = 1, SCT, instead, waits for an event to occur that you can configure to start the counter.

    c. Select Unidirectional or Bidirectional counting mode (CTRL[BIDIR_L] and/or CTRL[BIDIR_H]) for each counter.
    d. Select the prescale factor for the counter clock (CTRL[PRE_L] or CTRL[PRE_H]).
    e. Write 0 to CTRL[HALT_L] and/or CTRL[HALT_H]. By default, the counters halt and no events can occur.
4. Stop or halt the counters at any time by using CTRL[STOP_L] and/or CTRL[STOP_H], or CTRL[HALT_L] and/or CTRL[HALT_H], respectively.
    - After the counters stop, both you and an event configured to write 0 to CTRL[STOP_L] and/or CTRL[STOP_H] can start the counter again.
    - After the counter halts, only writing 0 to CTRL[HALT_L] and/or CTRL[HALT_H] can start the counter again. No events can occur.
    - After the counters halt, you can set any SCT output as high or low by writing directly to the OUT register.

You can read the current SCT state at any time by using State Variable (STATE).

To change the current state (that is, independently of any event occurring), write 1 to CTRL[HALT] and write to State Variable (STATE) to change the state value. You are allowed to write to State Variable (STATE) only when the counter halts (CTRL[HALT_L] and/or CTRL[HALT_H] = 1) and no events can occur.

64.6.2 SCT configuration example

This section discusses SCT configuration by using an example, showing the simple application of SCT with two sets of match events (EV0/1 and EV3/4) to set and clear SCT output 0. The timer automatically resets whenever it reaches the MAT0 match value.

In the initial state 0, match event EV0 sets output 0 to high and match event EV1 clears output 0.

When monitoring SCT input 0:

- If input 0 is low at the next timer reset (EV2), the state changes to state 1, enabling events EV3 and EV4. This leads to the same output but with different match value triggers.
- If input 0 is high at the next timer reset, the associated event (EV5) causes the state to change back to the state enabling events EV0 and EV1.

Figure 265 uses this SCT configuration:

- One input
- One output
- Five match registers
- Six events, and match 0 used with the autolimit function
- Two states

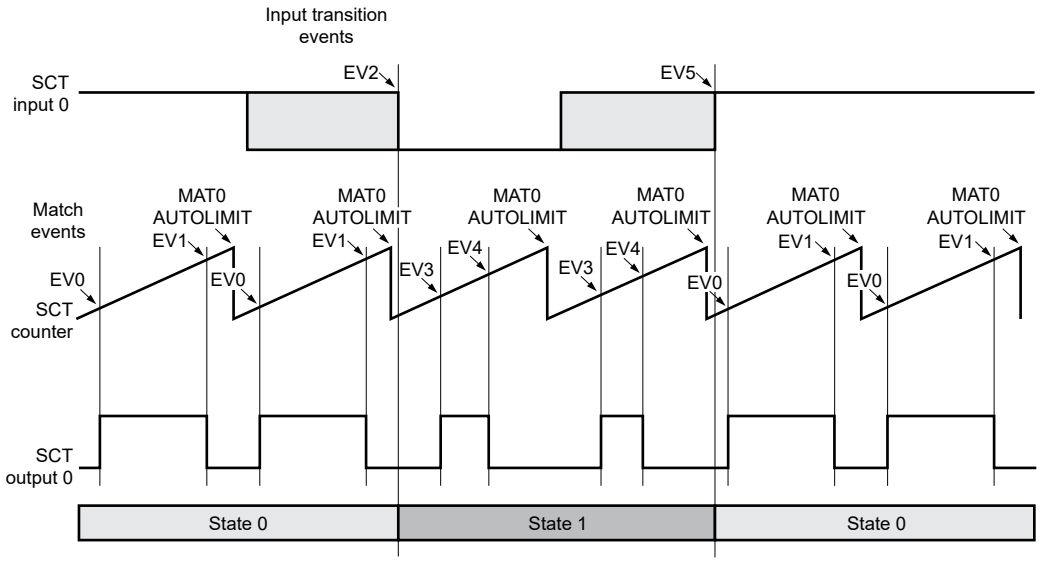


Figure 265. SCT configuration example: pictorial representation

This application of SCT uses the configuration listed in the following table (all register values not listed are their default values):

Table 604. SCT configuration example: settings

Configuration	Register	Setting
Counter	SCT Configuration (CONFIG)	Use a unified counter (CONFIG[UNIFY] = 1).
	SCT Configuration (CONFIG)	Enable the autolimit for MAT0 (CONFIG[AUTOLIMIT_L] = 1).
	SCT Control (CTRL)	Use a unidirectional counter (CTRL[BIDIR_L] = 0).
Clock base	SCT Configuration (CONFIG)	Use the default values for clock configuration.
Match and capture registers	Match and Capture Register Mode (REGMODE)	Configure one match register for each match event by writing 0 to bits 0, 1, 2, 3, and 4 of REGMODE[REGMOD_L*n*].
Define match values	MATCH (0/1/2/3/4)	Set a match value MATCH0/1/2/4_L in each register. The match register 0 serves as an automatic limit event that resets the counter, without using an event. To enable the automatic limit, write 1 to CONFIG[AUTOLIMIT].

Table continues on the next page...

Table 604. SCT configuration example: settings (continued)

Configuration	Register	Setting
Define match reload values	MATCHREL (0/1/2/3/4)	Set a match reload value RELOAD0/1/2/3/4_L in each register (same as the match value in this example).
Define when event 0 occurs	EV0_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x1. Event 0 uses match condition only. • Set EV*n*_CTRL[MATCHSEL] = 1. Select the match value of match register 1. The match value of MAT1 is associated with event 0.
Define when event 1 occurs	EV1_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x1. Event 1 uses match condition only. • Set EV*n*_CTRL[MATCHSEL] = 2. Select the match value of match register 2. The match value of MAT2 is associated with event 1.
Define when event 2 occurs	EV2_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x3. Event 2 uses match condition and I/O condition. • Set EV*n*_CTRL[IOSEL] = 0. Select input 0. • Set EV*n*_CTRL[IOCOND] = 0x0. Input 0 is low. • Set EV*n*_CTRL[MATCHSEL] = 0. This selects match register 0 to qualify the event = 0.
Define how event 2 changes the state	EV2_CTRL	Write 1 to EV*n*_CTRL[STATEV] and EV*n*_CTRL[STATELD]. Event 2 changes the state to state 1.
Define when event 3 occurs	EV3_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x1. Event 3 uses match condition only. • Set EV*n*_CTRL[MATCHSEL] = 0x3. Select match value of match register 3. The match value of MAT3 is associated with event 3.
Define when event 4 occurs	EV4_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x1. Event 4 uses match condition only. • Set EV*n*_CTRL[MATCHSEL] = 0x4. Select match value of match register 4. The match value of MAT4 is associated with event 4.
Define when event 5 occurs	EV5_CTRL	• Set EV*n*_CTRL[COMBMODE] = 0x3. Event 5 uses match condition and I/O condition. • Set EV*n*_CTRL[IOSEL] = 0. Select input 0. • Set EV*n*_CTRL[IOCOND] = 0x3. Input 0 is high. • Set EV*n*_CTRL[MATCHSEL] = 0. Chooses match register 0 to qualify the event.
Define how event 5 changes the state	EV5_CTRL	Write 0 to EV*n*_CTRL[STATEV] and 1 to EV*n*_CTRL[STATELD]. Event 5 changes the state to state 0.
Define by which events output 0 is set	OUT0_SET	Set SET0 bits 0 (for event 0) and 3 (for event 3) to one to set the output when these events 0 and 3 occur.
Define by which events output 0 is cleared	OUT0_CLR	Set CLR0 bits 1 (for events 1) and 4 (for event 4) to one to clear the output when events 1 and 4 occur.

Table continues on the next page...

Table 604. SCT configuration example: settings (continued)

Configuration	Register	Setting
Configure states to enable event 0	OUT0_STATE	Set STATEMSK0 bit 0 to 1, enabling event 0 in state 0. Set all other bits to 0.
Configure states to enable event 1	EV1_STATE	Set STATEMSK1 bit 0 to 1, enabling event 1 in state 0. Set all other bits to 0.
Configure states to enable event 2	EV2_STATE	Set STATEMSK2 bit 0 to 1, enabling event 2 in state 0. Set all other bits to 0.
Configure states to enable event 3	EV3_STATE	Set STATEMSK3 bit 1 to 1, enabling event 3 in state 1. Set all other bits to 0.
Configure states to enable event 4	EV4_STATE	Set STATEMSK4 bit 1 to 1, enabling event 4 in state 1. Set all other bits to 0.
Configure states to enable event 5	EV5_STATE	Set STATEMSK5 bit 1 to 1, enabling event 5 in state 1. Set all other bits to 0.

64.7 Memory map and register definition

64.7.1 Register configuration

For most of the SCT registers, the register function depends on the settings of certain other register fields:

- CONFIG[UNIFY] determines whether SCT is used as one 32-bit register (for operation as one 32-bit counter or timer) or as two 16-bit counters or timers named L and H.
  - If CONFIG[UNIFY] = 1, only one register is used (for operation as one 32-bit counter or timer).
  - If CONFIG[UNIFY] = 0, you can access the L and H registers using a 32-bit read or write operation, or you can read or write to them individually.

Typically, you configure CONFIG[UNIFY] by writing to SCT Configuration (CONFIG) before accessing any other registers.

- REGMOD[REGMOD_H*n*] and REGMOD[REGMOD_L*n*] determine whether each set of match and capture registers uses the match or capture functionality:
  - If REGMODE*n* = 0, registers operate as match and reload registers.
  - If REGMODE*n* = 1, registers operate as capture and capture control registers.

64.7.2 Register functional grouping

Most SCT registers either configure an event or select an event for a specific action of the counter (or counters) and outputs. The following figure shows the registers and register fields that you must configure for each event. In Figure 266, the following letters are used to represent the maximum quantity of certain SCT features:

- i = match and captures
- j = events
- k = states
- p = outputs
- q = inputs

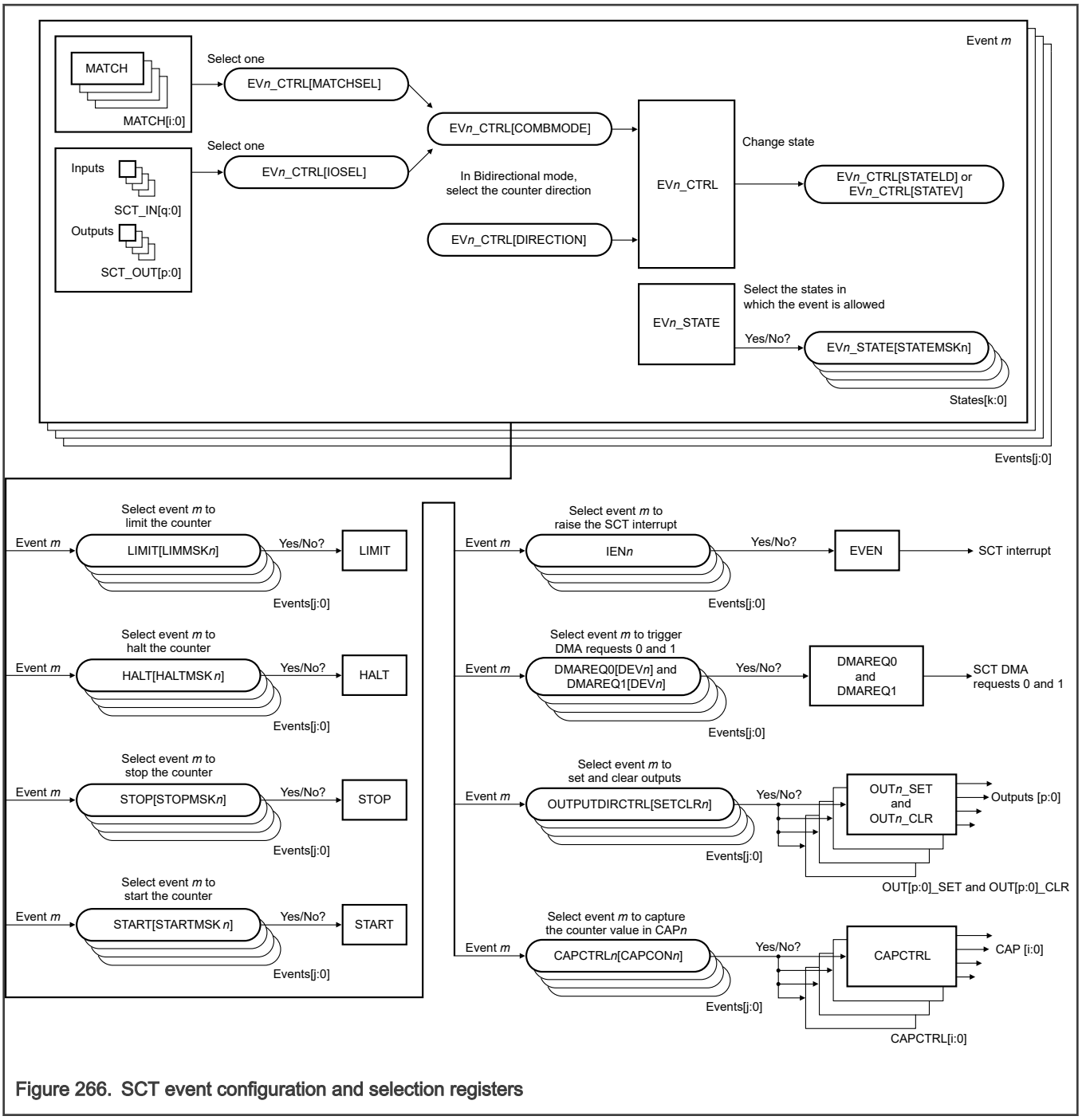


Figure 266. SCT event configuration and selection registers

64.7.2.1 Counter configuration and control registers

SCT contains these registers for configuration, monitoring, and control:

- SCT Configuration (CONFIG) configures SCT in a single, 32-bit counter mode or in dual, 16-bit counter mode. This register helps you perform clocking, clock synchronization, configure automatic limits, and use reload registers.
- SCT Control (CTRL) helps you monitor and set the counter direction. Additionally, using this register, you can clear, start, stop, or halt the 32-bit counter or individual 16-bit counters in Dual-Counter mode.

64.7.2.2 Event configuration registers

Each event is associated with these registers:

- Event n State (EV0_STATE - EV15_STATE) to enable the event
- Event n Control (EV0_CTRL - EV15_CTRL) to define what triggers the event

64.7.2.3 Match and capture registers

SCT includes a set of registers to store the SCT match or capture values:

- Each Match Value (MATCH0 - MATCH15) is associated with a corresponding Match Reload Value (MATCHREL0 - MATCHREL15), which automatically reloads Match Value (MATCH0 - MATCH15) at the beginning of each counter cycle.
- One Match and Capture Register Mode (REGMODE) to configure each match and capture register for either storing a match value or capture value.
- A set of match and capture registers with each register, depending on the setting of Match and Capture Register Mode (REGMODE), for either storing a match value or counter capture value.
- One Capture Control (CAPCTRL0 - CAPCTRL15) for each Capture Value (CAP0 - CAP15).

Figure 267 shows the match logic for SCT.

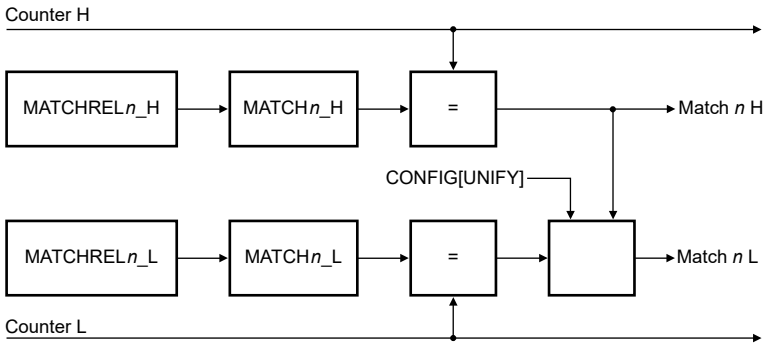


Figure 267. Match logic

Figure 268 shows the capture logic for SCT.

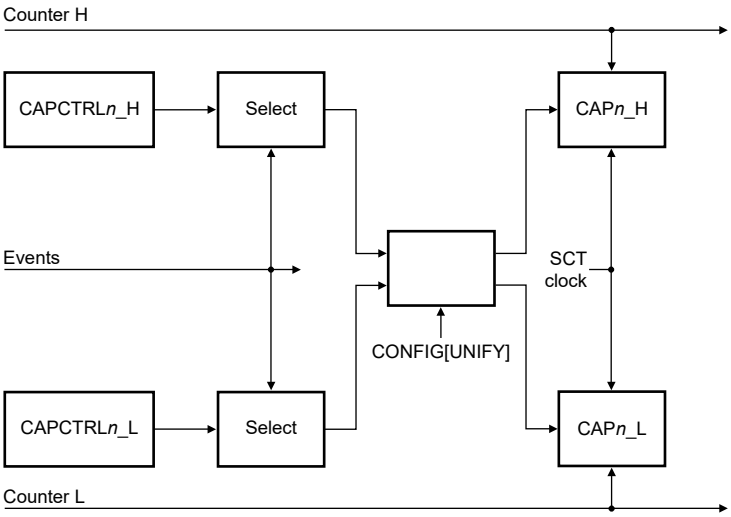


Figure 268. Capture logic

64.7.2.4 Event select registers for counter operations

This group of registers helps you select the events that affect the counter. The counter actions are limit, halt, start, and stop, which apply to the unified counter or to the two 16-bit counters:

- SCT Limit Event Select (LIMIT) selects the events that limit the counter.
- Start Event Select (START) and Stop Event Select (STOP) select the events that start or stop the counter.
- Halt Event Select (HALT) selects the events that halt the counter.
- Counter Value (COUNT) contains the counter value (or values) in the dual-counter setup.

All the aforementioned registers except Start Event Select (START) contain one field per event that selects whether each event limits, stops, starts, or halts the counter (or counters) in Dual-Counter mode. In this mode, you can select the events independently for each counter.

64.7.2.5 Event select registers for setting or clearing outputs

This group of registers helps you select the events that affect the level of each SCT output. Also included in this group are registers to manage conflicts that occur when events try to set or clear the same output:

- One Output n Set (OUT0_SET - OUT9_SET) for each output to select the events that set the output
- One Output n Clear (OUT0_CLR - OUT9_CLR) for each output to select the events that clear the output
- Output Conflict Resolution (RES) that defines an action when multiple events try to control an output at the same time
- Conflict Flag (CONFLAG) and Conflict Interrupt Enable (CONEN) that monitor interrupts arising from "output set and clear conflicts"
- Output Counter Direction Control (OUTPUTDIRCTRL) that interchanges the set and clear output operation caused by an event in Bidirectional mode

Output n Set (OUT0_SET - OUT9_SET) and Output n Clear (OUT0_CLR - OUT9_CLR) contain one field each per event that selects whether the event changes the state for a given output *n*.

In Dual-Counter mode, you can select the events independently for each output.

The following figure shows an SCT output slice.

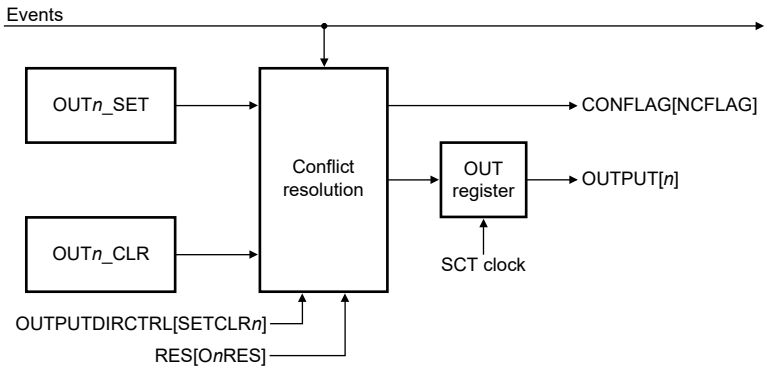


Figure 269. Output slice *n*

64.7.2.6 Event select registers for capturing a counter value

This group of registers helps you select events that capture the counter value and store it in one of Capture Value (CAP0 - CAP15). Each capture register *n* includes an associated Capture Control (CAPCTRL0 - CAPCTRL15), which in turn selects the events to capture the counter value.

64.7.2.7 Event select register for initiating DMA transfers

This register allows each of the two DMA requests to select the events that can generate a DMA request.

DMA Request 0 (DMAREQ0) and DMA Request 1 (DMAREQ1) contain a field for each event that selects whether the event generates a DMA request. An additional field enables the DMA trigger after the match registers are reloaded.

64.7.2.8 Interrupt handling registers

The following registers provide flags that are set by events and select the events that request an interrupt:

- Event Flag (EVFLAG) provides a flag for each event that is set when the event occurs.
- Event Interrupt Enable (EVEN) provides a field for each event to be enabled for the SCT interrupt.

This following figure shows an event triggering SCT to generate an interrupt.

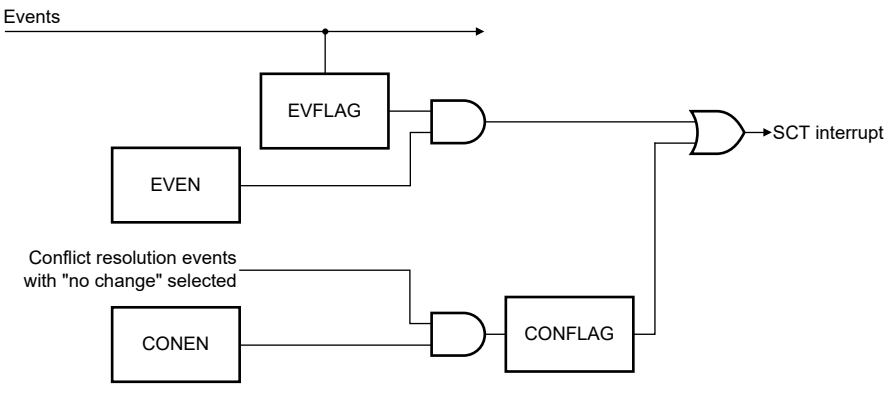


Figure 270. SCT interrupt generation

64.7.2.9 Registers for controlling SCT inputs and outputs

The following registers allow you (as opposed to events) to set SCT input and outputs:

- Input State (INPUT) to read the state of any of the SCT inputs
- Output State (OUTPUT) to set or clear any of the SCT outputs or to read their states

64.7.3 SCT register descriptions

SCT is a powerful, flexible timer module capable of creating complex PWM waveforms and performing other advanced timing and control operations with minimal or no CPU intervention.

64.7.3.1 SCT memory map

SCT0 base address: 4011_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SCT Configuration (CONFIG)	32	RW	See section
4h	SCT Control (CTRL)	32	RW	See section
8h	SCT Limit Event Select (LIMIT)	32	RW	0000_0000h
Ch	Halt Event Select (HALT)	32	RW	0000_0000h
10h	Stop Event Select (STOP)	32	RW	0000_0000h
14h	Start Event Select (START)	32	RW	0000_0000h
18h	Dither Condition (DITHER)	32	RW	0000_0000h
40h	Counter Value (COUNT)	32	RW	0000_0000h
44h	State Variable (STATE)	32	RW	See section
48h	Input State (INPUT)	32	R	See section
4Ch	Match and Capture Register Mode (REGMODE)	32	RW	0000_0000h
50h	Output State (OUTPUT)	32	RW	See section
54h	Output Counter Direction Control (OUTPUTDIRCTRL)	32	RW	0000_0000h
58h	Output Conflict Resolution (RES)	32	RW	0000_0000h
5Ch	DMA Request 0 (DMAREQ0)	32	RW	See section
60h	DMA Request 1 (DMAREQ1)	32	RW	See section
F0h	Event Interrupt Enable (EVEN)	32	RW	See section
F4h	Event Flag (EVFLAG)	32	RW	See section
F8h	Conflict Interrupt Enable (CONEN)	32	RW	See section
FCh	Conflict Flag (CONFLAG)	32	RW	See section
100h - 13Ch	Capture Value (CAP0 - CAP15)	32	RW	0000_0000h
100h - 13Ch	Match Value (MATCH0 - MATCH15)	32	RW	0000_0000h
140h - 154h	Fractional Match (FRACMAT0 - FRACMAT5)	32	RW	0000_0000h
200h - 23Ch	Capture Control (CAPCTRL0 - CAPCTRL15)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
200h - 23Ch	Match Reload Value (MATCHREL0 - MATCHREL15)	32	RW	0000_0000h
240h - 254h	Fractional Match Reload (FRACMATREL0 - FRACMATREL5)	32	RW	0000_0000h
300h	Event n State (EV0_STATE)	32	RW	See section
304h	Event n Control (EV0_CTRL)	32	RW	See section
308h	Event n State (EV1_STATE)	32	RW	See section
30Ch	Event n Control (EV1_CTRL)	32	RW	See section
310h	Event n State (EV2_STATE)	32	RW	See section
314h	Event n Control (EV2_CTRL)	32	RW	See section
318h	Event n State (EV3_STATE)	32	RW	See section
31Ch	Event n Control (EV3_CTRL)	32	RW	See section
320h	Event n State (EV4_STATE)	32	RW	See section
324h	Event n Control (EV4_CTRL)	32	RW	See section
328h	Event n State (EV5_STATE)	32	RW	See section
32Ch	Event n Control (EV5_CTRL)	32	RW	See section
330h	Event n State (EV6_STATE)	32	RW	See section
334h	Event n Control (EV6_CTRL)	32	RW	See section
338h	Event n State (EV7_STATE)	32	RW	See section
33Ch	Event n Control (EV7_CTRL)	32	RW	See section
340h	Event n State (EV8_STATE)	32	RW	See section
344h	Event n Control (EV8_CTRL)	32	RW	See section
348h	Event n State (EV9_STATE)	32	RW	See section
34Ch	Event n Control (EV9_CTRL)	32	RW	See section
350h	Event n State (EV10_STATE)	32	RW	See section
354h	Event n Control (EV10_CTRL)	32	RW	See section
358h	Event n State (EV11_STATE)	32	RW	See section
35Ch	Event n Control (EV11_CTRL)	32	RW	See section
360h	Event n State (EV12_STATE)	32	RW	See section
364h	Event n Control (EV12_CTRL)	32	RW	See section
368h	Event n State (EV13_STATE)	32	RW	See section
36Ch	Event n Control (EV13_CTRL)	32	RW	See section
370h	Event n State (EV14_STATE)	32	RW	See section

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
374h	Event n Control (EV14_CTRL)	32	RW	See section
378h	Event n State (EV15_STATE)	32	RW	See section
37Ch	Event n Control (EV15_CTRL)	32	RW	See section
500h	Output n Set (OUT0_SET)	32	RW	See section
504h	Output n Clear (OUT0_CLR)	32	RW	See section
508h	Output n Set (OUT1_SET)	32	RW	See section
50Ch	Output n Clear (OUT1_CLR)	32	RW	See section
510h	Output n Set (OUT2_SET)	32	RW	See section
514h	Output n Clear (OUT2_CLR)	32	RW	See section
518h	Output n Set (OUT3_SET)	32	RW	See section
51Ch	Output n Clear (OUT3_CLR)	32	RW	See section
520h	Output n Set (OUT4_SET)	32	RW	See section
524h	Output n Clear (OUT4_CLR)	32	RW	See section
528h	Output n Set (OUT5_SET)	32	RW	See section
52Ch	Output n Clear (OUT5_CLR)	32	RW	See section
530h	Output n Set (OUT6_SET)	32	RW	See section
534h	Output n Clear (OUT6_CLR)	32	RW	See section
538h	Output n Set (OUT7_SET)	32	RW	See section
53Ch	Output n Clear (OUT7_CLR)	32	RW	See section
540h	Output n Set (OUT8_SET)	32	RW	See section
544h	Output n Clear (OUT8_CLR)	32	RW	See section
548h	Output n Set (OUT9_SET)	32	RW	See section
54Ch	Output n Clear (OUT9_CLR)	32	RW	See section

64.7.3.2 SCT Configuration (CONFIG)

Offset

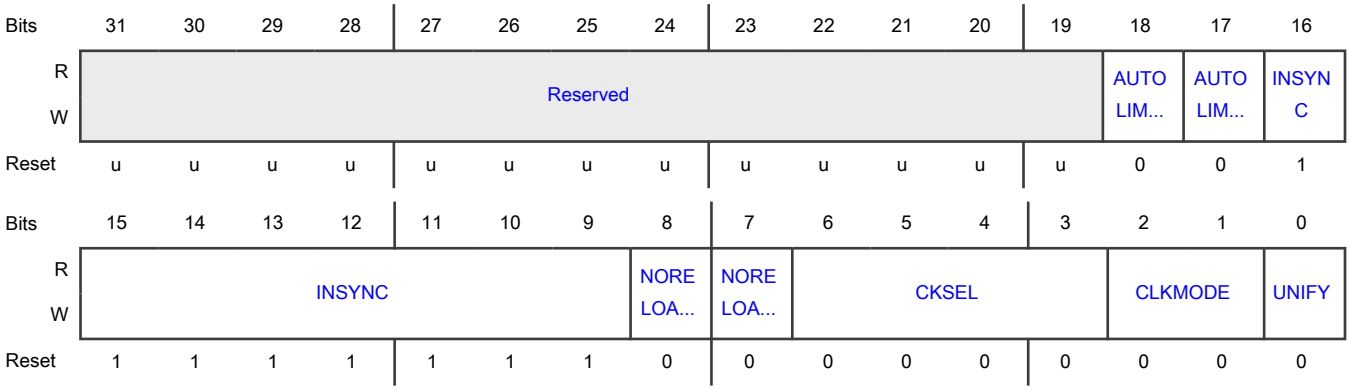
Register	Offset
CONFIG	0h

Function

Configures the overall SCT operation. You must write to this register before writing to any other registers.

Only word writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31-19 —	Reserved
18 AUTOLIMIT_H	Auto Limit Higher Enables the counter to be cleared to zero in Unidirectional mode or to change the direction of count in Bidirectional mode, as with any SCT Limit Event Select (LIMIT) event. You can write to this field at any time, except when CONFIG[UNIFY] = 1. If this field = 1, a match on match register 0 is considered as the limit condition without the need to define an associated event. 0b - Disables 1b - Enables
17 AUTOLIMIT_L	Auto Limit Lower Enables the counter to be cleared to zero in Unidirectional mode or to change the direction of count in Bidirectional mode, as with any SCT Limit Event Select (LIMIT) event. You can write to this field at any time, and the field applies to both the higher and lower registers when CONFIG[UNIFY] = 1. If this field = 1, a match on match register 0 is considered as the limit condition. You need not define an associated event. 0b - Disables 1b - Enables
16-9 INSYNC	Input Synchronization Specifies synchronization for input *n* (bit 9 = input 0, bit 10 = input 1, bit 16 = input 7, and so on). If you write 1 to any of these bits, it subjects the corresponding input to synchronization with the SCT clock, before it is used to create an event. If an input is already known to be synchronous with the SCT clock, you may write 0 to this field for a faster input response.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE SCT clock is the system clock for clock modes 0-2. It is the selected, asynchronous SCT input clock for clock mode 3. This field only affects inputs used for event generation, and does not apply to the clock input specified in CONFIG[CKSEL].
8 NORELOAD_H	No Reload Higher Match Specifies whether higher match registers are reloaded from their respective reload registers. Writing 1 to this field eliminates the need to write to Match Reload Value (MATCHREL0 - MATCHREL15) if the match values are fixed. It prevents the higher match registers to be reloaded from their respective reload registers. You can write to this field at any time, except when CONFIG[UNIFY] = 1. 0b - Reloaded 1b - Not reloaded
7 NORELOAD_L	No Reload Lower Match Specifies whether lower match registers are reloaded from their respective reload registers. Writing 1 to this field eliminates the need to write to Match Reload Value (MATCHREL0 - MATCHREL15) if the match values are fixed. It prevents the lower match registers to be reloaded from their respective reload registers. You can write to this field at any time, and the field applies to both the higher and lower registers when CONFIG[UNIFY] = 1. 0b - Reloaded 1b - Not reloaded
6-3 CKSEL	SCT Clock Select Specifies SCT clock. The specific functionality of the designated input or edge is dependent on the mode you select using CONFIG[CLKMODE]. 0000b - Rising edges on input 0 0001b - Falling edges on input 0 0010b - Rising edges on input 1 0011b - Falling edges on input 1 0100b - Rising edges on input 2 0101b - Falling edges on input 2 0110b - Rising edges on input 3 0111b - Falling edges on input 3 1000b - Rising edges on input 4

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1001b - Falling edges on input 4 1010b - Rising edges on input 5 1011b - Falling edges on input 5 1100b - Rising edges on input 6 1101b - Falling edges on input 6 1110b - Rising edges on input 7 1111b - Falling edges on input 7
2-1 CLKMODE	SCT Clock Mode Specifies SCT clock mode. • If this field = 0, SCT uses System Clock mode. In this mode, the system clock clocks the entire SCT module, including all counters and counter prescalers. • If this field = 1, SCT uses Sampled System Clock mode. In this mode, the system clock clocks the SCT module, but the counter and prescalers are only enabled to count when the designated edge is detected on the input that CONFIG[CKSEL] selects. The minimum pulse width on the selected clock-gate input is 1 bus clock period. This is a high-performance mode. • If this field = 10, SCT uses Input Clock mode. The input or edge that CONFIG[CKSEL] selects clocks the SCT module, including all counters and prescalers, after first being synchronized with the system clock. The minimum pulse width on the clock input is one bus clock period. This is a low-power, sampled-clock mode. • If this field = 11, SCT uses Asynchronous mode. In this mode, the input or edge that CONFIG[CKSEL] selects clocks the entire SCT module. The SCT outputs are switched synchronously to the SCT input clock, and not the system clock. The input clock rate must be at least half the system clock rate and can be the same or faster than the system clock. 00b - System Clock mode 01b - Sampled System Clock mode 10b - SCT Input Clock mode 11b - Asynchronous mode
0 UNIFY	SCT Operation Determines whether SCT is used as a single, unified 32-bit counter or timer or as dual, independent 16-bit counters or timers named low (L) and high (H). You must define the setting of this field before accessing other registers. 0b - Dual counters, COUNTER_L and COUNTER_H 1b - Unified counter

64.7.3.3 SCT Control (CTRL)

Offset

Register	Offset
CTRL	4h

Function

Allows SCT control.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, CTRL_L and CTRL_H. You can read or write to both these registers, individually.

You can write to all fields in this register when the counter is stopped or halted. When the counter is running, the only fields that you can write to are STOP or HALT. You can write to the other fields in a subsequent write, after HALT becomes 1.

NOTE

- If you select CLKMODE = 0x3, wait for at least 12 system clock cycles between a write access to the H, L, or the unified version of this register and the next write access. This restriction does not apply when writing to the HALT field(s) and then writing to this register again to restart the counters (for example, you must update Match Value (MATCH0 - MATCH15), and this is allowed only when the counters are halted).
- If SCT is operating as two 16-bit counters, events can only modify the state of the outputs when no counter is halted. This is true regardless of what triggered the event.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				PRE_H							BIDIR_H	CLRCTR_H	HALT_H	STOP_H	DOWN_H
Reset	u	u	u	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				PRE_L							BIDIR_L	CLRCTR_L	HALT_L	STOP_L	DOWN_L
Reset	u	u	u	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Function
31-29 —	Reserved
28-21	Prescaler for High Counter

Table continues on the next page...

Table continued from the previous page...

Field	Function
PRE_H	Specifies the factor by which the SCT clock is prescaled to produce the high counter clock, which is clocked at the rate of SCT clock ÷ PREL_H + 1. **NOTE** Clear the counter (by writing 1 to CLRCTR) when changing the value of this field.
20 BIDIR_H	Bidirectional Select High Selects the direction of the high counter. If this field = 0, the high counter counts up to its limit condition, and is then cleared to zero. If this field = 1, the high counter counts up to its limit, and then counts down to a limit condition or to zero. 0b - Up 1b - Up-down
19 CLRCTR_H	Clear Counter High Clears the high counter. Writing 1 to this field clears the high counter. CLRCTR_H always reads as 0.
18 HALT_H	Halt Counter High Halts the high counter. When this field = 1, the H counter does not run and no events can occur. Also, CTRL[STOP_H] clears. You can remove the halt condition when SCT is not running (is in a stop condition), by using a single write to this register, to simultaneously clear the HALT and STOP fields. **NOTE** Once set, HALT_H can only be cleared by software to restore counter operation. HALT_H = 1 on reset. 0b - Disable 1b - Enable
17 STOP_H	Stop Counter High Stops the high counter from running. When STOP_H = 1 and HALT = 0, the H counter does not run but I/O events related to the counter can occur. If such an event matches the mask in Start Event Select (START), CTRL[STOP_H] is cleared and counting resumes. 0b - Disabled 1b - Enabled
16 DOWN_H	Down Counter High Specifies whether the high counter is counting up or down.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Hardware sets DOWN_H when the counter is counting up, a counter limit condition occurs, and BIDIR = 1. Hardware clears DOWN_H when the counter is counting down and a limit condition occurs, or when the counter reaches 0. 0b - Up 1b - Down
15-13 —	Reserved
12-5 PRE_L	Prescaler for Low Counter Specifies the factor by which the SCT clock is prescaled to produce the low or unified counter clock. The counter clock is clocked at the rate of SCT clock ÷ PRE_L + 1. **NOTE** You can clear the counter by writing 1 to CLRCTR when changing the value of this field.
4 BIDIR_L	Bidirectional Select Low Selects the direction of the low or unified counter. If this field = 0, the counter counts up to a limit condition and is then cleared to 0. If this field = 1, the counter counts up to a limit, and then counts down to a limit condition or to 0. 0b - Up 1b - Up-down
3 CLRCTR_L	Clear Counter Low Clears the low or unified counter. Writing 1 to this field clears the low or unified counter. The field always reads 0.
2 HALT_L	Halt Counter Low Specifies whether the L or unified counter is enabled. If this field = 1, the low or unified counter does not run and no events can occur. STOP_L also becomes 0. It is possible to remove the halt condition while SCT is in the stop condition (not running), with a single write to this register, to simultaneously clear the HALT field and set the STOP field. Once set, only software can clear HALT_L to restore counter operation. HALT_L = 1 on reset. 0b - Disabled 1b - Enabled
1 STOP_L	Stop Counter Low Specifies whether the low or unified counter is enabled. If STOP_L = 1 and HALT = 0, the low or unified counter does not run, but I/O events related to the counter can occur. If a designated start event occurs, STOP_L is cleared and counting resumes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disabled 1b - Enabled
0 DOWN_L	Down Counter Low Specifies whether the low or unified counter is counting up or down. Hardware sets DOWN_L when the counter is counting up, counter limit occurs, and BIDIR = 1. Hardware clears this field when the counter is counting down and a limit condition occurs, or when the counter reaches 0. 0b - Up 1b - Down

64.7.3.4 SCT Limit Event Select (LIMIT)

Offset

Register	Offset
LIMIT	8h

Function

Selects the SCT limit event.

An event can limit a running counter. When any of the events selected in this register occur, the counter is cleared to zero from its current value or changes the counting direction, if it is in Bidirectional mode. Each field of this register is associated with a different event (bit 0 with event 0, and so on). Writing 1 to a field causes its associated event to serve as a LIMIT event, which whenever occurs, resets the counter to zero in Unidirectional mode, or changes the count direction in Bidirectional mode and keeps running. To define the actual limiting event (a match, an I/O pin toggle, and so on), see Event n Control (EV0_CTRL - EV15_CTRL).

NOTE

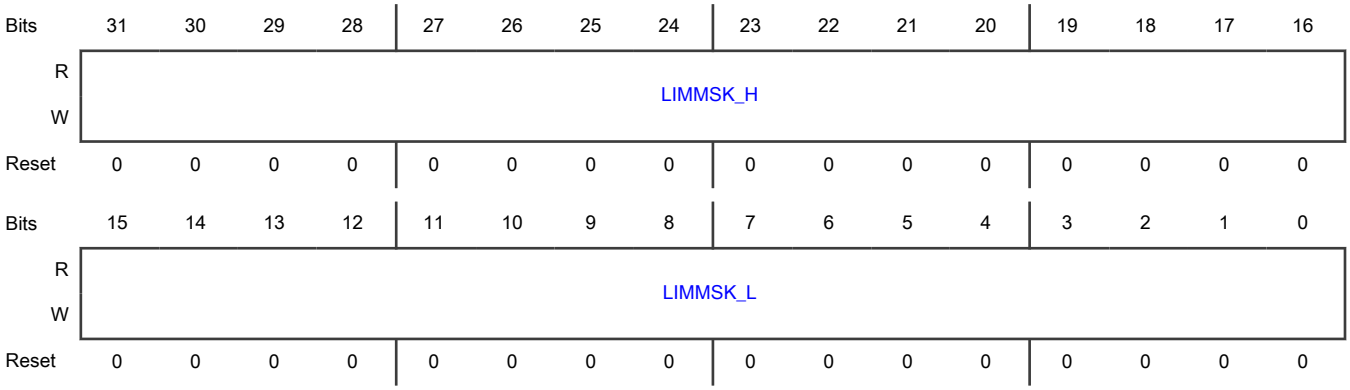
Counting up to all ones or counting down to zero is always equivalent to a limit event occurring.

In addition to using this register to specify events that serve as limits, it is also possible to automatically cause a limit condition whenever a match register 0 match occurs. This eliminates the need to define an event for the sole purpose of creating a limit. CONFIG[AUTOLIMIT_L] and CONFIG[AUTOLIMIT_H] enable or disable this feature.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, LIMIT_L and LIMIT_H. You can read or write to both the _L and _H registers individually.

Diagram



Fields

Field	Function
31-16 LIMMSK_H	Limit Event Counter High Specifies the counter limit for the high counter. If bit *n* is 1, event *n* is used as a counter limit for the high counter (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of events in this SCT.
15-0 LIMMSK_L	Limit Event Counter Low Specifies the counter limit for the low counter. If bit *n* is 1, event *n* is used as a counter limit for the low or unified counter (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.5 Halt Event Select (HALT)

Offset

Register	Offset
HALT	Ch

Function

Halts high and low events.

A halt event can disable (or halt) a running counter. When any of the events selected in this register occur, the counter stops running and all further events are disabled.

Each field of this register is associated with a different event (bit 0 with event 0, and so on). Setting a bit causes the associated event to serve as a HALT event. To define the actual events that cause the counter to halt (a match, an I/O pin toggle, and so on), see Event n Control (EV0_CTRL - EV15_CTRL).

NOTE

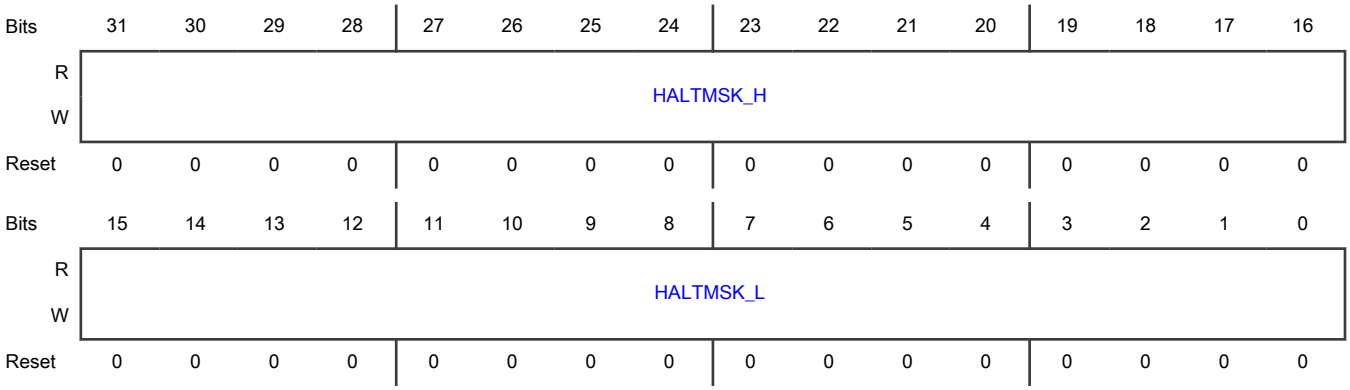
You can remove a HALT condition by clearing CTRL[HALT].

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.

- If CONFIG[UNIFY] = 0, you can write to this register as two registers, HALT_L and HALT_H. You can read or write to both the _L and _H registers individually.

Diagram



Fields

Field	Function
31-16 HALTMSK_H	Halt Event High Halts the high event. If bit *n* is 1, event *n* writes 1 to CTRL[HALT_H] (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of events in this SCT.
15-0 HALTMSK_L	Halt Event Low Halts the low event. If bit *n* is 1, event *n* writes 1 to CTRL[HALT_L] (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.6 Stop Event Select (STOP)

Offset

Register	Offset
STOP	10h

Function

Stops high and low events.

A stop event can stop a running counter. When any of the events selected in this register occur, counting is suspended. This means that the counter stops running and remains at its current value. Event generation remains enabled, and any event selected in Start Event Select (START), such as an I/O event or an event generated by the other counter, can restart the counter.

This register specifies the events that stop the counter. Each field of the register is associated with a different event (bit 0 with event 0, and so on). Setting a bit causes its associated event to serve as a STOP event. To define the actual event that causes the counter to stop (a match, an I/O pin toggle, and so on), see Event n Control (EV0_CTRL - EV15_CTRL).

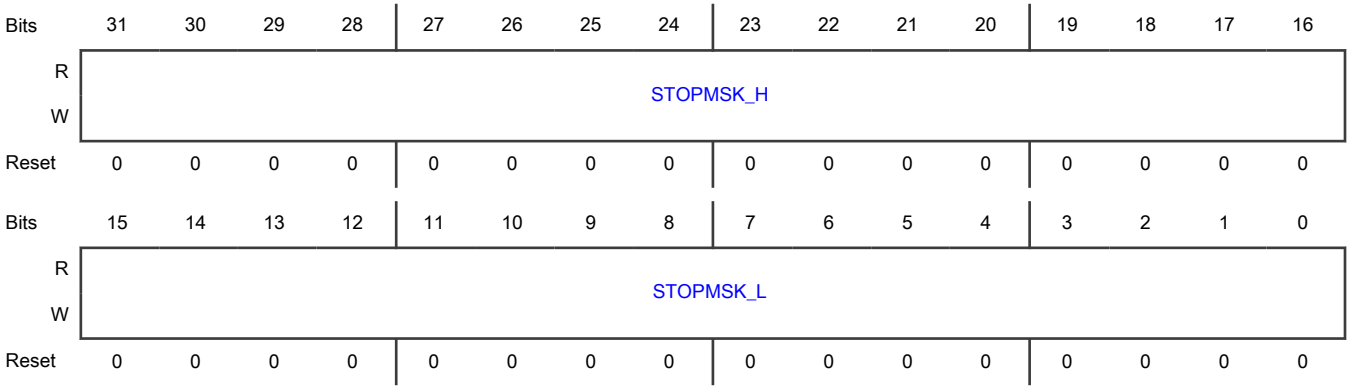
NOTE

You can stop and restart the counter by writing to SCT Control (CTRL).

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, STOP_L and STOP_H. You can read or write to both the _L and _H registers individually.

Diagram



Fields

Field	Function
31-16 STOPMSK_H	Stop Event High Stops the high event. If bit *n* is 1, the event *n* writes 1 to CTRL[STOP_H] (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of events in this SCT.
15-0 STOPMSK_L	Stop Event Low Stops the low event. If bit *n* is 1, the event *n* writes 1 to CTRL[STOP_L] (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.7 Start Event Select (START)

Offset

Register	Offset
START	14h

Function

Starts high and low events.

A start event can restart a stopped counter.

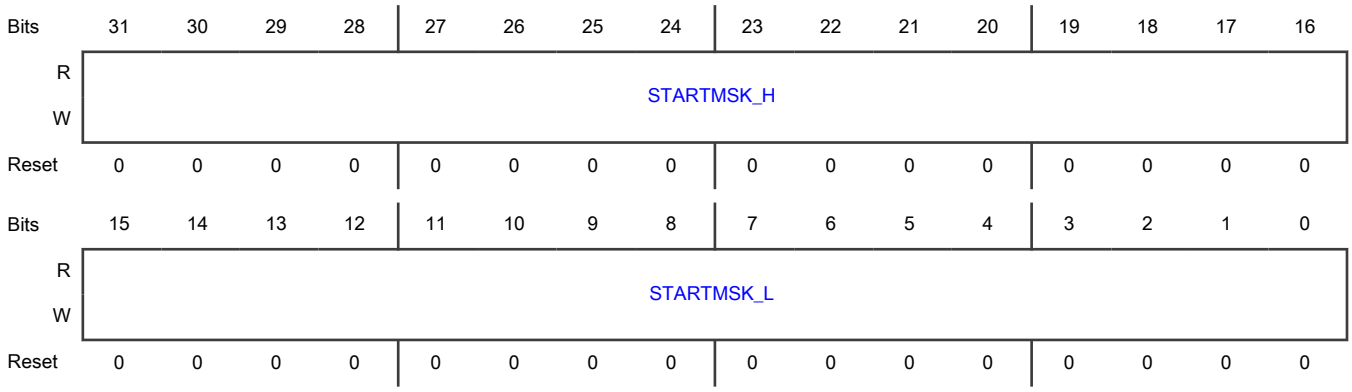
When any of the events selected in Start Event Select (START) occur, counting restarts from the current counter value.

Each field of this register associates with a different event (bit 0 with event 0, and so on). Setting a bit causes the associated event to serve as a START event. When a START event occurs, hardware clears CTRL[STOP]. A START event has no effect on CTRL[HALT]. Only software can remove a HALT condition. To define the actual event that starts the counter (an I/O pin toggle or an event generated by the other running counter in Dual-Counter mode), see Event n Control (EV0_CTRL - EV15_CTRL).

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, START_L and START_H. You can read or write to both the _L and _H registers individually.

Diagram



Fields

Field	Function
31-16 STARTMSK_H	Start Event High Starts the high event. If bit *n* is 1, event *n* writes 0 to CTRL[STOP_H] (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of events in this SCT.
15-0 STARTMSK_L	Start Event Low Starts the low event. If bit *n* is 1, event *n* writes 0 to CTRL[STOP_L] (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.8 Dither Condition (DITHER)

Offset

Register	Offset
DITHER	18h

Function

Enables the dither engine to advance to the next count in the dither pattern every time the SCT counter reaches zero (that is, at the start of every new SCT counter cycle). Using this register, you can alter this behavior by qualifying the advancement through the dither pattern with designated events. As with the other condition or mask registers (such as Halt Event Select (HALT), Stop Event Select (STOP), and SCT Limit Event Select (LIMIT)), each field of this register corresponds to an event.

Writing 1 to one or more fields in this register causes the dither engine to advance to the next element in the dither pattern (that is, increment the 16-state cycle counter) by only following SCT counter cycles during which one or more of the designated dither events have occurred.

There exists a global dither condition register per 16-bit SCT. This register controls advancement through the dither patterns for all the match registers associated with that half of SCT.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONGIG[UNIFY] = 1, only the _L bits are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, START_L and START_H. You can read and write to both the _L and _H registers individually or in a single 32-bit read or write operation.

For details on the dither engine and dither pattern, see Dithering.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DITHER_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DITHER_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 DITHER_H	Dither High Dithers high events. If bit *n* is 1, event *n* writes 0 to CTRL[STOP_H] (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of events in this SCT.
15-0 DITHER_L	Dither Low Dithers low events. If bit *n* is 1, event *n* writes 0 to CTRL[STOP_L] (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.9 Counter Value (COUNT)

Offset

Register	Offset
COUNT	40h

Function

Defines high and low counters.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, this register acts as a unified 32-bit register and both the _L and _H fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, COUNT_L and COUNT_H. You can read and write to both the _L and _H registers individually. In this case, the _L and _H registers count independently under the control of other registers.

You are allowed to write to COUNT_L, COUNT_H, or the unified register only when the corresponding counter is halted (CTRL[HALT] = 1). In case you attempt to write to the counter when it is not halted, a bus error takes place. However, you can read the counter registers at any time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTR_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTR_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 CTR_H	Counter High Allows reads and writes to the upper bits of the unified counter. If CONFIG[UNIFY] = 0, read or write the 16-bit H counter value, and if CONFIG[UNIFY] = 1, read or write the upper 16 bits of the 32-bit unified counter.
15-0 CTR_L	Counter Low Allows reads and writes to the lower bits of the unified counter. If CONFIG[UNIFY] = 0, read or write the 16-bit L counter value, and if CONFIG[UNIFY] = 1, read or write the lower 16 bits of the 32-bit unified counter.

64.7.3.10 State Variable (STATE)

Offset

Register	Offset
STATE	44h

Function

Specifies high and low state variables.

Each group of enabled and disabled events is assigned a number called the state variable. For example, a state variable with a value of 0 could have events 0, 2, and 3 enabled and all other events disabled. A state variable with the value of 1 could have events 1, 4, and 5 enabled and all others disabled.

NOTE

Event n State (EV0_STATE - EV15_STATE) defines the event that is enabled in each group.

You can read the state associated with a counter at any time, and are allowed to write to STATE[STATE_L], STATE[STATE_H], or the unified register only when the corresponding counter is halted (CTRL[HALT_L] and CTRL[HALT_H] = 1).

The state variable is the main feature that distinguishes SCT from other counters, timers, and PWM blocks. Events can be made to occur only in certain states, and these events can, in turn, perform the following actions:

- Set and clear outputs
- Limit, stop, and start the counter
- Cause interrupts and DMA requests
- Modify the state variable

The application completely controls the value of a state variable. If an application does not use states, the value of the state variable remains zero, which is the default value.

You can use a state variable to track and control multiple cycles of the associated counter in any desired operational sequence.

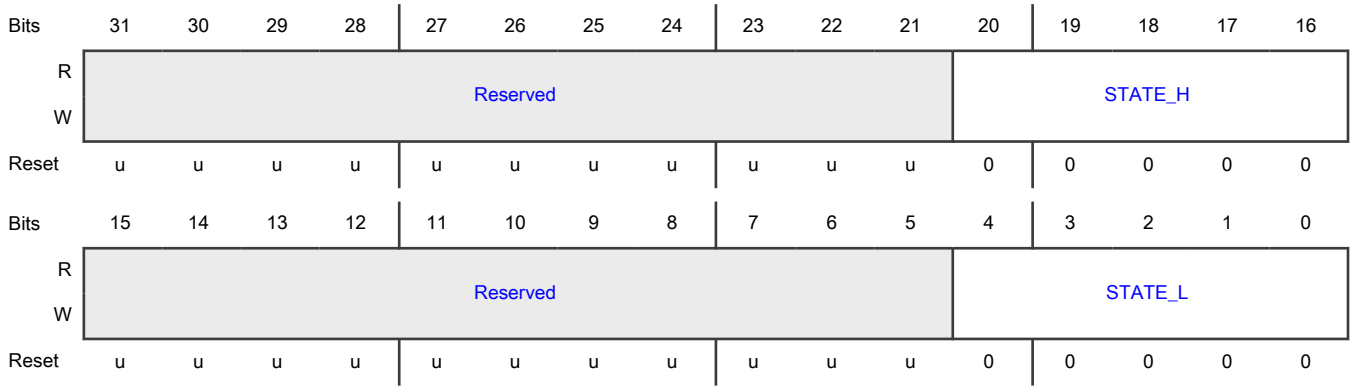
EV*n*_CTRL[STATEV] and EV*n*_CTRL[STATELD] of all defined events set all possible values for the state variable. The change of the state variable during multiple counter cycles reflects how the associated state machine moves from one state to the next.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L bits are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, STATE_L and STATE_H.

You can read or write to both the _L and _H registers individually.

Diagram



Fields

Field	Function
31-21 —	Reserved
20-16 STATE_H	State Variable High
15-5 —	Reserved
4-0 STATE_L	State Variable Low

64.7.3.11 Input State (INPUT)

Offset

Register	Offset
INPUT	48h

Function

Indicates the state and form of SCT inputs.

You can read the state of SCT inputs, in slightly different forms, using this register:

- The AIN fields of this register indicate the state of the inputs captured on the last rising edge of the SCT clock. This corresponds to a nearly direct read-out of the input but can cause spurious fluctuations in case of an asynchronous input signal.
- The SIN fields of this register indicate the form of the inputs used for event detection. This may include additional stages of synchronization, depending on what is specified for that input in CONFIG[INSYNC]:
  - If CONFIG[INSYNC] = 1, the input is triple-synchronized to the SCT clock, resulting in a stable signal that is delayed by three SCT clock cycles.

— If CONFIG[INSYNC] = 0, the value of SIN is identical to the value of AIN.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SIN15	SIN14	SIN13	SIN12	SIN11	SIN10	SIN9	SIN8	SIN7	SIN6	SIN5	SIN4	SIN3	SIN2	SIN1	SIN0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AIN15	AIN14	AIN13	AIN12	AIN11	AIN10	AIN9	AIN8	AIN7	AIN6	AIN5	AIN4	AIN3	AIN2	AIN1	AIN0
W																
Reset	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u

Fields

Field	Function
31 SIN15	Input 15 state
30 SIN14	Input 14 state
29 SIN13	Input 13 state
28 SIN12	Input 12 state
27 SIN11	Input 11 state
26 SIN10	Input 10 state
25 SIN9	Input 9 state
24 SIN8	Input 8 state
23 SIN7	Input 7 state
22	Input 6 state

Table continues on the next page...

Table continued from the previous page...

Field	Function
SIN6	
21 SIN5	Input 5 state
20 SIN4	Input 4 state
19 SIN3	Input 3 state
18 SIN2	Input 2 state
17 SIN1	Input 1 state
16 SIN0	Input 0 state
15 AIN15	Input 15 state
14 AIN14	Input 14 state
13 AIN13	Input 13 state
12 AIN12	Input 12 state
11 AIN11	Input 11 state
10 AIN10	Input 10 state
9 AIN9	Input 9 state
8 AIN8	Input 8 state

Table continues on the next page...

Table continued from the previous page...

Field	Function
7 AIN7	Input 7 state
6 AIN6	Input 6 state
5 AIN5	Input 5 state
4 AIN4	Input 4 state
3 AIN3	Input 3 state
2 AIN2	Input 2 state
1 AIN1	Input 1 state
0 AIN0	Input 0 state

64.7.3.12 Match and Capture Register Mode (REGMODE)

Offset

Register	Offset
REGMODE	4Ch

Function

Specifies whether each register acts as a match register or capture register. SCT contains multiple match and capture registers, and each register includes an accompanying register that functions as a reload register when the primary register is used as a match or capture-control register.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only REGMODE[REGMOD_L*n*] is used. REGMODE[REGMOD_H*n*] is used only when CONFIG[UNIFY] = 0.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, REGMODE_L and REGMODE_H. You can read or write to both the _L and _H registers individually. REGMODE[REGMOD_L*n*] controls the lower match and capture registers, and REGMODE[REGMOD_H*n*] controls the higher or upper match and capture registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...	REGMOD_...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 REGMOD_Hn	Register Mode High Controls match and capture register *n*. If this field = 0, register *n* operates as a match register, and if this field = 1, register *n* operates as a capture register. 0b - Match 1b - Capture
15-0 REGMOD_Ln	Register Mode Low Controls match and capture register *n*. If this field = 0, register *n* operates as a match register, and if this field = 1, register *n* operates as a capture register. 0b - Match 1b - Capture

64.7.3.13 Output State (OUTPUT)

Offset

Register	Offset
OUTPUT	50h

Function

Determines SCT output state.

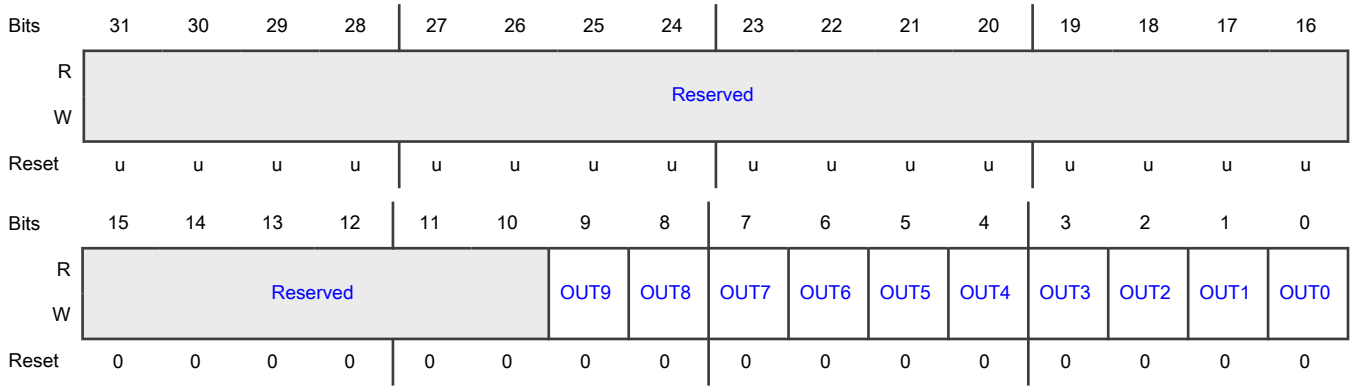
Each SCT output has a corresponding field in this register to allow you to control the output state directly or read its current state.

While the counter is running, only events can set, clear, and toggle these outputs. However, using this register, you can write to any of the output registers when both the counters are halted to control the outputs directly. Writing to the OUT register is only allowed when all counters (L-counter, H-counter, or unified counter) are halted (CTRL[HALT] = 1).

You can read this register at any time to sense the state of the outputs.

Only word writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-10 —	Reserved
9-0 OUTn	Output Low and High Specifies the SCT output state. The number of bits equals the number of outputs in this SCT. 0b - Forces the corresponding output low 1b - Forces the corresponding output high

64.7.3.14 Output Counter Direction Control (OUTPUTDIRCTRL)

Offset

Register	Offset
OUTPUTDIRCTRL	54h

Function

Specifies, for each output, the impact of the counting direction on the meaning of set and clear operations on the output, when using Bidirectional mode. The purpose of this register is to facilitate the creation of center-aligned output waveforms without the need to define additional events.

Only word writes are permitted to this register; attempting to write a halfword value results in a bus error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												SETCLR9		SETCLR8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SETCLR7		SETCLR6		SETCLR5		SETCLR4		SETCLR3		SETCLR2		SETCLR1		SETCLR0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-18: SETCLR9 17-16: SETCLR8 15-14: SETCLR7 13-12: SETCLR6 11-10: SETCLR5 9-8: SETCLR4 7-6: SETCLR3 5-4: SETCLR2 3-2: SETCLR1 1-0: SETCLR0	Set and Clear Operation on Output Specifies the set and clear operation on output *n* depending on the counter direction. 00b - Not dependent on the direction of any counter 01b - Reversed when counter L or the unified counter is counting down 10b - Reversed when counter H is counting down (do not use this value when CONFIG[UNIFY] = 1) 11b - Reserved (do not program this value)

64.7.3.15 Output Conflict Resolution (RES)

Offset

Register	Offset
RES	58h

Function

Specifies the action to be taken if multiple events (or even the same event) dictate that a given output must be both set and cleared at the same time.

To enable an event to toggle an output each time the event occurs, write 1 to the fields for that event in both Output n Set (OUT0_SET - OUT9_SET) and Output n Clear (OUT0_CLR - OUT9_CLR), and write 0x3 to RES[O*n*RES].

Only word writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												O9RES		O8RES	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	O7RES		O6RES		O5RES		O4RES		O3RES		O2RES		O1RES		O0RES	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-18: O9RES 17-16: O8RES 15-14: O7RES 13-12: O6RES 11-10: O5RES 9-8: O4RES 7-6: O3RES 5-4: O2RES 3-2: O1RES 1-0: O0RES	Output Resolution Specifies the effect of simultaneous set and clear on output *n*. 00b - No change 01b - Set output (or clear, based on OUTPUTDIRCTRL[SETCLRn]) 10b - Clear output (or set, based on OUTPUTDIRCTRL[SETCLRn]) 11b - Toggle output

64.7.3.16 DMA Request 0 (DMAREQ0)

Offset

Register	Offset
DMAREQ0	5Ch

Function

Enables DMA requests to be generated when a particular event occurs or when counter match registers are loaded from the reload registers. SCT includes two DMA request outputs.

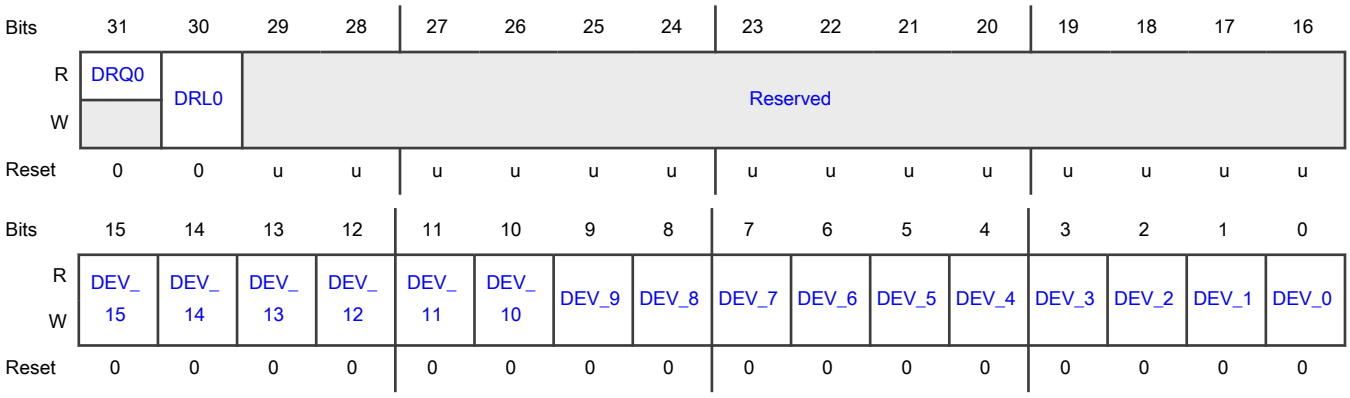
The DMA request registers are word-write only. Attempting to write a halfword value to these registers results in a bus error.

Event-generated DMA requests are particularly useful for launching DMA activity to or from other peripherals under SCT control.

NOTE

In this chip, SCT DMA requests are connected to the DMA trigger inputs, and the DMA trigger input mux registers control them.

Diagram



Fields

Field	Function
31 DRQ0	DMA Request 0 State Indicates the state of DMA request 0. **NOTE** If the related DMA channel is enabled and properly set up, it is unlikely for you to see this flag. The DMA service rapidly clears this field. If the flag remains set, there could be an issue with the DMA setup.
30 DRL0	DMA Request Low 0 Triggers DMA request 0. Writing 1 to this field triggers DMA request 0 when it loads MATCH*n*[MATCH_L] or the unified registers, using MATCHREL*n*[RELOAD_L] or the unified registers.
29-16 —	Reserved
15-0 DEV_n	DMA Request Event Triggers DMA request 0 using event *n*. If bit *n* = 1, event *n* triggers DMA request 0 (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.17 DMA Request 1 (DMAREQ1)

Offset

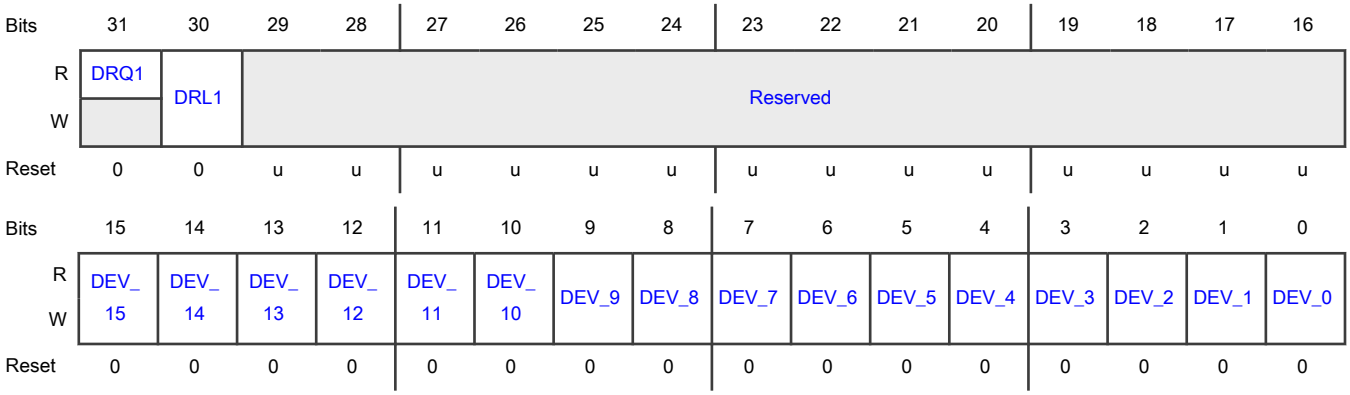
Register	Offset
DMAREQ1	60h

Function

Specifies DMA request 1.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31 DRQ1	DMA Request 1 State Specifies the state of DMA request 1. **NOTE** If the related DMA channel is enabled and properly set up, it is unlikely for you to see this flag, which the DMA service clears rapidly. If the flag remains set, there could be an issue with the DMA setup.
30 DRL1	DMA Request Low 1 Triggers DMA request 1. Writing 1 to this field triggers DMA request 1 when it loads MATCH*n*[MATCH_L] or the unified registers, using MATCHREL*n*[RELOAD_L] or the unified registers.
29-16 —	Reserved
15-0	DMA Request Event

Table continues on the next page...

Table continued from the previous page...

Field	Function
DEV_n	Triggers DMA request using event *n*. If bit *n* = 1, event *n* triggers DMA request 1 (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT.

64.7.3.18 Event Interrupt Enable (EVEN)

Offset

Register	Offset
EVEN	F0h

Function

Enables flags to request an interrupt if EVFLAG[FLAG*n*] = 1.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEN15	IEN14	IEN13	IEN12	IEN11	IEN10	IEN9	IEN8	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 IENn	Event Interrupt Enable Enables event interrupts. 0b - Disables 1b - Enables

64.7.3.19 Event Flag (EVFLAG)

Offset

Register	Offset
EVFLAG	F4h

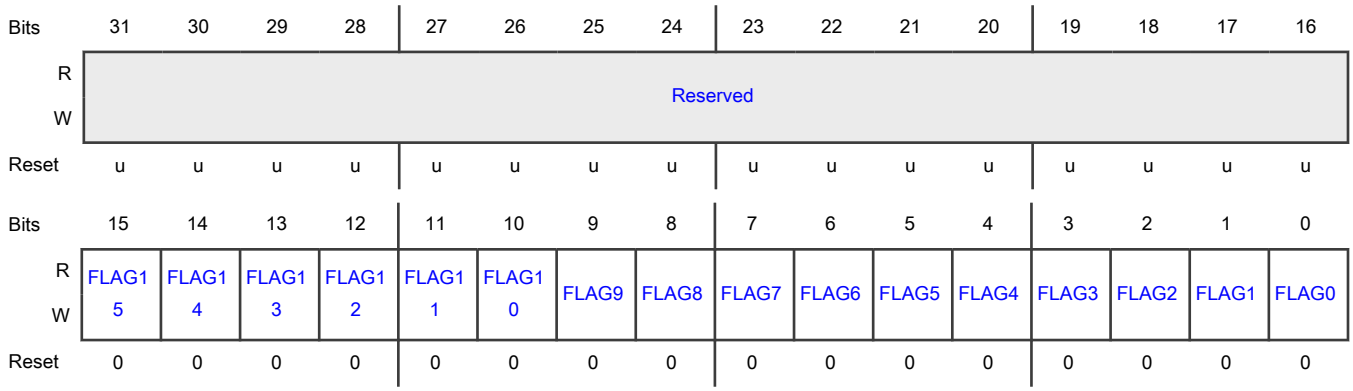
Function

Records events.

Writing 1 to this register clears the corresponding flags and negates the SCT interrupt request if all enabled flags are 0.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 FLAGn	Event Flag Specifies event flag *n*. Bit *n* is 1 if event *n* has occurred since reset or a 1 was last written to this bit (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of events in this SCT. 0b - No flag 1b - Event n flag

64.7.3.20 Conflict Interrupt Enable (CONEN)

Offset

Register	Offset
CONEN	F8h

Function

Enables no-change conflict events specified in Output Conflict Resolution (RES) to generate an interrupt request.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						NCEN 9	NCEN 8	NCEN 7	NCEN 6	NCEN 5	NCEN 4	NCEN 3	NCEN 2	NCEN 1	NCEN 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-10 —	Reserved
9-0 NCENn	No Change Conflict Event and Interrupt Enable Specifies whether an interrupt is generated. SCT requests an interrupt when bit *n* of this register and Conflict Flag (CONFLAG) are both = 1 (output 0 = bit 0, output 1 = bit 1, and so on). 0b - No interrupt 1b - Interrupt

64.7.3.21 Conflict Flag (CONFLAG)

Offset

Register	Offset
CONFLAG	FCh

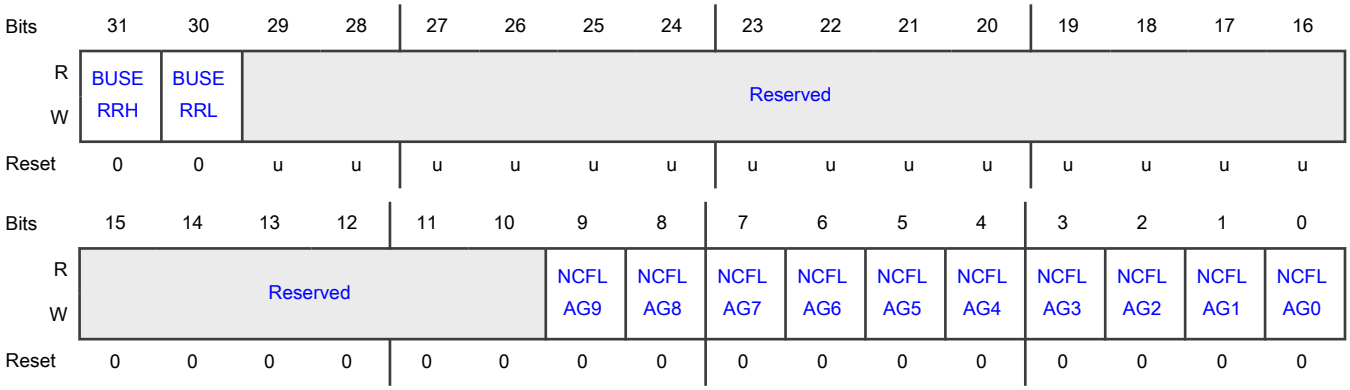
Function

Records a no-change conflict occurrence and also provides details of any bus errors.

Writing 1 to the fields of this register clears the corresponding read bits and negates the SCT interrupt request if all enabled flags are zero.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31 BUSERRH	Bus Error High Specifies whether the bus error is low or unified. The most recent bus error from this SCT involved writing CTR H, STATE H, MATCH H, or the Output register when the H counter was not halted.
30 BUSERRL	Bus Error Low or Unified Specifies whether the bus error is low or unified. The most recent bus error from this SCT involved writing to CTR L/Unified, STATE L/Unified, MATCH L/Unified, or the Output register when the L/U counter was not halted. A word write to certain L and H registers can be half successful and half unsuccessful.
29-10 —	Reserved
9-0 NCFLAGn	No Change Conflict Event Flag Specifies whether a no-change conflict event occurred. 0b - Did not occur 1b - Occurred

64.7.3.22 Capture Value (CAP0 - CAP15)

Offset

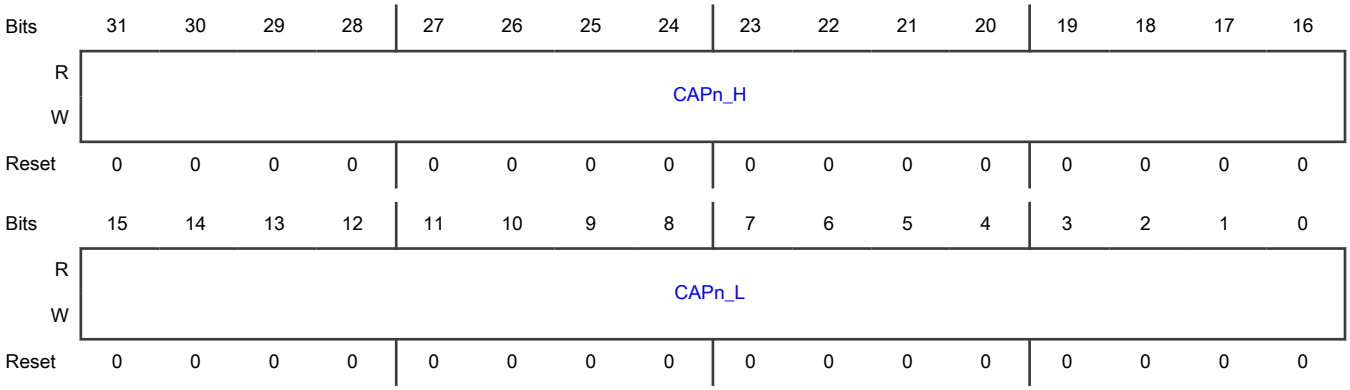
For n = 0 to 15:

Register	Offset
CAPn	100h + (n × 4h)

Function

Allows you to record counter values when the events selected by the corresponding capture control registers occur.

Diagram



Fields

Field	Function
31-16 CAPn_H	Capture High Captures the higher counter value. If CONFIG[UNIFY] = 0, read the 16-bit counter value at which this register was last captured, and if CONFIG[UNIFY] = 1, read the upper 16 bits of the 32-bit value at which this register was last captured.
15-0 CAPn_L	Capture Low Captures the lower counter value. If CONFIG[UNIFY] = 0, read the 16-bit counter value at which this register was last captured, and if CONFIG[UNIFY] = 1, read the lower 16 bits of the 32-bit value where this register was last captured.

64.7.3.23 Match Value (MATCH0 - MATCH15)

Offset

For n = 0 to 15:

Register	Offset
MATCHn	100h + (n × 4h)

Function

Is compared to counters to help create events.

You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 0, the L and H registers are independently compared to the L and H counters.
- If CONFIG[UNIFY] = 1, the combined L and H registers hold a 32-bit value that is compared to the unified counter. A match can only occur in a clock for which the counter is running (Stop Event Select (STOP) and Halt Event Select (HALT) are 0).

You can read this register at any time. Writing to MATCH[MATCH*n*_L], MATCH[MATCH*n*_H], or the unified register is only allowed when the corresponding counter is halted (CTRL[HALT_L] and CTRL[HALT_H] are 1). Match events occur in the SCT clock for which the counter is (or would be) incremented to the next value. When a match event limits its counter, the value in this register is the last value of the counter before it is cleared to zero (or decremented if BIDIR is 1).

There is no “write-through” from reload registers to match registers. Before starting a counter, you can write one value to this register, used in the first cycle of the counter, and a different value to the corresponding match reload register used in the second cycle.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCHn_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCHn_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 MATCHn_H	Match High Compares upper bits to the unified counter. If CONFIG[UNIFY] = 0, read or write the 16-bit value to be compared to the H counter. If CONFIG[UNIFY] = 1, read or write the upper 16 bits of the 32-bit value to be compared to the unified counter.
15-0 MATCHn_L	Match Low Compares lower bits to the unified counter. If CONFIG[UNIFY] = 0, read or write the 16-bit value to be compared to the lower counter. If CONFIG[UNIFY] = 1, read or write the lower 16 bits of the 32-bit value to be compared to the unified counter.

64.7.3.24 Fractional Match (FRACMAT0 - FRACMAT5)

Offset

Register	Offset
FRACMAT0	140h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
FRACMAT1	144h
FRACMAT2	148h
FRACMAT3	14Ch
FRACMAT4	150h
FRACMAT5	154h

Function

Provides a higher average resolution over time by applying a dither pattern as described in Dithering. This dither pattern results in delaying recognition of a match for one counter clock for *n* (0 to 15) out of every 16 counter cycles. You must program the value of *n* in these fractional match registers, which are provided for up to first six of the match registers.

You can read fractional match registers at any time, but can write to FRACMAT*n*[FRACMAT_L], FRACMAT*n*[FRACMAT_H], or the unified register only when the corresponding counter is halted (CTRL[HALT_L] and CTRL[HALT_H] are 1).

Each fractional match register has an associated fractional match reload register. The contents of these reload registers are transferred into the fractional match registers at the start of every new SCT counter cycle unless CONFIG[NORELOAD] = 1 for the appropriate half counter.

You can write to the reload registers at any time, regardless of whether the counter is running.

There is no write-through from the fractional match reload registers to the fractional match registers. Before starting a counter, you can write a value to the fractional match register that is used in the first cycle or period of operation, and a different value to the corresponding fractional match reload register that is used in the second cycle or period.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												FRACMAT_H			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												FRACMAT_L			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-16	Fractional Match High

Table continues on the next page...

Table continued from the previous page...

Field	Function
FRACMAT_H	Allows reads or writes to upper bits for fractional match. If CONFIG[UNIFY] = 0, read or write a 4-bit value specifying the dither pattern to be applied to the corresponding MATCH*n*[MATCH_H] field (*n* = 0 to 5).
15-4 —	Reserved
3-0 FRACMAT_L	Fractional Match Low Allows reads or writes to lower bits for fractional match. If CONFIG[UNIFY] = 0, read or write a 4-bit value specifying the dither pattern to be applied to the corresponding MATCH*n*[MATCH_L] field (*n* = 0 to 5). If CONFIG[UNIFY] = 1, the value applies to the unified, 32-bit fractional match register.

64.7.3.25 Capture Control (CAPCTRL0 - CAPCTRL15)

Offset

For n = 0 to 15:

Register	Offset
CAPCTRLn	200h + (n × 4h)

Function

Controls the events that cause loading of the corresponding capture register from the counter.

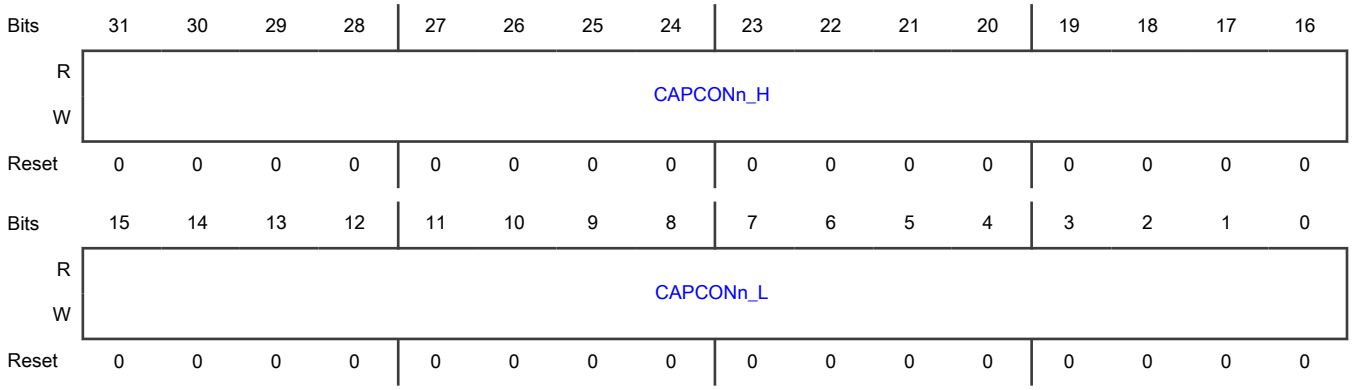
You can use this register depending on the configuration of CONFIG[UNIFY]:

- If CONFIG[UNIFY] = 1, only the _L fields are used.
- If CONFIG[UNIFY] = 0, you can write to this register as two registers, CAPCTRL*n*_L and CAPCTRL*n*_H. You can read both or write to the _L and _H registers individually.

Based on a selected event, the capture registers can be loaded with the current counter value when the event occurs.

Each capture control register (L, H, or unified 32-bit) controls which events cause the load of the corresponding capture register from the counter.

Diagram



Fields

Field	Function
31-16 CAPCONn_H	Capture Control High Captures higher bits. If bit *n* = 1, event *n* causes CAP*n*[CAP_H] (CONFIG[UNIFY] = 0) to be loaded (event 0 = bit 16, event 1 = bit 17, and so on). Number of bits = number of match and captures in SCT.
15-0 CAPCONn_L	Capture Control Low Captures lower bits. If bit *n* = 1, event *n* causes CAP*n*[CAP_L] (CONFIG[UNIFY] = 0) or CAP*n* (CONFIG[UNIFY] = 1) to be loaded (event 0 = bit 0, event 1 = bit 1, and so on). Number of bits = number of match and captures in SCT.

64.7.3.26 Match Reload Value (MATCHREL0 - MATCHREL15)

Offset

For n = 0 to 15:

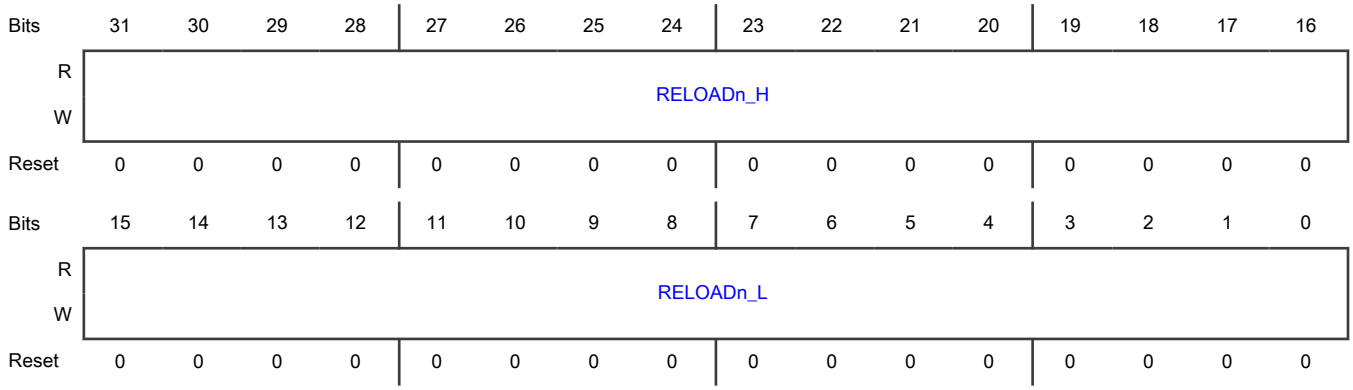
Register	Offset
MATCHRELn	200h + (n × 4h)

Function

Reloads a match register (L, H, or unified 32-bit counter) from its corresponding reload register at the start of each new counter cycle, which is when:

- CTRL[BIDIR_H] and CTRL[BIDIR_L] are 0 and the counter is cleared to zero upon reaching its limit condition.
- CTRL[BIDIR_H] and CTRL[BIDIR_L] are 1 and the counter counts down to 0, unless CONFIG[NORELOAD_H] and CONFIG[NORELOAD_L] are 1.

Diagram



Fields

Field	Function
31-16 RELOADn_H	Reload High Specifies the value to be loaded into MATCH*n*[MATCH_H]. If CONFIG[UNIFY] = 0, this field specifies the 16-bit value to be loaded into MATCH*n*[MATCH_H]. If CONFIG[UNIFY] = 1, the field specifies the upper 16 bits of the 32-bit value to be loaded into MATCH*n*[MATCH_H].
15-0 RELOADn_L	Reload Low Specifies the value to be loaded into MATCH*n*[MATCH_L]. If CONFIG[UNIFY] = 0, this field specifies the 16-bit value to be loaded into MATCH*n*[MATCH_L]. If CONFIG[UNIFY] = 1, the field specifies the lower 16 bits of the 32-bit value to be loaded into MATCH*n*[MATCH_L].

64.7.3.27 Fractional Match Reload (FRACMATREL0 - FRACMATREL5)

Offset

Register	Offset
FRACMATREL0	240h
FRACMATREL1	244h
FRACMATREL2	248h
FRACMATREL3	24Ch
FRACMATREL4	250h
FRACMATREL5	254h

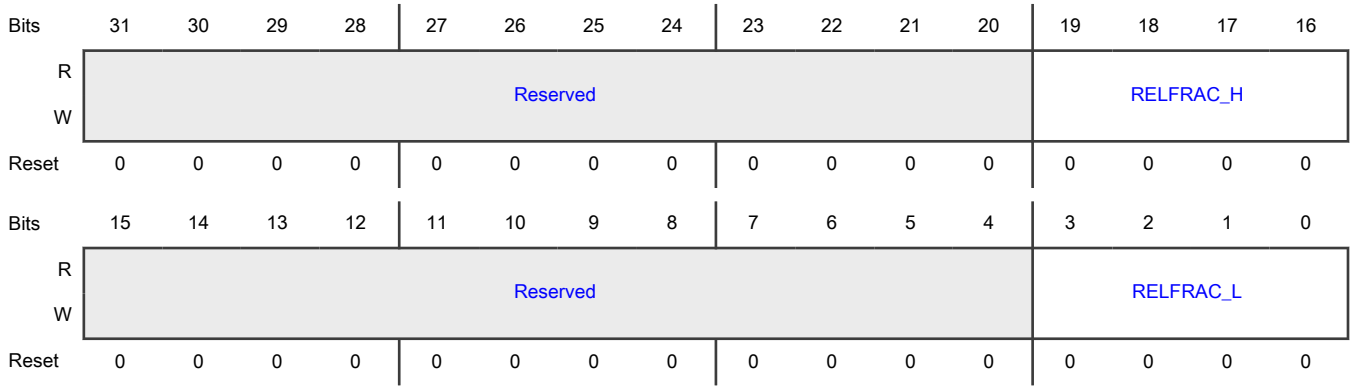
Function

Reloads a fractional match register (_L, _H, or the unified 32-bit counter) from the corresponding fractional match reload register at the start of each new counter cycle.

A fractional match register is loaded from the corresponding fractional match reload register at the start of each new counter cycle, which means when:

- CTRL[BIDIR] = 0 and the counter is cleared to zero upon reaching its limit condition.
- CTRL[BIDIR] = 1 and the counter counts down to 0, unless the appropriate CONFIG[NORELOAD] field = 1.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-16 RELFRAC_H	Reload Fractional Match High Specifies the value to be loaded into FRACMAT*n*[FRACMAT_H]. If CONFIG[UNIFY] = 0, read or write the 4-bit value to be loaded into FRACMAT*n*[FRACMAT_H]. If CONFIG[UNIFY] = 1, read or write the upper 4 bits with the 4-bit value to be loaded into FRACMAT*n*[FRACMAT_H].
15-4 —	Reserved
3-0 RELFRAC_L	Reload Fractional Match Low Specifies the value to be loaded into FRACMAT*n*[FRACMAT_L]. If CONFIG[UNIFY] = 0, read or write the 4-bit value to be loaded into FRACMAT*n*[FRACMAT_L]. If CONFIG[UNIFY] = 1, read or write the lower 4 bits to be loaded into FRACMAT*n*[FRACMAT_L].

64.7.3.28 Event n State (EV0_STATE - EV15_STATE)

Offset

For n = 0 to 15:

Register	Offset
EVn_STATE	300h + (n × 8h)

Function

Specifies event *n* state.

You can enable each event in some contexts (or states) and disable it in others. Each event defined in Event n Control (EV0_CTRL - EV15_CTRL) has an associated Event Interrupt Enable (EVEN) that can enable or disable the event for each available state.

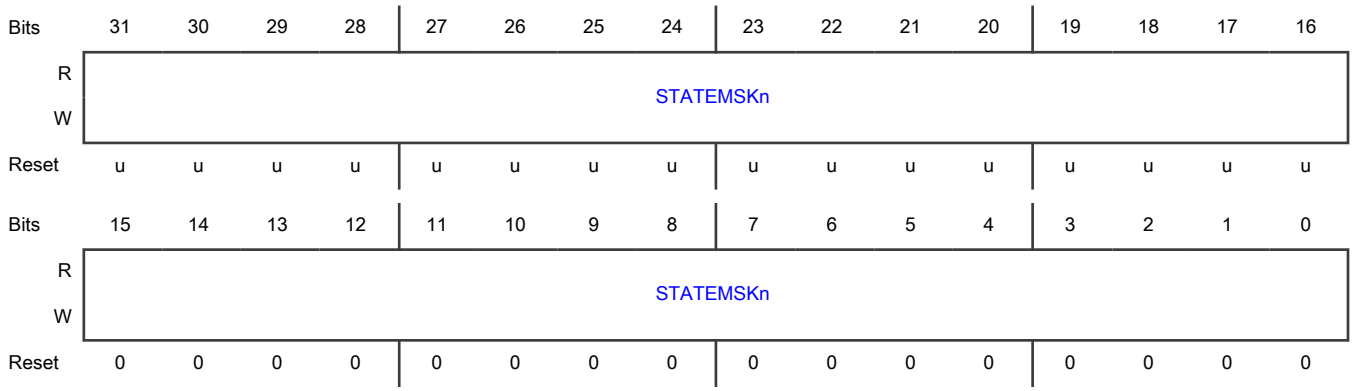
Each event also has an associated Event n State (EV0_STATE - EV15_STATE) that allows this event to happen in one or more states of the counter selected by the corresponding EV*n*_CTRL[HEVENT] field.

An event *n* is disabled when its corresponding Event n State (EV0_STATE - EV15_STATE) contains all zeros because it is masked regardless of the current state.

In simple applications that do not use states, write 0x01 to this register to enable each event in exactly one state. Because the state does not change (which means, the state variable always remains at its reset value of 0), writing 0x01 permanently enables this event.

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31-0 STATEMSKn	Event State Mask Specifies the state in which event *n* happens. If bit *m* = 1, event *n* happens in state *m* of the counter selected by the HEVENT field (*n* = event number, *m* = state number; state 0 = bit 0, state 1= bit 1, and so on). Number of bits = number of states in this SCT.

64.7.3.29 Event n Control (EV0_CTRL - EV15_CTRL)

Offset

For n = 0 to 15:

Register	Offset
EVn_CTRL	304h + (n × 8h)

Function

Defines the conditions for an event to occur, based on the counter values or input and output states.

After you configure an event, it can be selected to trigger multiple actions (for example, stopping the counter and toggling an output) unless the event is blocked in the current SCT state, or the counter is halted. To block a particular event from occurring, use Event n State (EV0_STATE - EV15_STATE). To block all events for a given counter, write 1 to CTRL[HALT] or select an event to halt the counter.

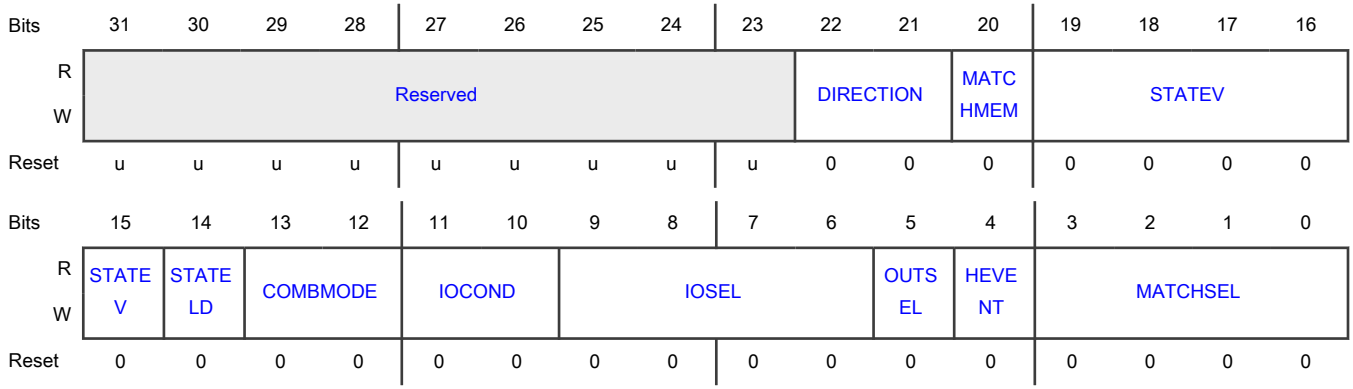
An event can be programmed to occur based on a selected input or output edge or level and/or based on its counter value, matching a selected match register. In Bidirectional mode, you can enable events based on the direction of count.

If CONFIG[UNIFY] = 0, each event is associated with a particular counter in EV*n*_CTRL[HEVENT]. An event is permanently disabled when its event state mask register contains all 0s.

Each event can modify its counter STATE value. If more than one event associated with the same counter occurs in a given clock cycle, only the state change specified for the highest-numbered event among them takes place. Other actions dictated by any simultaneously occurring events all take place.

Only word writes are permitted to this register. Attempting to write a halfword value results in a bus error.

Diagram



Fields

Field	Function
31-23 —	Reserved
22-21 DIRECTION	Direction Specifies the direction for event generation. This field only applies when the counters are operating in BIDIR mode. If CTRL[BIDIR] = 0, SCT ignores this field. 00b - Direction independent (event triggered regardless of the count direction) 01b - Counting up (event triggered only during up-counting when CTRL[BIDIR] = 1) 10b - Counting down (event triggered only during down-counting when CTRL[BIDIR] = 1) 11b - Reserved
20 MATCHMEM	Match Mem If this field = 1 and EV*n*_CTRL[COMBMODE] specifies a match component to the triggering of this event, a match is considered to be active whenever the counter value is greater than or equal to the value specified

Table continues on the next page...

Table continued from the previous page...

Field	Function
	in the match register when counting up, and less than or equal to the match value when counting down. If this field = 0, a match is only active during the cycle when the counter is equal to the match value.
19-15 STATEV	State Value Specifies the state value. The state value is loaded into or added to the state that EV*n*_CTRL[HEVENT] selects, depending on EV*n*_CTRL[STATELD], when this event is the highest-numbered event occurring for that state. If both the STATELD and STATEV fields are 0, there is no change to the value of STATE.
14 STATELD	State Load Controls how EV*n*_CTRL[STATEV] modifies the state that EV*n*_CTRL[HEVENT] selects when this event is the highest-numbered event occurring for that state. 0b - Value of STATEV added to that of STATE (the carry out is ignored) 1b - Value of STATEV loaded into that of STATE
13-12 COMBMODE	Combination Mode Selects how the specified match and I/O condition are used and combined. 00b - OR (the event occurs when either the specified match or I/O condition occurs) 01b - MATCH (uses the specified match only) 10b - IO (uses the specified I/O condition only) 11b - AND (the event occurs when the specified match and I/O condition occur simultaneously)
11-10 IOCOND	Input or Output Condition Selects the I/O condition for event *n*. The detection of edges on outputs lags the conditions that switch the outputs by one SCT clock. To guarantee proper edge and state detection, an input must have a minimum pulse width of at least one SCT clock period. 00b - Low 01b - Rise 10b - Fall 11b - High
9-6 IOSEL	Input or Output Signal Select Selects the input or output signal number associated with this event, if any. Do not select an input in this register if CONFIG[CLKMODE] is 1x. In this case, the clock input is an implicit ingredient of every event.
5 OUTSEL	Input and Output Select Selects the inputs or outputs that EV*n*_CTRL[IOSEL] specifies.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Inputs 1b - Outputs
4 HEVENT	High Event Selects the low and high counters. Do not write 1 to this field if CONFIG[UNIFY] = 1. 0b - Low counter (selects the L state and the L match register that the MATCHSEL field specifies) 1b - High counter (selects the H state and the H match register that the MATCHSEL field specifies)
3-0 MATCHSEL	Match Select Selects the match register associated with this event, if any. A match can occur only when the counter that EV*n*_CTRL[HEVENT] selects is running.

64.7.3.30 Output n Set (OUT0_SET - OUT9_SET)

Offset

For n = 0 to 9:

Register	Offset
OUTn_SET	500h + (n × 8h)

Function

Selects the events that can set each SCT output.

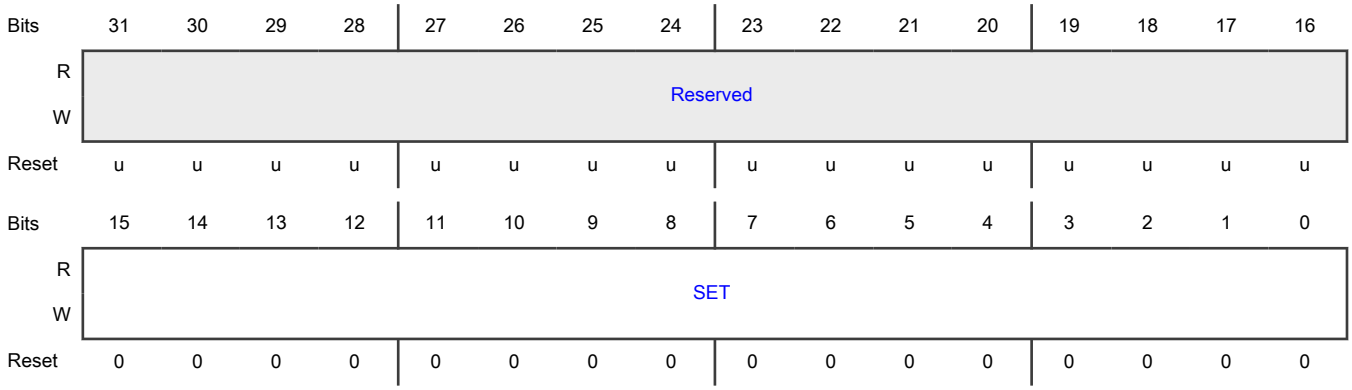
Based on a selected event, each SCT output can be set. There is one output set register for each SCT output. Each bit of an output set register is associated with a different event (bit 0 with event 0, and so on). The selected event can set or clear the output depending on the setting of OUTPUTDIRCTRL[SETCLR*n*]. To define the actual event that sets the output (a match, an I/O pin toggle, and so on), see Event n Control (EV0_CTRL - EV15_CTRL).

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

NOTE

If SCT is operating as two 16-bit counters, events can modify output states only when no counter is halted. This is true regardless of what triggered the event.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 SET	Set Output Selects events to set outputs. If bit *m* = 1, it selects event *m* to set output *n* (or clear it if OUTPUTDIRCTRL[SETCLR*n*] = 0x1 or 0x2). Output 0 = bit 0, output 1 = bit 1, and so on. Number of bits = number of events in this SCT. If the counter is used in Bidirectional mode, it is possible to reverse the action that the output set and clear registers specify when counting down (see Output Counter Direction Control (OUTPUTDIRCTRL) for more information).

64.7.3.31 Output n Clear (OUT0_CLR - OUT9_CLR)

Offset

For n = 0 to 9:

Register	Offset
OUTn_CLR	504h + (n × 8h)

Function

Selects the events that can clear each SCT output.

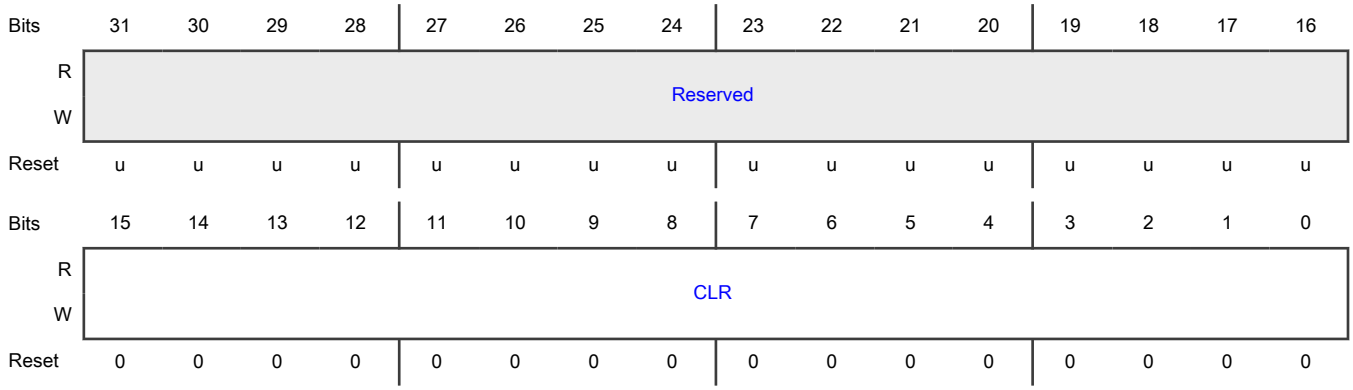
Based on a selected event, each SCT output can be cleared. There is one register for each SCT output that selects which events can clear that output. Each bit of an output clear register is associated with a different event (bit 0 with event 0, and so on). A selected event can clear or set the output depending on the setting of OUTPUTDIRCTRL[SETCLR*n*]. To define the actual event that clears the output (a match, an I/O pin toggle, and so on), see Event n Control (EV0_CTRL - EV15_CTRL).

Only word-writes are permitted to this register. Attempting to write a halfword value results in a bus error.

NOTE

If SCT is operating as two 16-bit counters, events can modify output states only when no counter is halted. This is true regardless of what triggered the event.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CLR	Clear Output Selects events to clear outputs. If bit *m* = 1, it selects event *m* to clear output *n* (or set it if OUTPUTDIRCTRL[SETCLR*n*] = 0x1 or 0x2). Event 0 = bit 0; event 1 = bit 1, and so on. Number of bits = number of events in this SCT. If the counter is used in Bidirectional mode, it is possible to reverse the action specified by the output set and clear registers when counting down (see Output Counter Direction Control (OUTPUTDIRCTRL) for more information).

Chapter 65
Multi-Rate Timer (MRT)

65.1 Chip-specific MRT information

Table 605. Reference links to related information

Topic	Related module	Reference
Full description	MRT	MRT
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

65.1.1 Module instances

This device has two instances of the MRT module, MRT0 and MRT1.

Table 606. MRT instances

	MRT0	MRT1
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

65.1.2 Chip-specific MRT initialization

To initialize the MRT module:

1. Enable the clock to MRT0 via CLKCTL0_PSCCTL2 register and enable the clock to MRT1 via CLKCTL1_PSCCTL1 register.
2. Clear the MRT0 reset via RSTCTL0_PRSTCTL3[MRT0] and clear the MRT1 reset via RSTCTL2_PRSTCTL0[MRT1].
3. Ensure that the MRT can issue an interrupt in the chip. See registers in INPUTMUX:
   - MRT0: HIFI4_INTERRUPT0 - HIFI4_INTERRUPT26, EZH_TRIG0 - EZH_TRIG7 in INPUTMUX0
   - MRT1: HIFI1_INTERRUPT0 - HIFI1_INTERRUPT26 in INPUTMUX1

65.2 Overview

MRT provides a repetitive interrupt timer with 4 channels.

65.2.1 Block diagram

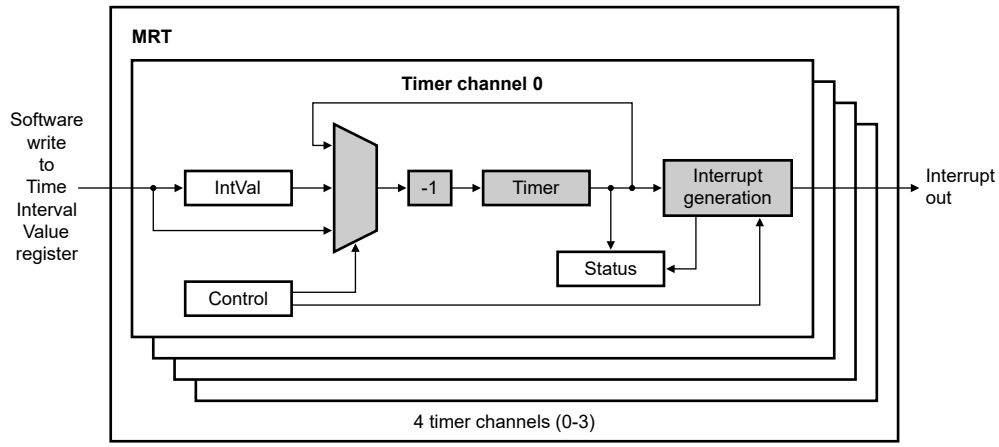


Figure 271. Block diagram

65.2.2 Features

- 24-bit interrupt timer.
- There are 4 channels independently counting down from individually set values.
- Repeat Interrupt, One-Shot Interrupt, and One-Shot Stall modes.
- You can program individual time intervals for each channel.
- Each channel operates independently of the other channels.

65.3 Functional description

The following sections discuss various modes that affect the functionality of MRT.

65.3.1 Operating modes

MRT has the following three operating modes:

- Repeat Interrupt mode
- One-Shot Interrupt mode
- One-Shot Stall mode

65.3.1.1 Repeat Interrupt mode

• Generates repeated interrupts after a selected time interval.

• Gets used for application-based Pulse Width Modulation (PWM) or Pulse Position Modulation (PPM).

When TIMER*n* is in idle state, writing a nonzero value (IVALUE) to INTVAL*n* immediately loads the time interval value (IVALUE -1). Then the timer begins to count down from this value.

After the timer reaches 0

- an interrupt is generated.
- the value in INTVAL*n* (IVALUE -1) is automatically reloaded, and
- the timer starts to count down again.

When the timer is running in Repeat Interrupt mode, you can perform various interval timer actions (change or stop):

Table 607. Possible actions in Repeat Interrupt mode

Action	To perform action	Timer does this	Interrupt
Change the interval value on the next interval timer cycle.	• Write new value (>0) to INTVAL*n*. • Write 0 to LOAD.	On the next cycle, the timer counts down from the new value.	An interrupt is generated when the timer reaches 0.
Change the interval value immediately (also called "change on-the-fly") in the interval timer.	• Write new value (>0) to INTVAL*n*. • Write 1 to LOAD.	The timer immediately starts to count down from the new timer interval value.	An interrupt is generated when the timer reaches 0.
Stop the interval timer at the end of the time interval.	• Write 0 to INTVAL*n*. • Write 0 to LOAD.	The timer stops at the end of the time interval.	An interrupt is generated when the timer reaches 0.
Stop the interval timer immediately.	• Write 0 to INTVAL*n*. • Write 1 to LOAD.	The timer stops immediately.	No interrupt is generated when INTVAL*n* is written.

65.3.1.2 One-Shot Interrupt mode

• In this mode, the timer generates one interrupt after a one-time count.

• This mode allows you to generate an interrupt at any time.

• Using this mode, you can introduce a specific delay in a software task.

When the timer is in an idle state, writing a nonzero value (IVALUE) to INTVAL*n* immediately loads the time interval value (IVALUE -1), and the timer starts to count down. As soon as the timer reaches 0, the timer stops and an interrupt is generated (and then enters the idle state).

When the timer is running in One-Shot Interrupt mode, you can perform the following interval timer actions (load or stop):

Table 608. Possible actions in One-Shot Interrupt mode

Action	To perform action	Timer does this	Interrupt
Load a timer interval value.	• Write a new time interval value (>0) to INTVAL*n*. • Write 1 to LOAD.	The timer immediately reloads the new time interval, and starts counting down from the new value.	An interrupt is generated when the timer reaches 0.
Stop the interval timer.	• Write a 0 to INTVAL*n*. • Write 1 to LOAD.	The timer immediately stops counting and moves to the idle state.	No interrupt is generated when INTVAL*n* is updated.

65.3.1.3 One-Shot Stall mode

• Used for very short delays (for example, when the delay needed is less than the time it takes to arrive at an interrupt service routine).

• Designed for very low software overhead that requires only a single write to INTVAL*n* (if the channel is already configured for One-Shot stall mode). MRT counts the requested delay when stalling the bus write operation, and then concluding the write operation when the delay has finished. No interrupt or status polling is needed.

Use this mode:

- when a short delay is needed between two software-controlled events.
- when a delay is expected before your application can continue.

The CPU consumes a minimum amount of power during One-Shot Stall mode, because no bus transactions occur during that period.

One-Shot Stall mode provides a minimum amount of time between the execution of the instruction that performs the write to INTVAL*n* and the time that your software resumes execution. Other system events, such as interrupts or other bus managers accessing the APB bus where MRT resides, can cause the delay to be longer.

65.3.2 Clocking

Table 609. MRT clocks

Type of clock	Description
Bus clock (vpb_clk)	Controls the access to the MRT registers and functions the MRT module.

65.3.3 Interrupts

A channel provides 3 different interrupts modes listed below and can select the interrupt mode individually per channel. This interrupt can be programmed with an independent time interval.

- Repeat interrupt
- Shot interrupt
- Shot bus stall interrupt

65.4 External signals

This module has no external signals.

65.5 Initialization

To initialize:

1. Enable the clock to MRT.
2. Deassert MRT reset.
3. Ensure that MRT can issue an interrupt in the chip.

See the Chip-specific section for more details.

65.6 Memory map and register descriptions

65.6.1 MRT register descriptions

65.6.1.1 MRT memory map

MRT0 base address: 4002_D000h

MRT1 base address: 4004_D000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Time Interval Value (INTVAL0)	32	RW	See section

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Timer (TIMER0)	32	R	See section
8h	Control (CTRL0)	32	RW	See section
Ch	Status (STAT0)	32	RW	See section
10h	Time Interval Value (INTVAL1)	32	RW	See section
14h	Timer (TIMER1)	32	R	See section
18h	Control (CTRL1)	32	RW	See section
1Ch	Status (STAT1)	32	RW	See section
20h	Time Interval Value (INTVAL2)	32	RW	See section
24h	Timer (TIMER2)	32	R	See section
28h	Control (CTRL2)	32	RW	See section
2Ch	Status (STAT2)	32	RW	See section
30h	Time Interval Value (INTVAL3)	32	RW	See section
34h	Timer (TIMER3)	32	R	See section
38h	Control (CTRL3)	32	RW	See section
3Ch	Status (STAT3)	32	RW	See section
F0h	Module Configuration (MODCFG)	32	RW	See section
F4h	Idle Channel (IDLE_CH)	32	R	See section
F8h	Global Interrupt Flag (IRQ_FLAG)	32	RW	See section

65.6.1.2 Time Interval Value (INTVAL0 - INTVAL3)

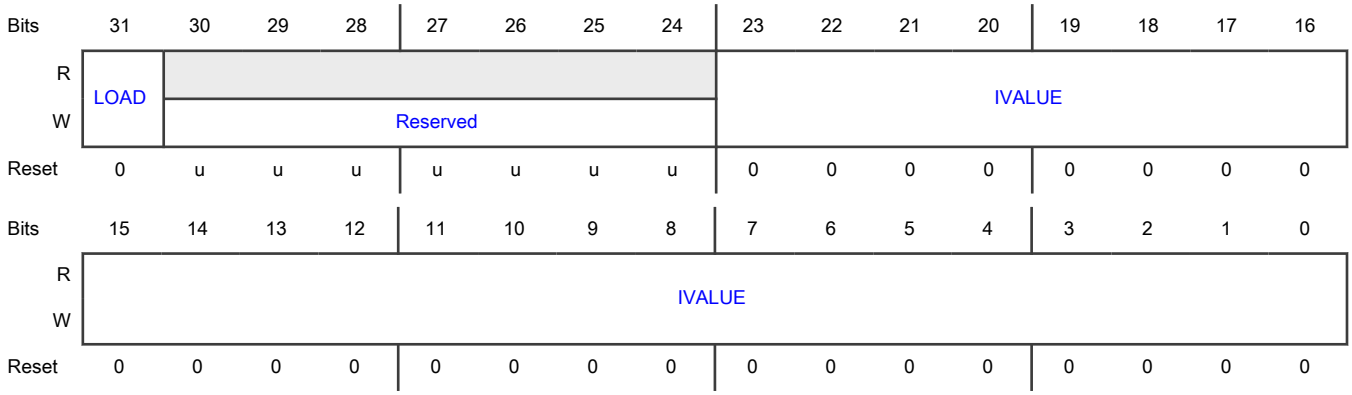
Offset

Register	Offset
INTVAL0	0h
INTVAL1	10h
INTVAL2	20h
INTVAL3	30h

Function

Contains MRT load value and controls how the timer is reloaded. The load value is IVALUE -1.

Diagram



Fields

Field	Function
31 LOAD	Force Load Enable Determines how the timer interval value (IVALUE -1) is loaded into the TIMER*n*. LOAD is write only. Reading LOAD always returns 0. If the repeat mode is selected, the load from the INTVAL*n* to the TIMER*n* is processed at the end of the time interval. The INTVAL*n* interval value (IVALUE -1) is immediately loaded into the TIMER*n* when it is running. 0b - No force load 1b - Force load
30-24 —	Reserved Reserved. Read value is undefined; only 0 should be written.
23-0 IVALUE	Time Interval Load Value. INTVAL*n*[IVALUE] is loaded into TIMER*n*, and MRT channel *n* starts counting down from IVALUE -1. If the timer is idle, then writing a nonzero value to INTVAL*n*[IVALUE] starts the timer immediately. If the timer is running, then writing a 0 to INTVAL*n*[IVALUE] does the following: if LOAD = 1, then the timer stops immediately; if LOAD = 0, then the timer stops at the end of the time interval.

65.6.1.3 Timer (TIMER0 - TIMER3)

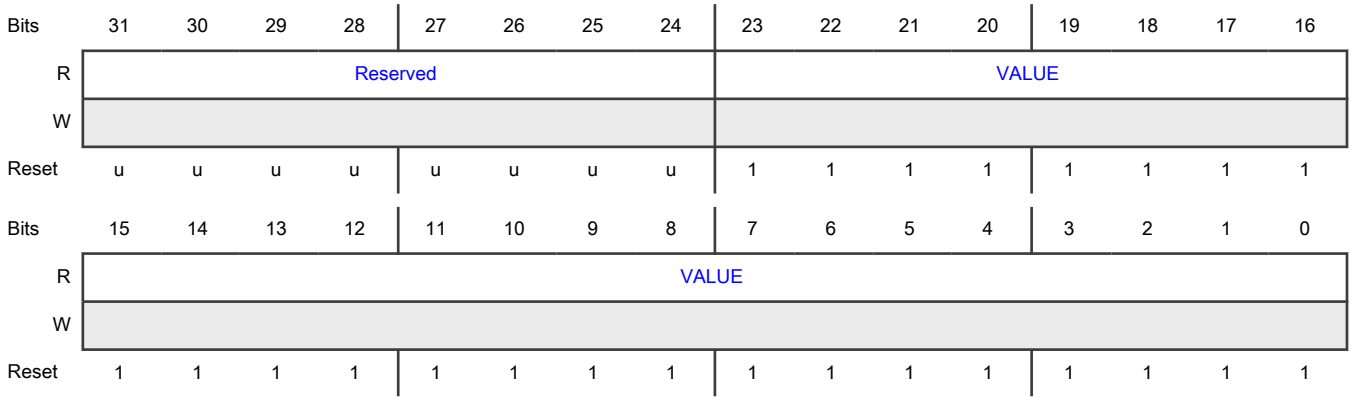
Offset

Register	Offset
TIMER0	4h
TIMER1	14h
TIMER2	24h
TIMER3	34h

Function

Holds the current timer value (the TIMER*n* reads the value of the down counter). TIMER*n* registers are read only.

Diagram



Fields

Field	Function
31-24 —	Reserved Reserved. Read value is undefined; only 0 should be written.
23-0 VALUE	Current Timer Value Holds the current timer value of the down counter. The initial value of TIMER*n* is loaded as (IVALUE -1) from INTVAL*n*, either at the end of the time interval or immediately in the following cases: INTVAL*n* is updated in the idle state or INTVAL*n* is updated with LOAD = 1. When the timer is in an idle state, reading TIMER*n*[VALUE] fields returns -1 (00FF FFFFh).

65.6.1.4 Control (CTRL0 - CTRL3)

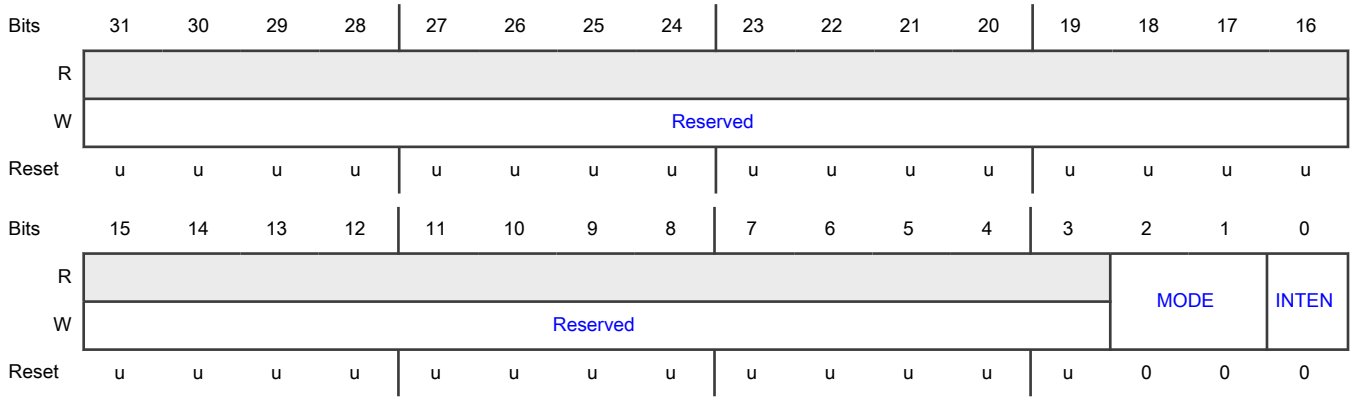
Offset

Register	Offset
CTRL0	8h
CTRL1	18h
CTRL2	28h
CTRL3	38h

Function

Controls the modes and enables the interrupt for each MRT.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-1 MODE	MRT Operating mode Selects the timer mode. 00b - Repeat Interrupt mode 01b - One-Shot Interrupt mode 10b - One-Shot Stall mode 11b - Reserved
0 INTEN	Interrupt request Enables the TIMER*n* interrupt. 0b - Disabled 1b - Enabled

65.6.1.5 Status (STAT0 - STAT3)

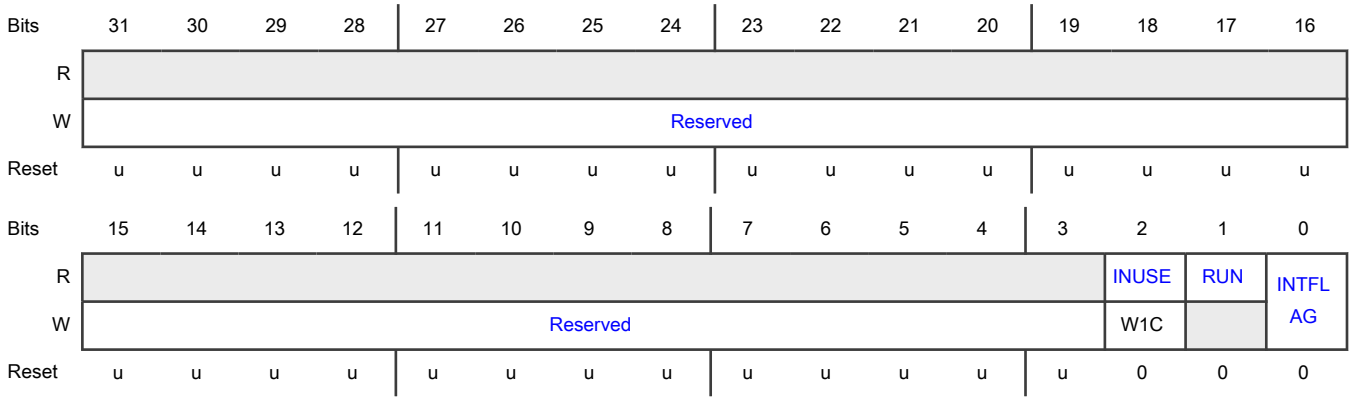
Offset

Register	Offset
STAT0	Ch
STAT1	1Ch
STAT2	2Ch
STAT3	3Ch

Function

Indicates the status of each MRT timer channel.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 INUSE	Channel-In-Use flag Affects the use of Idle Channel (IDLE_CH); the operating details depend on the operating mode bit in MODCFG[MULTITASK]. Writing 1 to this bit clears the status. 0b - This timer channel is not in use. 1b - This timer channel is in use.
1 RUN	Timer n State Indicates the state of TIMER*n*. STAT*n*[RUN] is read only. TIMER*n* stops in idle state and runs in running state. 0b - Idle state. 1b - Running.
0 INTFLAG	Interrupt Flag Monitors the interrupt flag. If the INTEN bit in the CONTROLn is also set to 1, then the interrupt for timer channel n and the global interrupt are generated. Writing a 1 to INTFLAG bit clears the interrupt request. 0b - No pending interrupt. 1b - Pending interrupt.

65.6.1.6 Module Configuration (MODCFG)

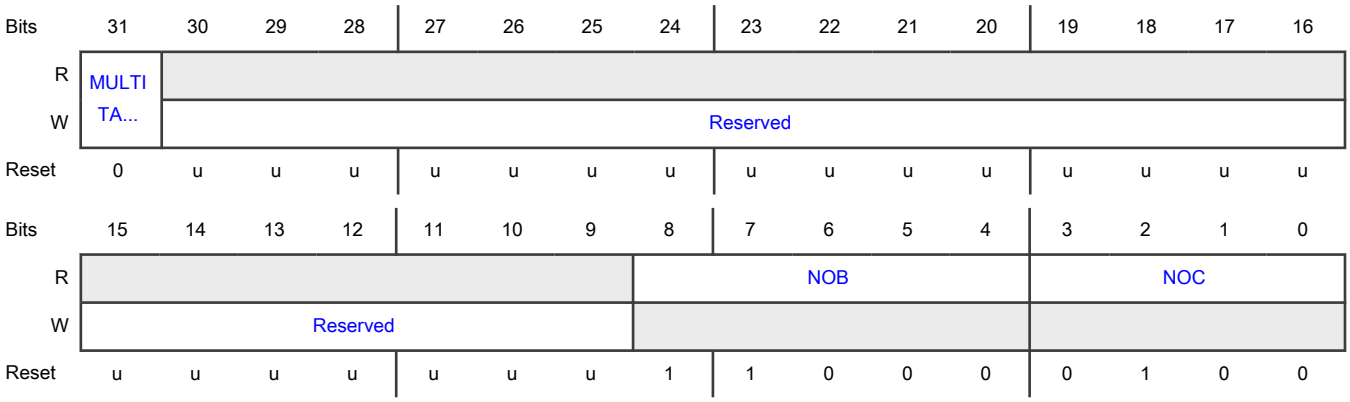
Offset

Register	Offset
MODCFG	F0h

Function

Provides information about this particular MRT instance. It also allows selecting an overall mode for the idle channel feature. MODCFG[MULTITASK] provides the configuration (number of channels and timer width) for this MRT. A reset of the INUSE(n) flags for all channels happens in Hardware Status mode.

Diagram



Fields

Field	Function
31 MULTITASK	MULTITASK Selects the operating mode for STAT*n*[INUSE] and Idle Channel (IDLE_CH). 0b - Hardware status mode. 1b - Multitask mode
30-9 —	Reserved Read value is undefined; only 0 should be written.
8-4 NOB	Number of Bits Identifies the number of timer bits in this MRT (24 bits on this chip).
3-0 NOC	Number of Channels Identifies the number of channels in this MRT.

65.6.1.7 Idle Channel (IDLE_CH)

Offset

Register	Offset
IDLE_CH	F4h

Function

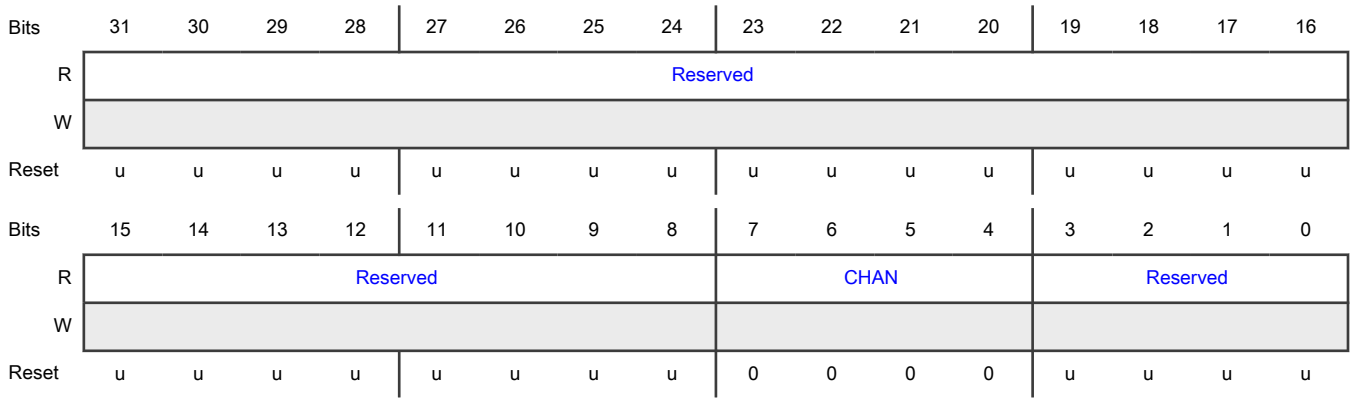
Returns the number of the first Idle Channel.

Use Idle Channel (IDLE_CH) for finding available channels in MRT. This allows you more flexibility when using MRT—no fixed channel assignments are required, and you do not need to search for an available channel. Generally, Idle Channel (IDLE_CH) returns the lowest available channel number.

MODCFG[MULTITASK] allows you to use Idle Channel (IDLE_CH) in two ways. MODCFG[MULTITASK] affects both the functions of Idle Channel (IDLE_CH), and the function of STAT*n*[INUSE] for each MRT channel:

- MULTITASK = 0: Hardware status mode. The STAT*n*[INUSE] for all MRT channels are reset. Idle Channel (IDLE_CH) returns the lowest idle channel number. A channel is considered idle if its RUN flag = 0, and there is no interrupt pending for that channel.
- MULTITASK = 1: Multitask mode. In Multitask mode, the STAT*n*[INUSE] allow more control over when MRT channels are released for further use. When Idle Channel (IDLE_CH) is read, returning a channel number of an idle channel, the STAT*n*[INUSE] for that channel is set by hardware. That channel is not considered idle until its RUN flag = 0, and there is no interrupt pending, and its INUSE flag = 0. This allows reserving an MRT channel with a single register read, and there is no need to start the channel before that channel is no longer considered idle by Idle Channel (IDLE_CH). It also allows you to identify a specific MRT channel. Then you can use that channel more than once without releasing it, removing the need to ask for an available channel for every use.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-4 CHAN	Idle Channel Helps in reading IDLE_CH[CHAN] and returns the lowest idle timer channel. The number is positioned so that you can use it as an offset from MRT base address. This provides access to the registers for the allocated channel. If all timer channels are running, then CHAN = Fh.
3-0 —	Reserved

65.6.1.8 Global Interrupt Flag (IRQ_FLAG)

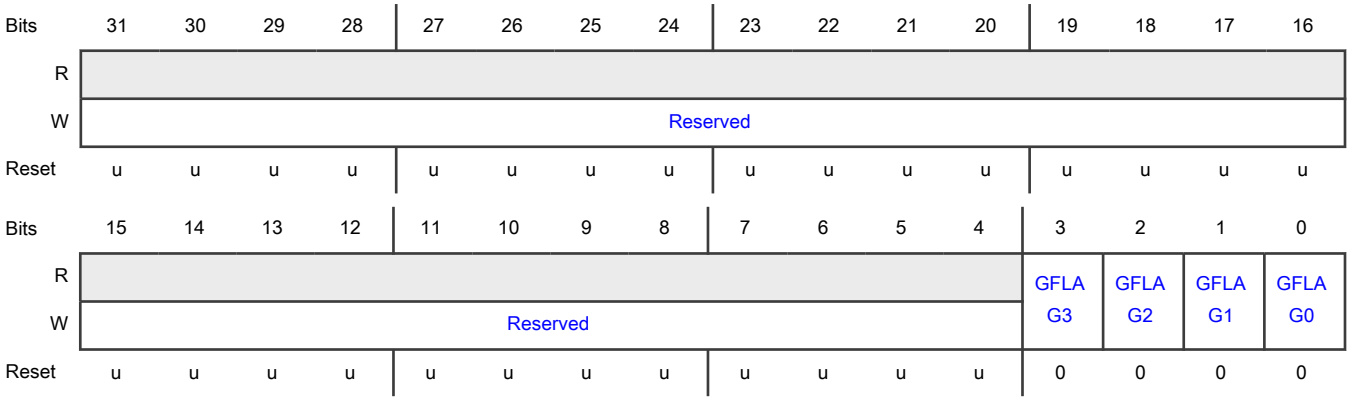
Offset

Register	Offset
IRQ_FLAG	F8h

Function

Combines the interrupt flags from the individual timer channels into one register. Setting and clearing each flag behaves in the same way as setting and clearing STAT*n*[INTFLAG] in each of the STATUS*n* registers.

Diagram



Fields

Field	Function
31-4 —	Interrupt Flag Reserved. Read value is undefined; only 0 should be written.
3 GFLAG3	Interrupt Flag Monitors the interrupt flag of TIMER3 and acts similar to channel 0.
2 GFLAG2	Interrupt Flag Monitors the interrupt flag of TIMER2 and acts similar to channel 0.
1 GFLAG1	Interrupt Flag Monitors the interrupt flag of TIMER1 and acts similar to channel 0.
0 GFLAG0	Interrupt Flag Monitors the interrupt flag of TIMER0. Writing 0 is equivalent to no operation. In the pending interrupt state, Timer (TIMER0 - TIMER3) has reached the end of the time interval. If CTRL*n*[INTEN] in CONTROL*n* also gets 1, then there is a timer channel *n* interrupt and a global interrupt. Writing 1 to IRQ_FLAG[GFLAG0] clears the interrupt request.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No pending interrupt. 1b - Pending interrupt

Chapter 66
Windowed Watchdog Timer (WWDT)

66.1 Chip-specific WWDT information

Table 610. Reference links to related information

Topic	Related module	Reference
Full description	WWDT	WWDT
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller

NOTE

A watchdog reset resets the chip. See Reset Control for details.

66.1.1 Module instances

This device has four instances of the WWDT module, numbered WWDT0 through WWDT3.

Table 611. WWDT instances

	WWDT0, WWDT1	WWDT2, WWDT3
Located in	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

66.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

66.1.3 MOD[LOCK] behavior

- When MOD[LOCK] of WWDT0 or/and WWDT1 is set to 1b, the write to the SLEEPCFG[LPOSC_PD] and RUNCFG[LPOSC_PD] fields of the SLEEPCON0 instance changes fields configuration but does not take effect.
- When MOD[LOCK] of WWDT2 or/and WWDT3 is set to 1b, the write to the SLEEPCFG[LPOSC_PD] and RUNCFG[LPOSC_PD] fields of the SLEEPCON1 instance changes fields configuration but does not take effect.

After MOD[LOCK] is set to 1b and a Watchdog feed is performed, hardware prevents disabling or powering down of the lposc, which is used as watchdog oscillator. That is, in order to keep lposc running for any WWDT instance, enable the lposc and set the MOD[LOCK] of related WWDT instances to 1b.

MOD[LOCK] can be configured once by software and is cleared by any reset.

66.1.4 Watchdog clock (WDCLK) source

For this chip, 1m_lposc is the clock source for all the WWDT instances, see Clock Controller (CLKCTL) for details.

66.2 Overview

WWDT helps you reset or interrupt a microcontroller within a programmable time, if the microcontroller (or core) is stuck in an infinite loop or is executing an unintended code. If an application fails to reload or feed a watchdog timer within the predefined duration of time, it generates a watchdog reset (if enabled).

66.2.1 Block diagram

In the following WWDT block diagram, the synchronization logic (APB bus clock to WDCLK) is not shown. See the chip-specific WWDT information section or clocking chapter for more details.

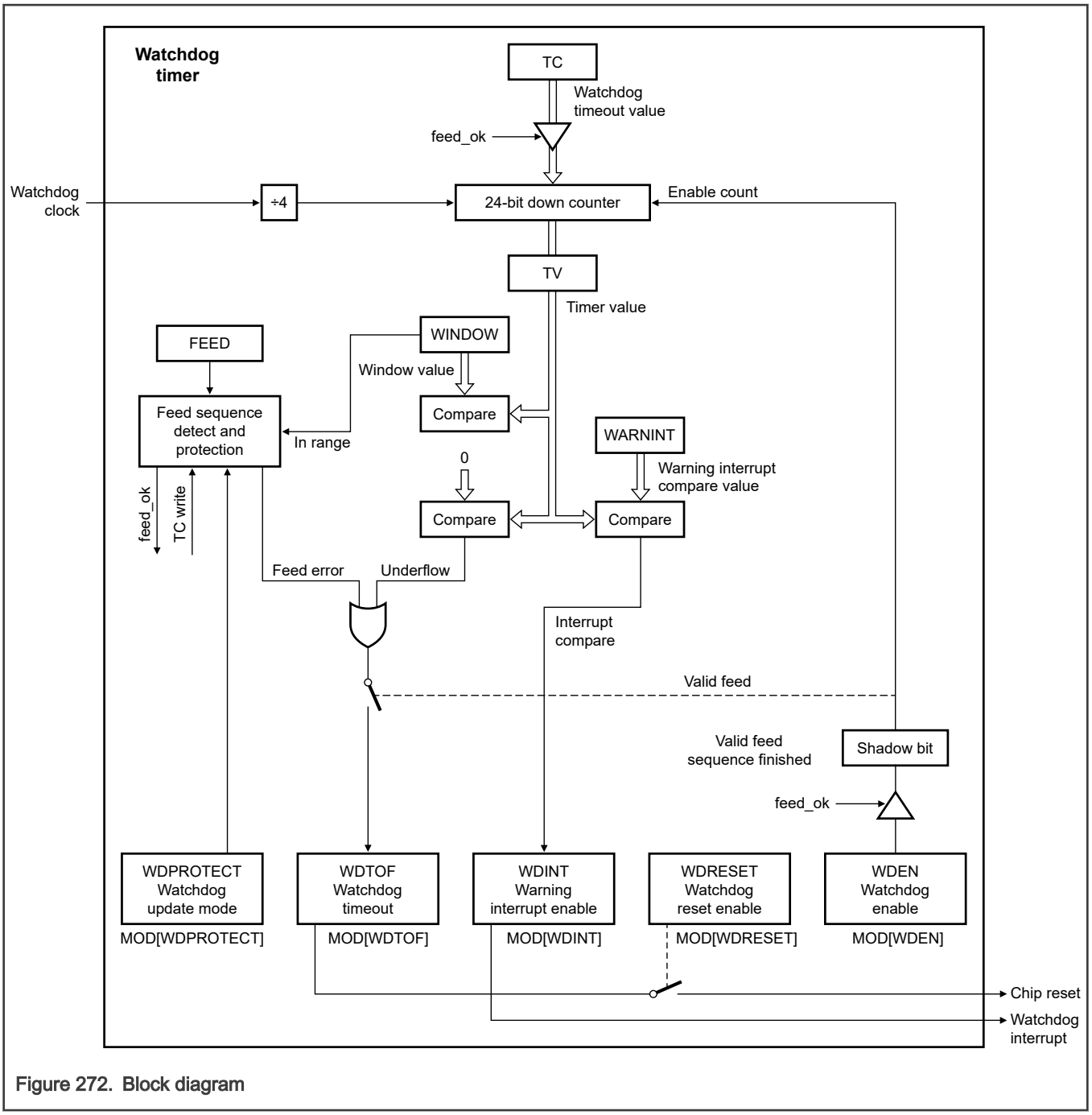


Figure 272. Block diagram

66.2.2 Features

- Can reset or interrupt the microcontroller within a programmable time
- Watchdog reset indication flag
- Clocking features:
  - Programmable 24-bit timer with an internal fixed (divided by 4) prescaler
  - Selectable time period in increments of four watchdog clocks: From 1024 watchdog clocks (T_{WDCLK} x 256 x 4) to over 67 million watchdog clocks (T_{WDCLK} x 2^{24} x 4)

  - Watchdog clock (WDCLK) source having a selectable frequency in the range of 6 kHz to 1.5 MHz (see the chip-specific information section and the Clocking chapter for detailed setting in this device)
- Can be configured to run in Low-Power mode (see the chip-specific information section and the Power Management chapter for detailed setting in this device)
- Other configurable features:
  - Windowed operation requiring a watchdog reload to occur between a programmable minimum and maximum timeout period
  - Generation of a warning interrupt at a programmable time before watchdog timeout
  - Protection of the watchdog reload value so that you can change it only after the warning interrupt time is reached

66.3 Functional description

When you program a watchdog window (a timing window), an early watchdog reload is also treated as a watchdog event, which prevents situations where a system failure may still reload the watchdog timer. For example, the application code could be stuck in an interrupt service that contains a watchdog reload (feed). You must set the timing window in a way that this situation causes an early watchdog timer reload, which generates a watchdog event, and then enables the system to recover from this situation.

WWDT consists of a fixed (divided by 4) prescaler and a 24-bit counter. The minimum value of the counter is FFh; if you write a value lower than FFh, the counter automatically loads with the value FFh.

- Minimum watchdog interval is (T_{WDCLK} X 256 X 4).
- Maximum watchdog interval is (T_{WDCLK} X 2^{24} X 4) in multiples of (T_{WDCLK} X 4).

66.3.1 Operating modes

This section describes all functional operating modes of the WWDT module: Disabled, Watchdog Interrupt, and Watchdog Reset.

Table 612. Watchdog operating modes selection

MOD[WDEN]	MOD[WDRESET]	Mode	Mode description
0	X (0 or 1)	Disabled	Debugs or operates without the watchdog running.
1	0	Watchdog Interrupt	Generates a watchdog warning interrupt but does not enable the watchdog reset. A watchdog counter equal to the value of WARNINT[WARNINT] sets MOD[WDINT] and generates a watchdog interrupt request.
1	1	Watchdog Reset	Enables both the watchdog interrupt and watchdog reset: • A watchdog counter equal to the value of WARNINT[WARNINT] sets MOD[WDINT] and generates a watchdog interrupt request. • A watchdog counter equal to 0 resets the microcontroller. • A watchdog feed before reaching the value of WINDOW[WINDOW] also causes a watchdog reset.

66.3.2 Example timing diagrams to show WWDT in operation

A feed is correct when both of the following conditions are true:

- A valid feed sequence completes, writing AAh followed by 55h to FEED[FEED].
- The value of TV[COUNT] is not greater than the value of WINDOW[WINDOW].

If either of these conditions is not true, a feed error occurs.

In the correct watchdog feed with Windowed mode enabled, the sequence of writing AAh followed by 55h occurs when the counter value (11FCh) is not greater than the value of WINDOW[WINDOW] (1200h). It is a correct feed (reload) event. After the correct feed event occurs, the watchdog counter reloads with the TC[COUNT] value (2000h).

In addition to the above two conditions, when MOD[WDPROTECT] = 1, the watchdog counter must not be greater than the value of WARNINT[WARNINT] for a correct feed to occur. See the following figure for a pictorial representation.

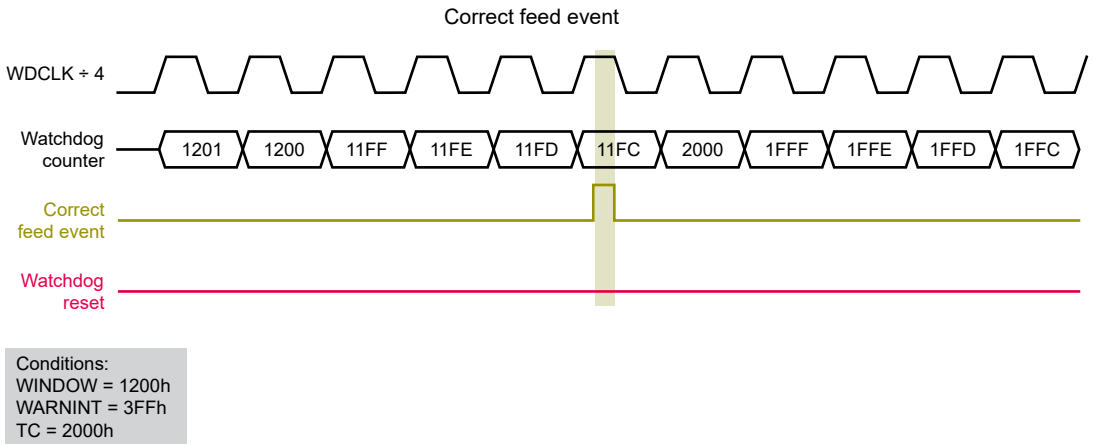


Figure 273. Correct watchdog feed with Windowed mode enabled

A correct feed sequence can only occur when the value of TV[COUNT] is not greater than the value of WINDOW[WINDOW] (1200h). Otherwise, a feed error occurs.

The following figure shows a reset trigger when you feed the watchdog too early.

An early feed event occurs if the feed sequence occurs when the watchdog counter value (1257h) is greater than the value of WINDOW[WINDOW] (1200h). This feed error can generate a reset if MOD[WDRESET] = 1.

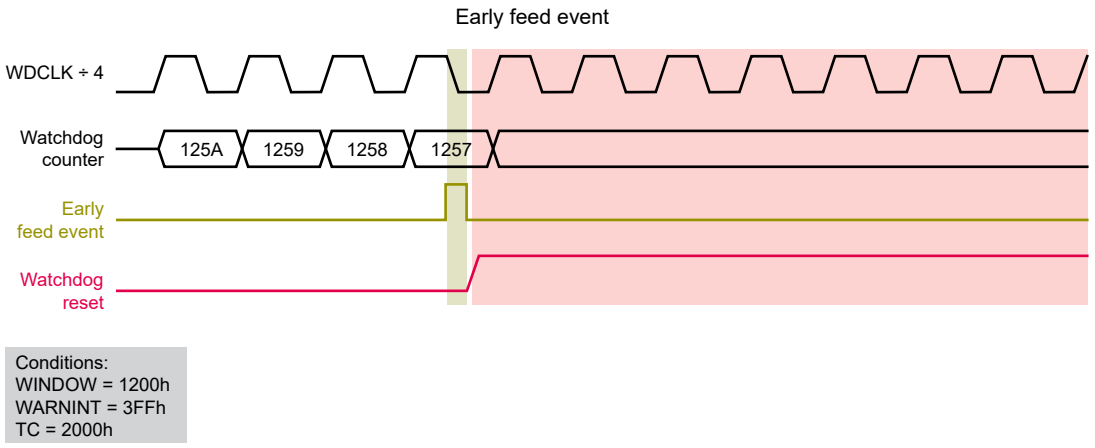


Figure 274. Early watchdog feed with Windowed mode enabled

The following figure shows WWDT generating a warning.

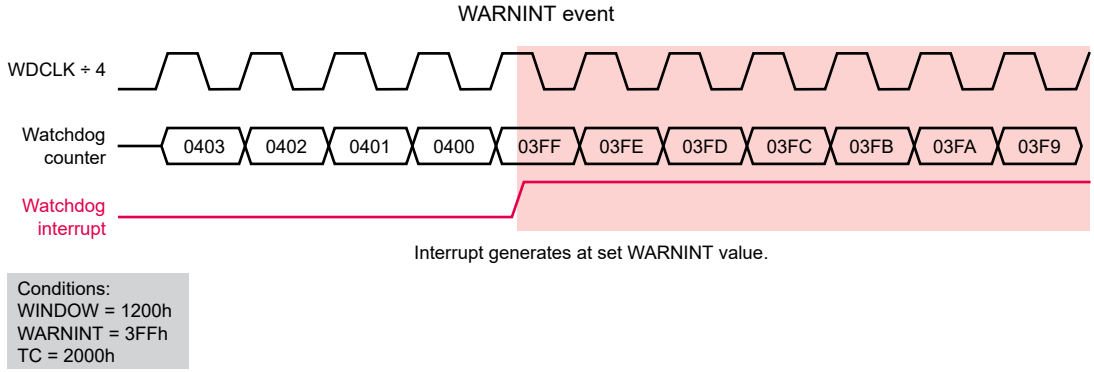


Figure 275. Watchdog warning interrupt

66.3.3 Clocking

WWDT uses two clocks: APB bus clock and WDCLK.

- The system clock derives the APB bus clock that is used for APB accesses to the watchdog registers.
- The watchdog oscillator derives WDCLK that is used for watchdog timer counting.

The synchronization logic between the two clock domains works as follows:

- When updating Mode (MOD) and Timer Constant (TC), the new value takes effect in three WDCLK cycles on the logic in the WDCLK clock domain.
- When the watchdog timer is counting on WDCLK, the synchronization logic first locks the value of the counter on WDCLK and then synchronizes it with the APB bus clock, so that the CPU can read Timer Value (TV).

NOTE

Because of the synchronization step, you must add a delay of three WDCLK clock cycles between the feed sequence and the time that you take to enable the functionality of MOD[WDPROTECT]. The length of the delay depends on the selected watchdog clock, WDCLK.

66.3.4 Using the WWDT lock

WWDT supports lock features to ensure continuous operation. These features prevent:

- The disabling of the WWDT clock source.
- The changing of the WWDT reload value.

66.3.4.1 Preventing the disabling of the WWDT clock source

If MOD[LOCK] = 1, the WWDT clock source is locked. You cannot disable the clock source when entering Sleep or Deep-Sleep modes. Therefore, enable the watchdog oscillator for each Power mode before writing 1 to MOD[LOCK].

66.3.4.2 Preventing the changing of the WWDT reload value

If MOD[WDPROTECT] = 1, then any attempt to change the value of TC[COUNT] with a feed sequence, before the watchdog counter value becomes less than the values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset.

Any type of reset disables the reload overwrite lock mechanism.

66.4 External signals

This module has no external signals.

66.5 Initialization

Perform the following steps to initialize WWDT:

1. Enable and configure the watchdog oscillator.
2. Set the watchdog timer constant reload value using TC[COUNT].
3. Set the watchdog timer operating mode using MOD[WDPROTECT].
4. Set a value for the watchdog window time using WINDOW[WINDOW], if the windowed operation is desired.
5. Set a value for the watchdog warning interrupt using WARNINT[WARNINT] and configure the system interrupt controller, if a warning interrupt is desired.
6. Enable the watchdog by writing AAh followed by 55h to FEED[FEED].

To prevent a watchdog event, you must feed or reload WWDT again before the watchdog counter reaches 0. If you set a value for the watchdog window time, then the feed or reload must also occur after the watchdog counter passes the value of WINDOW[WINDOW].

After you configure WWDT in a way that a watchdog event causes a reset (MOD[WDRESET] = 1), or when the counter reaches 0 (it causes a reset), then the microcontroller (or core) is also reset (loading the stack pointer and program counter from the vector table, just like the way it happens for an external reset).

66.6 Memory map and register definition

This section includes the WWDT module memory map and detailed descriptions of all registers.

66.6.1 WWDT register descriptions

66.6.1.1 WWDT memory map

WWDT0 base address: 4000_E000h

WWDT1 base address: 4002_E000h

WWDT2 base address: 4005_0000h

WWDT3 base address: 4005_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Mode (MOD)	32	RW	0000_0000h
4h	Timer Constant (TC)	32	RW	0000_00FFh
8h	Feed Sequence (FEED)	32	W	See section
Ch	Timer Value (TV)	32	R	0000_00FFh
14h	Warning Interrupt Compare Value (WARNINT)	32	RW	0000_0000h
18h	Window Compare Value (WINDOW)	32	RW	00FF_FFFFh

66.6.1.2 Mode (MOD)

Offset

Register	Offset
MOD	0h

Function

Controls the operation of WWDT. You must perform a watchdog feed before any changes to this register take effect.

NOTE

After you write 1 to MOD[WDEN], MOD[WDPROTECT], or MOD[WDRESET], you can return those fields to 0 only with an external reset or a WWDT reset. Until those resets occur, writing 0 to the aforementioned fields has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									Reserved	LOCK	WDPROTE...	WDINT	WDTOF	WDRESET	WDEN
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-7 —	Reserved
6 —	Reserved
5 LOCK	Lock Prevents disabling or powering down of the watchdog oscillator after you write 1 to this field and perform a watchdog feed. You can write 1 to this field, which can become 0 only with a reset. 0b - No Lock 1b - Lock
4	Watchdog Update Mode

Table continues on the next page...

Table continued from the previous page...

Field	Function
WDPROTECT	Protects the value of TC[COUNT] from changing at any time. In Flexible mode (when this field is 0), you can change the value of TC[COUNT] at any time. In Threshold mode (when this field is 1), you can change the value of TC[COUNT] only after the counter is less than the individual values of WARNINT[WARNINT] and WINDOW[WINDOW]. Any attempt to change the value of TC[COUNT] with a feed sequence, before the watchdog counter value becomes less than the values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset. You can write 1 to this field, which can become 0 only with a reset (see the note in MOD register). 0b - Flexible 1b - Threshold
3 WDINT	Warning Interrupt Flag Sets the flag when the timer is at or below the value defined in WARNINT[WARNINT]. Note that you cannot clear this flag while the value of WARNINT[WARNINT] is equal to the value of TV[COUNT]. This can occur if the value of both WARNINT[WARNINT] and MOD[WDRESET] is 0 when TV[COUNT] decrements to 0. When you configure WWDT to generate a warning interrupt, the interrupt occurs when the counter is no longer greater than the value defined by WARNINT[WARNINT]. If a feed sequence occurs when the watchdog counter value is greater than the value of WINDOW[WINDOW], this flag is set. Any reset clears this flag. 0b - No flag 1b - Flag
2 WDTOF	Watchdog Timeout Flag The bit could be set if: • Watchdog timer timeout has occurred • Feed error has occurred You must clear this flag by writing 0 to it. An external reset or POR clears this flag too. 0b - Watchdog event has not occurred. 1b - Watchdog event has occurred (causes a chip reset if WDRESET = 1).
1 WDRESET	Watchdog Reset Enable Allows a watchdog timeout to cause a chip reset, if this field = 1. After you write 1 to this field, you cannot write 0 to it (see the note in MOD register). **NOTE** If you write 0 to this field, and after an interrupt occurs, do not write 1 to this field immediately unless you refresh the watchdog. 0b - Interrupt 1b - Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 WDEN	Watchdog Enable Enables or disables the watchdog timer. After writing 1 to this field, perform a watchdog feed (reload) to enable the watchdog timer. After you write 1 to this field, you cannot write 0 to it (see the note in MOD register). 0b - Timer stopped 1b - Timer running

66.6.1.3 Timer Constant (TC)

Offset

Register	Offset
TC	4h

Function

Specifies the desired timeout value. Transferring the value of this register into the watchdog counter requires a feed sequence. The register resets to 00_00FFh. Writing a value below FFh causes 00_00FFh to load into the register. Therefore, the minimum timeout interval is T_{WDCLK} X 256 X 4.

If MOD[WDPROTECT] = 1, then any attempt to change the value of this register with a feed sequence, before the watchdog counter value becomes less than the individual values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								COUNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
23-0 COUNT	Watchdog Timeout Value Specifies the desired timeout value for WWDT.

66.6.1.4 Feed Sequence (FEED)

Offset

Register	Offset
FEED	8h

Function

Reloads the watchdog timer with the TC[COUNT] value, if you write AAh followed by 55h to FEED[FEED]. This operation also starts WWDT, if the watchdog timer is enabled by writing 1 to MOD[WDEN].

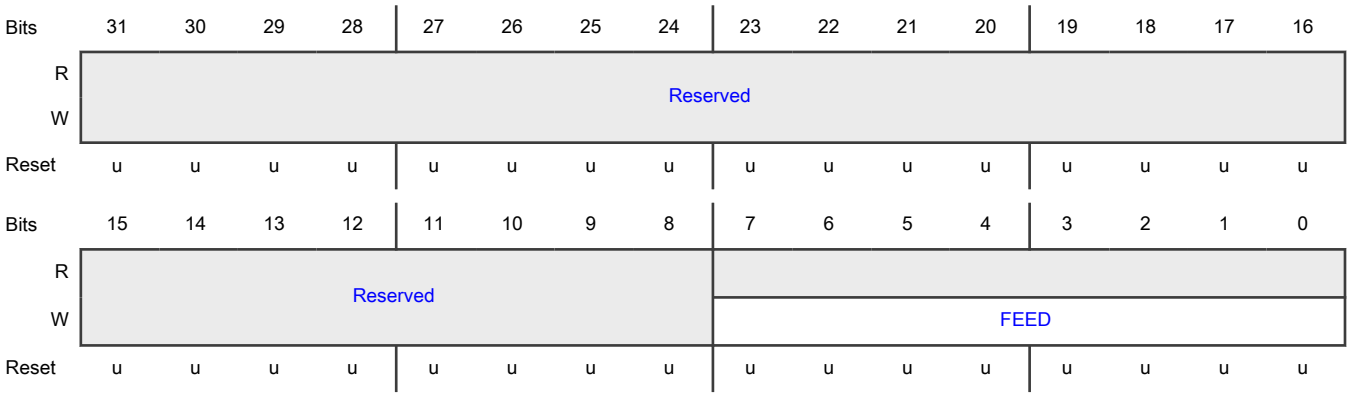
However, writing 1 to MOD[WDEN] is not sufficient to enable WWDT. Before the watchdog can generate a reset, MOD[WDEN] must be 1, followed by a valid feed sequence that completes itself. Until then, the watchdog ignores feed errors.

After writing AAh to FEED[FEED], access to any watchdog register other than writing 55h to FEED[FEED] causes an immediate reset or interrupt when WWDT is enabled and sets MOD[WDTOF]. An incorrect access to a watchdog register during a feed sequence generates the reset during the second APB bus clock.

To avoid an unintended interrupt, it is a good practice to disable interrupts around a feed sequence. Disable interrupts if they may result in rescheduling processor control away from the current task in the middle of the feed, and then lead to some other access to WWDT before control returns to the interrupted task.

If the value of WINDOW[WINDOW] is less than the default value, it may limit the time for which a watchdog feed is allowed.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 FEED	Feed Value Specifies feed value. The value must be AAh followed by 55h.

66.6.1.5 Timer Value (TV)

Offset

Register	Offset
TV	Ch

Function

Contains the current value of the WWDT counter. When the core (or CPU) reads this register, the value of this register is older than the actual value of the timer. Reading the timer using the lock and synchronization procedure takes up to six WDCLK cycles plus six APB bus clock cycles.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								COUNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Counter Timer Value Specifies the current value of the watchdog timer.

66.6.1.6 Warning Interrupt Compare Value (WARNINT)

Offset

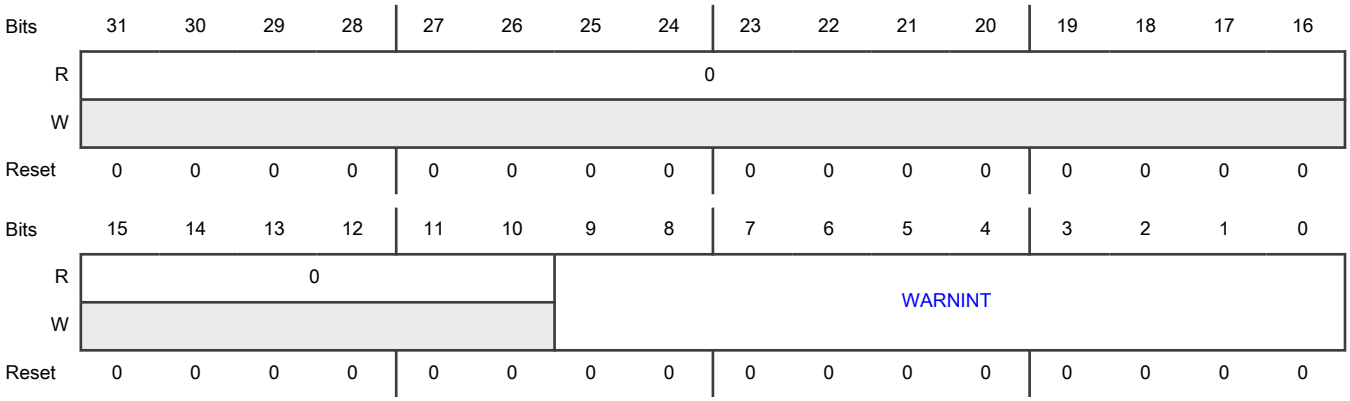
Register	Offset
WARNINT	14h

Function

Determines the value of TV[COUNT] that generates a watchdog interrupt. When this value is less than or equal to the value defined by WARNINT, an interrupt generates after the subsequent WDCLK.

A match of the WWDT counter to WARNINT occurs when the lower 10 bits of the timer counter have the same value as the lower 10 bits of WARNINT, and the remaining upper bits of the counter are all 0. This gives a maximum time of 1023 watchdog timer counts (4096 watchdog clocks) for the interrupt to occur before a watchdog event. If WARNINT is 0, then the interrupt occurs at the same time as the watchdog event.

Diagram



Fields

Field	Function
31-10 —	Reserved
9-0 WARNINT	Watchdog Warning Interrupt Compare Value Specifies a timer value that generates a warning interrupt.

66.6.1.7 Window Compare Value (WINDOW)

Offset

Register	Offset
WINDOW	18h

Function

Determines the highest timer value (TV[COUNT]) allowed during a watchdog feed. If a feed sequence occurs when the value of TV[COUNT] is greater than the value in WINDOW[WINDOW], a watchdog event occurs.

WINDOW[WINDOW] resets to the maximum value of TV[COUNT], where windowing is not in effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								WINDOW							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WINDOW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24 —	Reserved
23-0 WINDOW	Watchdog Window Value Specifies the highest timer value in which a watchdog feed can occur.

Chapter 67
Flexible I/O (FLEXIO)

67.1 Chip-specific FLEXIO information

Table 613. Reference links to related information

Topic	Related module	Reference
Full description	FLEXIO	FLEXIO
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Input multiplexing	INPUTMUX	See FLEXIO_TRIG0 registers in INPUTMUX

67.1.1 Module instances

This device has one instance of the FLEXIO module.

Table 614. FLEXIO instance

	FLEXIO
Located in	VDD2_MEDIA
Access	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3

NOTE

Both CPU0 and CPU1 can operate FLEXIO. You need to use CPU0 to configure FLEXIO ports by IOPCTL0 registers before either CPU to operate the FLEXIO.

67.1.2 FLEXIO external trigger connections

The FLEXIO external trigger inputs come from various sources within the chip and are selected by the FLEXIO peripheral input muxes residing at INPUTMUX_BASE address plus offsets 0x760h – 0x76Ch. See FLEXIO_TRIG registers in the INPUTMUX.

67.2 Overview

FLEXIO is a highly configurable module that provides:

- Emulation of various serial or parallel communication protocols.
- Flexible 16-bit timers with support for various trigger, reset, enable, and disable conditions.

- Programmable logic blocks that allow the implementation of digital logic functions on-chip and configurable interaction of internal and external modules.
- Programmable state machine for offloading basic system control functions from the CPU.

67.2.1 Block diagram

The following diagram provides a high-level overview of the FLEXIO timer and shifter configuration.

FLEXIO uses shifters, timers, and external triggers to shift data into or out of FLEXIO. As shown in the block diagram, timers control the timing of this data shift. You can configure the timers to use generic timer functions, external triggers, or various other conditions to determine the control.

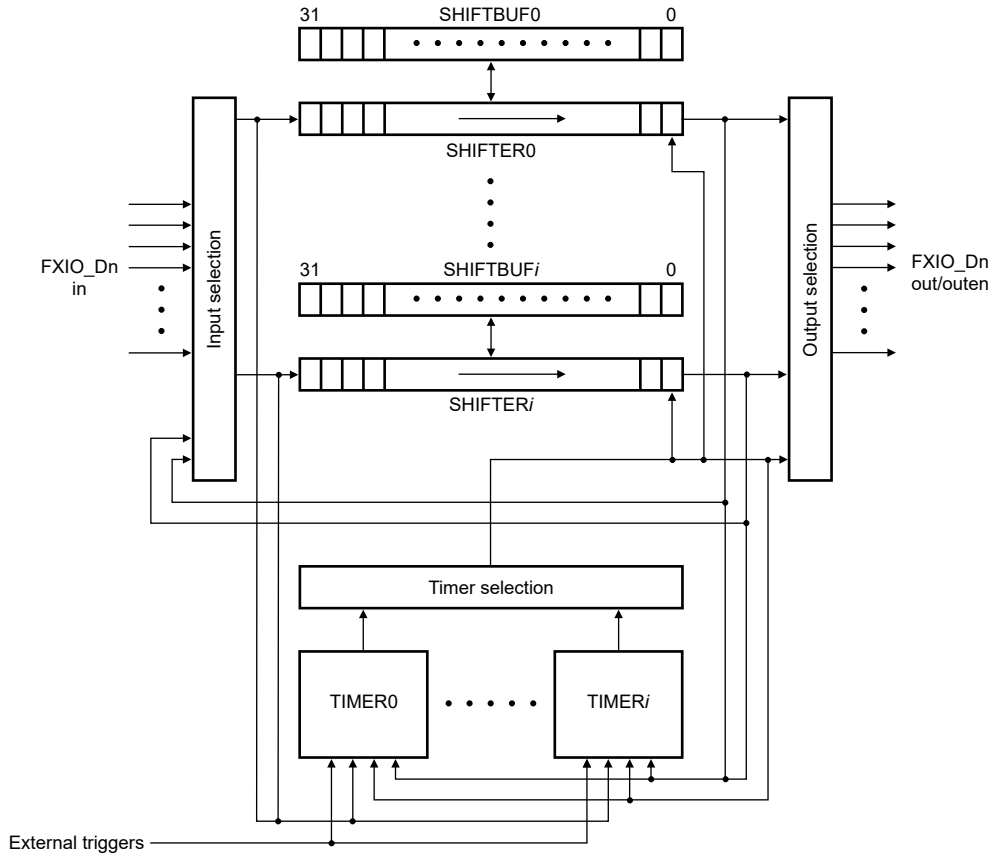


Figure 276. Block diagram

67.2.2 Features

- Array of 32-bit shift registers with transmit, receive, data match, logic, and state modes:
  - Double-buffered shifter operation for continuous data transfer
  - Shifter concatenation to support large transfer sizes
  - Automatic start and stop bit generation
  - 1, 2, 4, 8, 16, or 32 multi-bit shift widths for parallel interface support
  - Interrupt, DMA, or polled transmit and receive operation
- Highly flexible 16-bit timers with support for various internal or external triggers, reset, enable, and disable conditions:

— Programmable baud rates independent of bus clock frequency, with support for asynchronous operation during Stop mode

— Programmable logic mode for integrating external digital logic functions on-chip, or combining pin, shifter, or timer functions to generate complex outputs

— Programmable state machine for offloading basic system control functions from CPU, with support for up to eight states, eight outputs, and three selectable inputs per state

- Integrated general-purpose I/O registers and pin rising or falling edge interrupts to simplify software support
- Support for a wide range of protocols, including but not limited to:
  - — UART
  - — I2C
  - — SPI
  - — I2S
  - — Camera IF
  - — Motorola 68K or Intel 8080 bus
  - — PWM or waveform generation
  - — Input-capture (pulse-edge interval measurement), such as SENT

67.3 Functional description

67.3.1 Shifter operation

Shifters are responsible for buffering and shifting data into or out of FLEXIO. The timer assigned to the shifter controls the timing of shift, load, and store events via SHIFTCTL*n*[TIMSEL]. Shifters are designed to support either DMA, interrupt, or polled operations. The following figure provides a detailed view of the shifter microarchitecture.

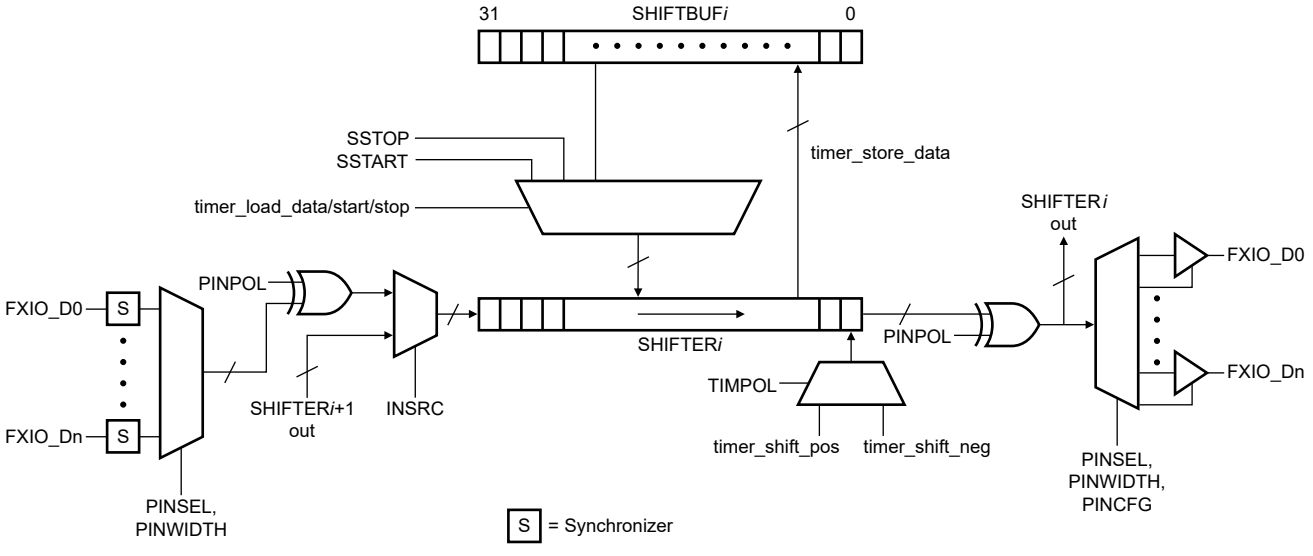


Figure 277. Shifter microarchitecture

67.3.1.1 Transmit mode

In Transmit mode (SHIFTCTL*n*[SMOD] = 010b), the shifter loads data from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) and shifts data out when the assigned timer signals a load event. An optional start and stop bit can be automatically loaded before or

after SHIFTBUF register data by configuring either SHIFTCFG[SSTART] and TIMCFG[TSTART], or SHIFTCFG[SSTOP] and TIMCFG[TSTOP] in the shifter and timer.

NOTE

If a stop bit is enabled, the shifter immediately loads a stop bit when it is initially configured for Transmit mode.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data has either been loaded from the SHIFTBUF register into the shifter or when the shifter is initially configured for Transmit mode. To clear the flag, write 1 or write new data to SHIFTBUF. In Transmit mode, write any value to the SHIFTBUF register to clear the corresponding shifter status flag, which is cleared regardless of what is writing to the register (DMA or interrupt), or the state of the DMA or interrupt enables. See the functional description of SHIFTSTAT[SSF] for information on how the flag is set and cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to load data from an empty SHIFTBUF register occurs (buffer underrun). Clear the flag by writing 1.

67.3.1.2 Receive mode

When the assigned timer signals a store event in Receive mode (SHIFTCTL*n*[SMOD] = 001b), the shifter shifts and stores data in Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). You can check for a start and stop bit before or after the shifter data is sampled by configuring either SHIFTCFG[SSTART] and TIMCFG[TSTART], or SHIFTCFG[SSTOP] and TIMCFG[TSTOP] in the shifter and timer.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data is stored in the SHIFTBUF register from the shifter. To clear the flag, write 1 to or read the data from SHIFTBUF. Any read of the SHIFTBUF register clears the corresponding shifter status flag when the shifter is in Receive mode. The flag is cleared regardless of what is reading the register (DMA or interrupt) or the state of the DMA or interrupt enables. See the functional description of SHIFTSTAT[SSF] for information on how the flag is set or cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set either when an attempt to store data into a full SHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start or stop bit check. Write 1 to clear the flag.

67.3.1.3 Match Store mode

In Match Store mode (SHIFTCTL*n*[SMOD] = 100b), the shifter shifts data in, checks for a match result, and stores matched data in Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) when the assigned timer signals a store event. By configuring either SHIFTCFG[SSTART], TIMCFG[TSTART], and SHIFTCFG[SSTOP], or TIMCFG[TSTOP] in the shifter and timer, you can check for a start and stop bit before or after the shifter data is sampled. You can compare up to 16 bits of data using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs and the matched data is stored in the SHIFTBUF register from the shifter. To clear the flag, read the matched data from the SHIFTBUF register or write 1 to the flag. Any read of the SHIFTBUF register clears the corresponding shifter status flag when the shifter is configured in Match Store mode. The flag is cleared regardless of what is reading the register (DMA or interrupt) or the state of the DMA or interrupt enables. See the functional description for SHIFTSTAT[SSF] to know how the flag is set or cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to store matched data into a full SHIFTBUF register occurs (buffer overrun), or when a mismatch occurs on a start or stop bit check. Write 1 to clear the flag.

67.3.1.4 Match Continuous mode

In Match Continuous mode (SHIFTCTL*n*[SMOD] = 101b), the shifter shifts data in and continuously checks for a match result whenever a shift event is signaled by the assigned timer. You can compare up to 16 bits of data using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs. The flag clears automatically as soon as no match exists between the shifter data and Shifter Buffer (SHIFTBUF0 - SHIFTBUF7).

You cannot clear the flag by reading the SHIFTBUF register.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when a match occurs. To clear the flag, write 1 or perform a read from the SHIFTBUF register.

67.3.1.5 State mode

State mode enables you to implement any state machine with up to eight states, eight outputs, and three selectable inputs per state. This feature allows basic control functions to be offloaded from the CPU, which can remain in a low-power mode.

In State mode (SHIFTCTL*n*[SMOD] = 110b), when the shifter is selected by the current state pointer (SHIFTSTATE[STATE]), use the SHIFTBUF register to drive the output and compute the next state values. The following figure provides a detailed view of the shifter microarchitecture when configured for State mode.

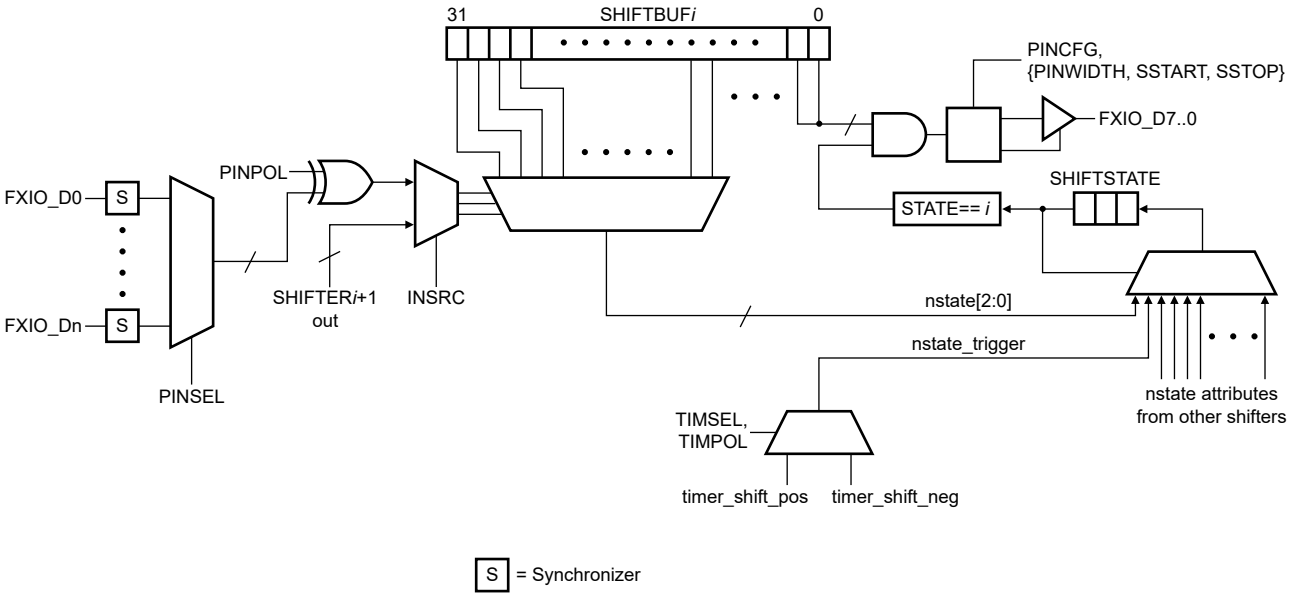


Figure 278. State microarchitecture

When the current state pointer selects a specific shifter (shifter *n*), output pins FXIO_D[7:0] are driven by SHIFTBUF*n*[31:24]; the configuration is defined by SHIFTCTL*n*[PINCFG]. Write 1 to Shifter Configuration (SHIFTCFG0 - SHIFTCFG7) {PWIDTH[3:0],SSTOP[1:0],SSTART[1:0]} to disable the output drive on pins FXIO_D[7:0] for state machine applications that require less than eight output pins.

Use the three input pins selected by SHIFTCTL*n*[PINSEL] and SHIFTBUF*n*[23:0] to compute the next state value.

NOTE

Each state can use a different set of three input pins.

The following table shows how the next state value is computed when the current state pointer is pointing to shifter *n*.

Table 615. Next state computation for SHIFTSTATE[STATE] = *n*

FXIO_D[PINSEL + 2]	FXIO_D[PINSEL + 1]	FXIO_D[PINSEL]	Next state value
0	0	0	SHIFTBUF*n*[2:0]
0	0	1	SHIFTBUF*n*[5:3]
0	1	0	SHIFTBUF*n*[8:6]
0	1	1	SHIFTBUF*n*[11:9]

Table continues on the next page...

Table 615. Next state computation for SHIFTSTATE[STATE] = *n* (continued)

FXIO_D[PINSEL + 2]	FXIO_D[PINSEL + 1]	FXIO_D[PINSEL]	Next state value
...	...	...	...
1	1	1	SHIFTBUF*n*[23:21]

NOTE

You can configure other shifters and timers to drive the input pins of a given state, allowing you to create complex combinations of shifters and timers as needed. For example, the output of a shifter configured for Logic mode can be used to drive a state machine input.

The next state transition is triggered using the timer output selected by SHIFTCTL*n*[TIMSEL], with polarity controlled by SHIFTCTL*n*[TIMPOL].

NOTE

Each state can use a different timer to trigger each next state transition, allowing various internal or external trigger sources and clocking configurations to be used. See Timer section for more information.

The current state pointer defaults to shifter 0 at reset; however, you can write to select a different shifter for the initial state. If the current state pointer selects a shifter that is not configured for State mode, then outputs are not driven and the next state transition is never triggered.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when the shifter is selected by the current state pointer. The flag is cleared when the current state pointer is updated to a different shifter.

67.3.1.6 Logic mode

Logic mode enables you to implement a small amount of programmable digital logic within a FLEXIO shifter.

In Logic mode (SHIFTCTL*n*[SMOD] = 111b), use Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) to implement a 5-input, 32-bit programmable logic lookup table. The following figure provides a detailed view of shifter microarchitecture when configured for Logic mode.

Use the SHIFTBUF register to configure the lookup table for the four pin inputs. You can also use SHIFTER*n* to configure a feedback or delayed pin source as the fifth input to the lookup table.

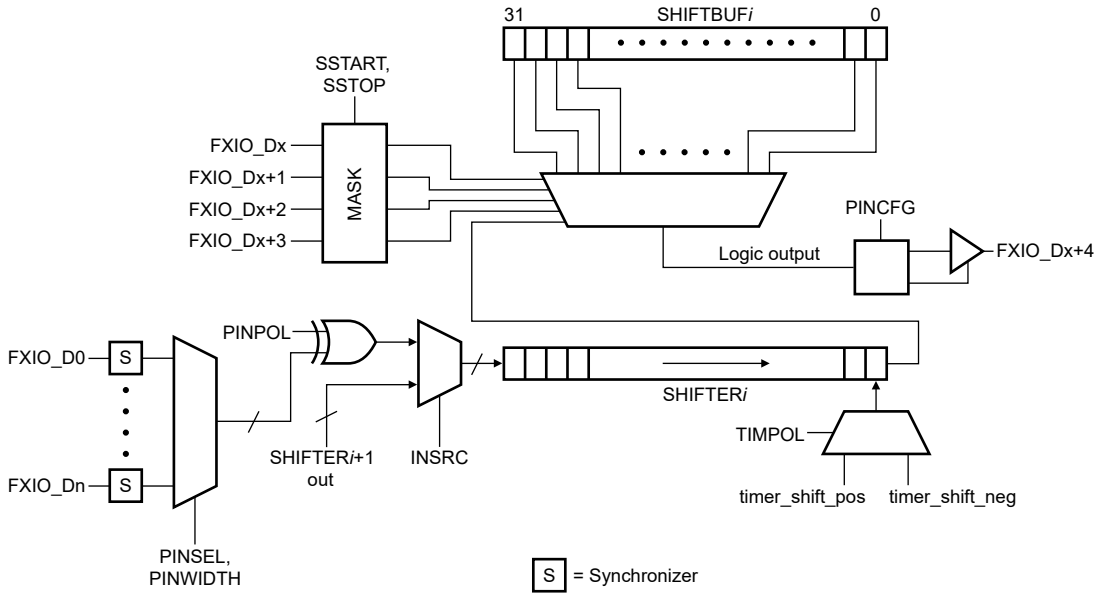


Figure 279. Logic microarchitecture

The lookup table is driven using four pin inputs (maskable using SHIFTCFG[SSTOP] and SHIFTCFG[SSTART]), plus one input from the internal shifter. It can be configured to drive an output pin using SHIFTCTL[PINCFG]. Pin inputs and outputs are fixed for each logic lookup table and are not selectable. The following table lists the logic output value selected by the lookup table for shifter *n*.

Table 616. Logic lookup table for shifter *n*

SHIFTER*n*[0]	FXIO_D[*x* + 3][1]	FXIO_D[*x* + 2]	FXIO_D[*x* + 1]	FXIO_D[*x*]	Logic output to FXIO_D[*x* + 4]
0	0	0	0	0	SHIFTBUF*n*[0]
0	0	0	0	1	SHIFTBUF*n*[1]
0	0	0	1	0	SHIFTBUF*n*[2]
0	0	0	1	1	SHIFTBUF*n*[3]
...	...	...	...	...	...
1	1	1	1	1	SHIFTBUF*n*[31]

1. for shifters *n* = 0...3, *x* = *n*

   for shifter *n* = 4...7, *x* = *n* + 4

To minimize output glitches, use SHIFTCFG*n*[SSTOP] and SHIFTCFG*n*[SSTART] to mask unused input pins. When these fields are 1, {SSTOP[1:0] and SSTART[1:0]} mask FXIO_D[*x* + 3]...FXIO_D[*x*] inputs respectively so that any transitions on these pins do not cause the logic output to glitch.

NOTE

You can configure other shifters and timers to drive the input pins of a given lookup table, allowing you to concatenate lookup tables or create complex combinations of shifters and timers as needed.

SHIFTCFG[PWIDTH] controls the number of delay stages introduced by the internal shifter input (SHIFTER*n*[0]). For example, when configured for a 1-bit shift (SHIFTCFG[PWIDTH] = 0), the internal shifter introduces a 32-shift clock delay before passing its

input (selected by SHIFTCTL[PINSEL]) to the lookup table. When configured for a 32-bit shift (SHIFTCFG[PWIDTH] = 16...31), the internal shifter introduces a 1-shift clock delay to its input.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set whenever the output pin allocated to the logic lookup table has a value of 1 after being synchronized with the FLEXIO clock. The flag clears when the output pin has a value of 0. This also allows SHIFTSTAT[SSF] to be used as a trigger to a timer if needed.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when the output pin allocated to the logic lookup table is asserted. Clear the flag by writing 1 to it.

The Logic mode input pins, including pins driven by other shifters and timers, are synchronized with the FLEXIO functional clock before they are input to the programmable logic lookup table.

67.3.2 Timer operation

The FLEXIO 16-bit timers control the loading, shifting, and storing of the shift registers. The counters load the contents of the compare register and decrement down to zero on the FLEXIO clock. The counters can perform generic timer functions such as generating a clock, select output, or a PWM waveform. You can configure these timers to perform any of the following functions:

- Enable in response to a trigger, pin, or shifter condition.
- Decrement always or only on a trigger or pin edge.
- Reset in response to a trigger or pin condition.
- Disable on a trigger or pin condition or on a timer compare.

Timers can optionally include a start condition and a stop condition.

Although each timer operates independently, you can configure a timer to enable or disable at the same time as the previous timer (for example, timer 1 can enable or disable at the same time as timer 0) and a timer output can be used to trigger any other timer. The trigger used by each timer is configured independently as a timer output, shifter status flag, pin input, or an external trigger input. The trigger configuration is separate from pin configuration; you can perform it to configure input, output data, or output enable. See the chip-specific FLEXIO information for information on external trigger connections.

You must configure Timer Configuration (TIMCFG0 - TIMCFG7) before writing 1 to TIMCTL*n*[TIMOD].

67.3.2.1 Timer 8-bit Baud Counter mode

In 8-bit Baud Counter mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the baud rate of the shift clock and the upper 8 bits are used to configure the number of shift clock edges in the transfer. When the lower 8 bits decrement to zero, the timer output is toggled and the lower 8 bits reload from the compare register. The upper 8 bits decrement when the lower 8 bits become zero and decrement.

NOTE

A timer reset event in 8-bit Baud Counter mode only resets the lower 8-bit counter. The upper 8-bit counter is not affected and can decrement if the timer reset is configured to update the state of the timer output, which toggles as a result of the timer reset event.

A timer compare event occurs when the upper 8 bits equal zero and decrement. The timer status flag is set on a timer compare event.

67.3.2.2 Timer 8-bit High PWM mode

In 8-bit High PWM mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the timer output high period and the upper 8 bits are used to configure the timer output low period. The lower 8 bits decrement when the output is high. When the lower 8 bits become zero and decrement, the timer output is cleared and the lower 8 bits are reloaded from the compare register. The upper 8 bits decrement when the output is low. When the upper 8 bits become zero and decrement, the timer output is set and the upper 8 bits are reloaded from the compare register.

A timer compare event occurs when the upper 8 bits become zero and decrement. The timer status flag is set on a timer compare event.

67.3.2.3 Timer 16-bit Counter mode

In 16-bit Counter mode, you can use the 16-bit counter to configure either the baud rate of the shift clock (for example, TIMDEC[1:0] ≠ 10 or 11) or the number of shift clock edges in the transfer (for example, TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer compare event.

67.3.2.4 Timer 16-bit Counter Disable mode

In 16-bit Counter Disable mode, the 16-bit counter can be used to configure either the baud rate of the shift clock (for example, TIMDEC[1:0] ≠ 10 or 11) or the number of shift clock edges in the transfer (for example, TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer disable event.

67.3.2.5 Timer 8-bit Word Counter mode

In 8-bit Word Counter mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the number of shift clock edges in each word and the upper 8 bits are used to configure the number of words in the transfer. When the lower 8 bits decrement to zero, the timer output is toggled and the lower 8 bits reload from the compare register. The upper 8 bits only decrement when the lower 8 bits become zero and decrement.

A timer compare event occurs when the lower 8 bits become zero and decrement. The timer status flag is set when the upper 8 bits become zero and decrement.

67.3.2.6 Timer 8-bit Low PWM mode

In 8-bit Low PWM mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the timer output low period and the upper 8 bits are used to configure the timer output high period. The lower 8 bits decrement when the output is low. When the lower 8 bits become zero and decrement, the timer output is set and the lower 8 bits are reloaded from the compare register. The upper 8 bits decrement when the output is high. When the upper 8 bits become zero and decrement, the timer output is cleared and the upper 8 bits are reloaded from the compare register.

A timer compare event occurs when the upper 8 bits become zero and decrement. The timer status flag is set on a timer compare event.

67.3.2.7 Timer enable and start functions

The following events occur when you configure TIMCTL*n*[TIMOD] for the desired mode and the condition configured by the timer enable (TIMCFG*n*[TIMENA]) is detected. When TIMCTL*n*[ONETIM] is 1, the timer status flag must be clear to generate a timer enable event; otherwise, the timer enable event is blocked. You can use this to enforce software intervention after each timer iteration:

- The timer counter loads the current value of the compare register and starts decrementing, as configured by TIMCFG*n*[TIMDEC].
- The timer output may update to its initial state depending on the configuration of TIMCFG*n*[TIMOUT]. Shifters that are controlled by this timer do not see this as a rising edge on the timer shift clock.
- Transmit shifters controlled by this timer either output their start bit value or load the shift register from the shift buffer and output the first bit, as configured by SHIFTCFG*n*[SSTART].

If the timer start bit is enabled, the timer counter reloads with the compare register on the first rising edge of the shift clock after the timer starts decrementing. If there is no falling edge on the shift clock before the first rising edge (for example, when TIMCFG*n*[TIMOUT] = 1), a shifter that is configured to shift on the falling edge and load on the first shift does not load correctly.

67.3.2.8 Timer decrement and reset functions

The timer generates the timer output and timer shift clock depending on the fields, TIMCTL*n*[TIMOD] and TIMCFG*n*[TIMDEC]. The shifter clock is either equal to the timer output (when TIMCFG*n*[TIMDEC] ≠ 10 or 11) or equal to the decrement clock (when TIMCFG*n*[TIMDEC] = 10 or 11). If you configure TIMCFG*n*[TIMDEC] to decrement from a pin or trigger, the timer decrements on both rising and falling edges.

If a timer is configured to decrement on the FLEXIO functional clock divided by 16 or 256 (when TIMCFG*n*[TIMDEC] = 100 or 101), then a common prescaler that is shared by all timers is used to generate the two divide ratios. This prescaler is reset when all timers are either idle or configured not to use the prescaler (TIMCFG*n*[TIMDEC] ≠ 100 or 101).

If you configure the timer to reset as determined by TIMCFG*n*[TIMRST], then the timer counter loads the current value of the compare register again. You can configure the timer output and timer shift clock to update on timer reset, as configured by TIMCFG*n*[TIMOUT]. If the time output toggles as a result of the timer reset, this can result in a timer shift clock edge. In 8-bit Baud Counter mode, this also decrements the upper 8 bits of the counter.

In general, when the timer counter decrements to zero, a timer compare event is triggered. The timer compare event causes:

- The timer counter to load the contents of the timer compare register.
- The timer output to toggle.
- Any configured transmit shift registers to load.
- Any configured receive shift registers to store.

Depending on the timer mode, the timer status flag may also be set.

67.3.2.9 Timer disable and stop functions

When the timer is configured to add a stop bit on each compare, the following additional events occur:

- Transmit shifters controlled by this timer output their stop bit value (if configured by SHIFTCFG*n*[SSTOP]).
- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SHIFTCFG*n*[SSTOP].
- The timer counter reloads the current value of the compare register on the first rising edge of the shifter clock after the compare.

If you configure the timer to insert a stop bit on each compare, you must configure the transmit shifters to load on the first shift.

When the condition configured by timer disable (TIMCFG*n*[TIMDIS]) is detected, the following events occur:

- Timer counter reloads the current value of the compare register and starts decrementing as configured by TIMCFG*n*[TIMDEC].
- Timer output clears. Shifters that are controlled by this timer do not see this as a falling edge on the timer shift clock, but can generate a shift event if the timer shift clock otherwise generates one.
- Transmit shifters controlled by this timer output their stop bit value (if configured by SHIFTCFG*n*[SSTOP]).
- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SHIFTCFG*n*[SSTOP].

If the timer stop bit is enabled, the timer counter continues decrementing until the next rising edge of the shift clock is detected, at which point it finishes decrementing. Although the timer output is forced low during the stop bit, the timer shift clock can toggle during the stop bit. The timer output does not generate shift events during the stop bit.

A timer enable condition can be detected in the same cycle as a timer disable condition (if timer stop bit is disabled), or on the first rising edge of the shift clock after the disable condition (if stop bit is enabled). When TIMCTL*n*[ONETIM] is 1, the timer status flag must be clear before the next timer enable condition is detected. When the timer is in the stop state condition, receive shift registers with stop bit enabled store the contents of the shift register into the shift buffer and verify the state of the input data on the configured shift edge. If there is no configured edge between the timer disable and the next rising edge of the shift clock, then the final store and verify do not occur.

67.3.3 Pin operation

The pin configuration for each timer and shifter can be set to use any FLEXIO pin with either polarity. You can configure each timer and shifter as an input, output data, output enable, or bidirectional output. A pin configured for output enable can be used as an open drain (with inverted polarity because the output enable assertion causes logic zero to be output on the pin) or to control the enable on the bidirectional output. You can configure any timer or shifter to control the output enable for a pin where the bidirectional output data is driven by another timer or shifter.

When more than one shifter or timer is configured to use the same pin for output data, then the output data to the pin is combined using an OR function. This also applies if more than one timer or shifter is configured to use the same pin for output enable.

67.3.3.1 Parallel interface

You can configure shifters to use multiple FLEXIO pins simultaneously by using SHIFTCFG*n*[PWIDTH], which is used to configure the following settings of a shifter:

1. Number of bits shifted per shift clock.
2. Number of pins driven by the shifter per shift clock (only on shifters supporting parallel transmit—that is, SHIFTER0 and SHIFTER4.)
3. Number of pins sampled by the shifter per shift clock (only on shifters supporting parallel receive—that is, SHIFTER3 and SHIFTER7.)

When configured for parallel shift, either 4, 8, 16, or 32 bits can be shifted on every shift clock. If an adjacent shifter is selected as the input source (SHIFTCFG*n*[INSRC] = 1), the least significant 4, 8, 16, or 32 bits from the adjacent shifter are sampled on each shift clock.

For shifters supporting parallel receive (SHIFTER3, SHIFTER7), you can configure the shifter to sample multiple pins (INSRC = 0), with PWIDTH and PINSEL selecting the pins as FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL].

NOTE

If PWIDTH is less than the number of bits being shifted on each shift clock, then the most significant bits are masked with 0. For example, if PINSEL = 7 and PWIDTH = 6, then SHIFTER[31:24] samples {0,0,FXIO_D[12:7]} on each shift clock.

For shifters supporting parallel transmit (SHIFTER0, SHIFTER4), you can configure the shifter to drive multiple pins using SHIFTCTL*n*[PINCFG], with PWIDTH and PINSEL selecting the pins as follows: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL].

NOTE

If PWIDTH is less than the number of bits being shifted on each shift clock, then the most significant pins are not driven. For example, if PINSEL = 7 and PWIDTH = 6, then SHIFTER[5:0] drives only FXIO_D[12:7] on each shift clock.

67.3.3.2 Pin synchronization

When you configure a pin as an input (this includes a timer trigger configured as a pin input), the input signal is first synchronized with the FLEXIO clock before a timer or shifter could use the signal. This introduces a small latency of 0.5–1.5 FLEXIO clock cycles when using an external pin input to generate an output or control a shifter. This sets the maximum setup time at 1.5 FLEXIO clock cycles.

If an input is used by more than one timer or shifter, then the synchronization occurs once to ensure any edge is seen on the same cycle by all timers and shifters using that input.

NOTE

FLEXIO pins are also connected internally. Configuring a FLEXIO shifter or timer to output data on an unused pin makes an internal connection that allows other shifters and timers to use this pin as an input. This allows a shifter output to trigger a timer or a timer output to be shifted into a shifter. This path is also synchronized with the FLEXIO clock and therefore incurs a one-cycle latency.

When using a pin input as a timer trigger, timer clock, or shifter data input, the following synchronization delays occur:

- 0.5–1.5 FLEXIO clock cycles for an external pin
- One FLEXIO clock cycle for an internally driven pin

See Application information for timing considerations such as output valid time and input setup time for specific applications (SPI controller, SPI target, I2C controller, I2S controller, and I2S target).

67.3.3.3 Pin override

You can change the state of any FLEXIO pin at any time. Pin Output Enable (PINOUTE) configures any pin as an output and drives that pin with the value in Pin Output Data (PINOUTD).

Alias registers for PINOUTE and data registers also exist. Writing a logic 1 to an alias register updates the corresponding register fields in both PINOUTE and PINOUTD as follows:

- Pin Output Disable (PINOUTDIS) clears Pin Output Enable (PINOUTE) and Pin Output Data (PINOUTD).
- Pin Output Clear (PINOUTCLR) sets Pin Output Enable (PINOUTE) and clears Pin Output Data (PINOUTD).
- Pin Output Set (PINOUTSET) sets Pin Output Enable (PINOUTE) and Pin Output Data (PINOUTD).
- Pin Output Toggle (PINOUTTOG) sets Pin Output Enable (PINOUTE) and toggles Pin Output Data (PINOUTD).

67.3.3.4 Pin interrupt

You can read the state of any FLEXIO pin at any time and also configure any pin to set a status flag when either a rising or falling edge is detected on that pin. Additionally, you can configure the pin status flag to generate an interrupt.

67.3.4 Low-power modes

FLEXIO remains functional during low-power modes, if CTRL[DOZEN] is 0 and the FLEXIO functional clock remains enabled.

The exception to this is in Deep Sleep mode. In this case, CTRL[DOZEN] is ignored and FLEXIO waits for all timers to complete any pending operation before acknowledging Deep Sleep mode entry.

67.3.5 Debug mode

FLEXIO remains functional in Debug mode, provided the value of CTRL[DBGE] is 1.

67.3.6 Clocking

Table 617. FLEXIO clocks

Clock	Description
Functional clock	Is asynchronous to the bus clock and can remain enabled in low-power modes. You must enable the FLEXIO functional clock before accessing any of the FLEXIO registers. Provided the FLEXIO functional clock is at least equal to the bus clock, you can configure CTRL[FASTACC] to support fast register accesses.
Bus clock	Is used only for bus accesses to the control and configuration registers.

67.3.7 Reset

Table 618. FLEXIO reset types

Reset	Description
Chip reset	Resets the FLEXIO logic and registers to their default states on chip reset.
Software reset	Resets, using CTRL[SWRST], all logic and registers to their default states, except for the Control register.

67.3.8 Interrupts and DMA requests

The following table shows the status flags that generate the FLEXIO interrupt and DMA requests.

Table 619. FLEXIO interrupts and DMA requests

Flag	Description	Interrupt	DMA request	Low-power wake-up
SHIFTSTAT[SSF]	Shifter status flag	Y	Y	Y
SHIFTERR[SEF]	Shifter error flag	Y	N	Y
TIMSTAT[TSF]	Timer status flag	Y	Y	Y
PINSTAT[PSF]	Pin status flag	Y	N	Y
TRGSTAT[ETSF]	External trigger status flag	Y	N	Y

67.3.9 Peripheral triggers

The connection between FLEXIO peripheral triggers and other peripherals is device-specific.

67.3.9.1 Output triggers

Each FLEXIO timer generates an output trigger equal to the timer output. The output trigger is not affected by the timer pin polarity configuration.

67.3.9.2 Input trigger

FLEXIO supports multiple external trigger inputs that can be used to trigger one or more FLEXIO timers. If a rising edge is detected on an external trigger when FLEXIO is enabled, then the external trigger status flag is set. The external triggers are synchronized to the FLEXIO functional clock and must assert for at least two cycles of the FLEXIO functional clock to be sampled correctly.

67.4 External signals

Table 620. External signals

Signal	Description	Direction
FXIO_D*n* (*n* = 0...15)	Bidirectional FLEXIO shifter and timer pin	Input or output

67.5 Initialization

Perform the following procedure to initialize FLEXIO registers:

1. Enable FLEXIO by writing 1 to CTRL[FLEXEN].
2. Configure shift registers for the given application. It is recommended to write to Shifter Configuration (SHIFTCFG0 - SHIFTCFG7) before writing to the corresponding register, Shifter Control (SHIFTCTL0 - SHIFTCTL7).

3. Configure timer registers for the given application. It is recommended to write to Timer Compare (TIMCMP0 - TIMCMP7) and Timer Configuration (TIMCFG0 - TIMCFG7) before writing to the corresponding register, Timer Control (TIMCTL0 - TIMCTL7).
4. Enable interrupts and/or DMA requests, as appropriate, for the given application.
5. Write transmit data to initiate a transfer (depending on the given application).

67.6 Application information

This section provides examples for a variety of FLEXIO module applications. See FLEXIO register descriptions for more information.

67.6.1 UART transmit

UART transmit can be supported using one timer, one shifter, and one pin (two pins, if supporting CTS). The start and stop bit insertion is handled automatically, and multiple transfers are supported using the DMA controller. The timer status flag is used to indicate when the stop bit of each word is transmitted.

Break and idle characters require software intervention. Before transmitting a break or idle character, you must modify SHIFTCFG*n*[SSTART] and SHIFTCFG*n*[SSTOP] to transmit the required state, and the data to transmit must equal FFh or 00h. Supporting a second stop bit requires the stop bit to be inserted into the data stream using software (and increasing the number of bits to transmit). When performing byte writes to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) (or Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) for transmitting MSB first), the rest of the register remains unaltered. This allows an address mark bit or additional stop bit to remain undisturbed.

NOTE

FLEXIO does not support automatic insertion of parity bits.

Table 621. UART transmit configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0. You can configure the PINPOL field to invert output data, or support open-drain by writing 1h to the PINPOL and PINCFG fields.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0000_2222h	Configure start bit, stop bit, enable on trigger asserted and disable on compare. You can support CTS by configuring the TIMENA field as 3h.
TIMCTL*n*	01C0_0001h	Configure the dual 8-bit counter using the shifter 0 status flag as an inverted internal trigger source. To support CTS, configure the PINSEL (for pin 1) and PINPOL fields as 1h.

Table continues on the next page...

Table 621. UART transmit configuration (continued)

Register	Value	Configuration
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer. Use the shifter status flag to indicate when data can be written using an interrupt or a DMA request. Write to SHIFTBUFBBS[7:0] instead to support MSB first transfer.

The following table shows an alternative configuration that supports slower baud rates. This configuration requires two timers.

Table 622. UART transmit configuration for slow baud rate

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0. Invert output data by writing 1 to the PINPOL field. Support open-drain by configuring the PINPOL and PINCFG fields as 1h.
TIMCMP*n*	0000_000Fh	Configure for 8-bit transfer, and configure TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG*n*	0030_2622h	Configure start bit, stop bit, enable on trigger rising edge, decrement on trigger and disable on compare.
TIMCTL*n*	0740_0003h	Configure the 16-bit counter using the timer 1 output as an internal trigger source.
TIMCMP(*n* + 1)	0000_0001h	Configure baud rate of divide by 4 of the FLEXIO clock, and configure TIMCMP[15:0] as (baud rate divider ÷ 2) - 1.
TIMCFG(*n* + 1)	0000_1200h	Configure enable on trigger asserted and disable on timer 0 disable. You can configure the TIMEN field as 3h to support CTS.
TIMCTL(*n* + 1)	01C0_0003h	Configure the 16-bit counter using the shifter 0 status flag as an inverted internal trigger source. You can support CTS by configuring the PINSEL (for pin 1) and PINPOL fields as 1h.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer. Use the shifter status flag to

Table continues on the next page...

Table 622. UART transmit configuration for slow baud rate (continued)

Register	Value	Configuration
		indicate when data can be written using an interrupt or a DMA request. Write to SHIFTBUFBBS[7:0] instead to support MSB first transfer.

67.6.2 UART receive

UART receive can be supported using one timer, one shifter, and one pin (two timers and two pins, if supporting RTS). The start and stop bit verification is handled automatically and multiple transfers are supported using the DMA controller. The timer status flag is used to indicate when the stop bit of each word is received.

FLEXIO does not support triple voting of the received data, which is sampled only once in the middle of each bit. You can use a timer to implement a glitch filter on the incoming data and a different timer to detect an idle line of programmable length. Break characters cause the error flag to set, and the shifter buffer register returns 00h.

NOTE

FLEXIO does not support automatic verification of parity bits.

Table 623. UART receiver configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0080_0001h	Configure receive using timer 0 on the negative edge of clock with input data on pin 0. You can invert input data by writing 1 to the PINPOL field.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0204_2422h	Configure start bit, stop bit, enable on pin positive edge and disable on compare. Enable resynchronization to received data with TIMOUT = 2h and TIMRST = 4h.
TIMCTL*n*	0000_0081h	Configure the dual 8-bit counter using the inverted pin 0 input.
SHIFTBUF*n*	Data to receive	You can read received data from SHIFTBUFBYS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from SHIFTBUFBIS[7:0] instead to support MSB first transfer.

The UART receiver with RTS configuration uses a second timer to generate the RTS output. RTS asserts when the start bit is detected and negates when the data is read from the shifter buffer register. If no start bit is detected when the RTS is asserted, the received data is ignored.

Table 624. UART receiver with RTS configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0080_0001h	Configure receive using timer 0 on the negative edge of clock with input data on pin 0. Invert input data by writing 1 to the PINPOL field.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0204_2522h	Configure start bit, stop bit, enable on pin positive edge with trigger asserted and disable on compare. Enable resynchronization to received data with TIMOUT = 2h and TIMRST = 4h.
TIMCTL*n*	02C0_0081h	Configure dual 8-bit counter using the inverted pin 0 input. Trigger is internal using the inverted pin 1 input.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.
TIMCFG(*n* + 1)	0030_6100h	Enable on timer *n* enable and disable on the trigger falling edge. Decrement on trigger to ensure no compare.
TIMCTL(*n* + 1)	0143_0003h	Configure 16-bit counter and output on pin 1. Trigger is internal using the shifter 0 flag.
SHIFTBUF*n*	Data to receive	You can read received data using SHIFTBUFBYS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from SHIFTBUFBIS[7:0] instead to support MSB first transfer.

67.6.3 SPI controller

SPI Controller mode can be supported using two timers, two shifters, and four pins. Using the DMA controller, either CPHA = 0 or CPHA = 1 and transfers can be supported. For CPHA = 1, the chip select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The stop bit is used to guarantee a minimum of one clock cycle between the target chip select negating and before the next transfer. To initiate each transfer, either the core or DMA writes to the transmit buffer.

NOTE

Because of synchronization delays, the setup time for the serial input data is 1.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 4 of the FLEXIO clock frequency.

Table 625. SPI controller (CPHA = 0) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0083_0002h	Configure transmit using timer 0 on the negative edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0000_0101h	Configure receive using timer 0 on the positive edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0100_2222h	Configure start bit, stop bit, enable on trigger high and disable on compare; initial clock state is logic 0.
TIMCTL*n*	01C3_0201h	Configure dual 8-bit counter using the pin 2 output (shift clock), with shifter 0 flag as the inverted trigger. Write 1 to the PINPOL field to invert the output shift clock.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.
TIMCFG(*n* + 1)	0000_1100h	Enable when timer 0 is enabled and disable when timer 0 is disabled.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter (never compare) using the inverted pin 3 output as target select.
SHIFTBUF*n*	Data to transmit	You can write transmit data to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

Table 626. SPI controller (CPHA = 1) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0021h	Start bit loads data on first shift.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on the negative edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits x 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0100_2222h	Configure start bit, stop bit, enable on trigger high and disable on compare; initial clock state is logic 0.
TIMCTL*n*	01C3_0201h	Configure dual 8-bit counter using pin 2 output (shift clock), with the shifter 0 flag as the inverted trigger. Write 1 to the PINPOL field to invert the output shift clock, and set the TIMDIS field as 3 to keep target select asserted for as long as there is data in the transmit buffer.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.
TIMCFG(*n* + 1)	0000_1100h	Enable when timer 0 is enabled and disable when timer 0 is disabled.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter (never compare) using inverted pin 3 output (as target select).
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF. Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

67.6.4 SPI target

SPI Target mode can be supported using one timer, two shifters, and four pins. Either CPHA = 0 or CPHA = 1 can be supported and transfers can be supported using the DMA controller. For CPHA = 1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

You must write the transmit data to the transmit buffer register before the external target select asserts; otherwise, the shifter error flag is set.

NOTE

Because of synchronization delays, the output valid time for the serial output data is 2.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

Table 627. SPI target (CPHA = 0) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0083_0002h	Configure transmit using timer 0 on the falling edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0000_0101h	Configure receive using timer 0 on the rising edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_003Fh	Configure 32-bit transfer. Set TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG*n*	0120_0600h	Configure enable on trigger rising edge. Initial clock state is logic 0 and decrements on pin input.
TIMCTL*n*	06C0_0203h	Configure 16-bit counter using pin 2 input (shift clock), with pin 3 input (target select) as the inverted trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

Table 628. SPI target (CPHA = 1) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0001h	Shifter configured to load on first shift and stop bit disabled.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on rising edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on falling edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_003Fh	Configure 32-bit transfer. Set TIMCMP[15:0] as (number of bits × 2) - 1).
TIMCFG*n*	0120_6602h	Configure start bit, enable on trigger rising edge, disable on trigger falling edge. Initial clock state is logic 0 and decrements on pin input.
TIMCTL*n*	06C0_0203h	Configure 16-bit counter using pin 2 input (shift clock), with pin 3 input (target select) as the inverted trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

67.6.5 I2C controller

I2C Controller mode can be supported using two timers, two shifters, and two pins. One timer is used to generate the SCL output and another one is used to control the shifters. The two shifters that are used to transmit and receive for every word, when receiving the transmitter, must transmit FFh to 3-state the output. FLEXIO inserts a stop bit after every word to generate and verify the ACK or NACK. FLEXIO waits for the first write to the transmit data buffer before enabling SCL generation. Data transfers can be supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The first timer generates the bit clock for the entire packet (start to repeated start or stop condition), so you must program the compare register with the total number of clock edges in the packet (minus one). The timer supports clock stretching using the reset counter when pin is equal to output. However, this increases both the clock high and clock low periods by at least one FLEXIO

clock cycle each. The second timer uses the SCL input pin to control the transmit and receive shift registers. This enforces an SDA data hold time by an extra two FLEXIO clock cycles.

Both the transmit and receive shifters must be serviced for each word in the transfer. The transmit shifter must transmit FFh when receiving, and the receive shifter returns the data present on the SDA pin. The transmit shifter loads one additional word on the last falling edge of the SCL pin. When generating a stop condition or a repeated start condition, this word must be 00h and FFh, respectively. During the last word of a controller-receiver transfer, you must set the transmit SHIFTCFG*n*[SSTOP] field to generate a NACK.

The receive shift register asserts an error interrupt if a NACK is detected, but you are responsible for generating the stop or repeated start condition. If a NACK is detected during controller-transmit, the interrupt routine must immediately write 00h (when generating a stop condition) or FFh (when generating a repeated start condition) to the transmit shifter register. You must wait for the next rising edge on SCL before disabling both timers. The transmit shifter must be disabled after the setup delay for a repeated start or stop condition.

NOTE

Because of synchronization delays, the data valid time for the transmit output is two FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

To guarantee SDA hold time, the I2C controller data valid is delayed by two cycles because the clock output is passed through a synchronizer before clocking the transmit or receive shifter. Because the SCL output is synchronous with FLEXIO clock, the synchronization delay is one cycle, and then an additional cycle is involved to generate the output.

Table 629. I2C controller configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Start bit enabled (logic 0) and stop bit enabled (logic 1).
SHIFTCTL*n*	0101_0082h	Configure transmit using timer 1 on the rising edge of clock with inverted output enable (open-drain output) on pin 0.
SHIFTCFG(*n* + 1)	0000_0020h	Start bit disabled and stop bit enabled (logic 0) for ACK or NACK detection.
SHIFTCTL(*n* + 1)	0180_0001h	Configure receive using timer 1 on the falling edge of clock with input data on pin 0.
TIMCMP*n*	0000_2501h	Configure 2 word transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of words × 18) + 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0102_2222h	Configure start bit, stop bit, enable on trigger high, disable on compare, reset if output equals pin. Initial clock state is logic 0 and is not affected by reset.
TIMCTL*n*	01C1_0101h	Configure dual 8-bit counter using pin 1 output enable (SCL open-drain), with the shifter 0 flag as the inverted trigger.
TIMCMP(*n* + 1)	0000_000Fh	Configure 8-bit transfer. Set TIMCMP[15:0] as (number of bits x 2) - 1.

Table continues on the next page...

Table 629. I2C controller configuration (continued)

Register	Value	Configuration
TIMCFG(*n* + 1)	0020_1112h	Enable when timer 0 is enabled; disable when timer 0 is disabled. Enable start bit and stop bit at the end of each word and decrement on pin input.
TIMCTL(*n* + 1)	01C0_0183h	Configure 16-bit counter using inverted pin 1 input (SCL).
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUFBBS[7:0]. Use the shifter status flag to indicate when data can be written using interrupt or DMA request.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from SHIFTBUFBIS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

67.6.6 I2S controller

I2S Controller mode can be supported using two timers, two shifters, and four pins. One timer is used to generate the bit clock and control the shifters and another timer is used to generate the frame sync. FLEXIO waits for the first write to the transmit data buffer before enabling bit clock and frame sync generation. Data transfers are supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The bit clock frequency is an even integer divide of the FLEXIO clock frequency. The initial frame sync assertion occurs at the same time as the first bit clock edge. The timer uses the start bit to ensure that the frame sync is generated one clock cycle before the first output data.

NOTE

Because of synchronization delays, the setup time for the receiver input is 1.5 FLEXIO clock cycles. This means that the maximum baud rate is divide by 4 of the FLEXIO clock frequency.

Table 630. I2S controller configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0001h	Load transmit data on first shift and stop bit disabled.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the rising edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on the falling edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2)

Table continues on the next page...

Table 630. I2S controller configuration (continued)

Register	Value	Configuration
		- 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0000_0202h	Configure start bit, enable on trigger high and never disable. Initial clock state is logic 1.
TIMCTL*n*	01C3_0281h	Configure dual 8-bit counter using inverted pin 2 output (bit clock), with shifter 0 flag as the inverted trigger. Write 0 to the PINPOL field to invert the polarity of the output shift clock.
TIMCMP(*n* + 1)	0000_007Fh	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:0] as (number of bits × baud rate divider) ÷ 1.
TIMCFG(*n* + 1)	0000_0100h	Enable when timer 0 is enabled and never disable.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter using inverted pin 3 output (as frame sync). Write 0 to the PINPOL field to invert the polarity of the output frame sync.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) instead to support LSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) instead to support LSB first transfer.

67.6.7 I2S target

I2S Target mode can be supported using three timers, two shifters, and four pins. For single transmit and single receive, other combinations of transmit and receive are possible.

The transmit data must be written to the transmit buffer register before the external frame sync asserts, otherwise the shifter error flag is set.

NOTE

Because of synchronization delays, the output valid time for the serial output data is 2.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

The output valid time of I2S target is maximum 2.5 cycles because there is a maximum 1.5 cycle delay on the clock synchronization, plus one cycle to output the data.

Timer 2 detects the falling edge of frame sync (start of new frame) and asserts output until the rising edge of bit clock (when the frame sync is normally sampled). Timer 0 detects the rising edge of bit clock with timer 2 output asserted and asserts output for length of frame. Timer 1 detects the falling edge of bit clock with timer 0 output asserted and controls shift registers for 32-bit transfers.

Table 631. I2S target configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0103_0002h	Configure transmit using timer 1 on the rising edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0180_0101h	Configure receive using timer 1 on the falling edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_007Fh	Configure two 32-bit transfers per frame. Set TIMCMP[15:0] as (number of bits × 4) - 1.
TIMCFG*n*	0020_2500h	Configure enable on pin rising edge (inverted bit clock) with trigger high (timer 2) and disable on compare. Initial clock state is logic 1 and decrements on pin input (bit clock).
TIMCTL*n*	0B40_0203h	Configure 16-bit counter using pin 2 input (bit clock), with timer 2 output as the trigger.
TIMCMP(*n* + 1)	0000_003Fh	Configure 32-bit transfers. Set TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG(*n* + 1)	0020_2500h	Configure enable on pin (bit clock) rising edge with trigger (timer 0) high and disable on compare. Initial clock state is logic 1 and decrement on pin input (bit clock).
TIMCTL(*n* + 1)	0340_0283h	Configure 16-bit counter using inverted pin 2 input (bit clock), with timer 0 output as the trigger.
TIMCMP(*n* + 2)	0000_0000h	Compare on zero (first edge).
TIMCFG(*n* + 2)	0020_6400h	Configure enable on inverted pin (frame sync) rising edge and disable on trigger falling edge (bit clock). Initial clock state is logic 1 and decrement on inverted pin input (frame sync).

Table continues on the next page...

Table 631. I2S target configuration (continued)

Register	Value	Configuration
TIMCTL(*n* + 2)	04C0_0383h	Configure 16-bit counter using inverted pin 3 input (frame sync), with pin 2 inverted input (bit clock) as the trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to the SHIFTBUF register instead to support LSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from the SHIFTBUF register instead to support LSB first transfer.

67.6.8 SENT receiver

The SENT receiver can be supported by using one timer and one pin. The timer is configured as Input-Capture mode, and captures the counter value at the falling edge of the pin input. After the counter value is captured, the counter is automatically restarted from counter value 0. Therefore, the captured value always indicates the period between the previous falling edge and current falling edge. You can configure the CPU interrupt or DMA trigger at each capture of the counter. The CPU software performs the entire SENT frame decoding with the latest tick width adjustment.

Table 632. SENT receiver configuration

Register	Value	Configuration
TIMCMP*n*	—	Stores counter value at the falling edge of the pin.
TIMCFG*n*	0000_6000h	Timer is always enabled. It is disabled on trigger falling edge, decrement counter on FLEXIO clock.
TIMCTL*n*	0040_0007h	Single 16-bit input capture mode. The PINSEL field selects the timer pin input and output. Timer pin output disabled, internal trigger selected, select pin 0 as trigger.

67.6.9 Camera interface

Camera interface can be supported using one timer, one or more shifters, and multiple pins. Multiple transfers can be supported using the DMA controller.

The example configuration shown in the following table describes the FLEXIO configuration for interfacing with an 8-bit CMOS sensor with PCLK, VSYNC, HREF, and D[7:0] outputs. The example uses a 128-bit buffer to capture 16 pixels of image data before an interrupt or DMA transfer. You can use a bigger or smaller buffer depending on system DMA performance and FLEXIO resource usage by other applications.

NOTE

You can use additional timers to track the number of pixels per row and number of rows per frame, or HREF or VSYNC can be assigned as GPIO interrupts for software tracking.

Table 633. Camera interface configuration for 8-bit CMOS sensor

Register	Value	Configuration
SHIFTCFG*n*...*n*+2[1]	0007_0100h	Configure 8-bit parallel shift in from adjacent shifter.
SHIFTCFG*n*+3	0007_0000h	Configure 8-bit parallel shift in from pins FXIO_D[7:0] (D[7:0]).
SHIFTCTL*n*...*n*+3	0080_0001h	Configure receive using timer 0 on the negative edge of clock.
TIMCMP*n*	0000_001Fh	Configure 16 pixel (8 bits/pixel × 16 pixels = 128 bits) transfer. Set TIMCMP[15:0] as (number of pixels × 2) - 1.
TIMCFG*n*	0120_6600h	Configure enable on trigger (HREF) rising edge and disable on trigger falling edge. Initial shift clock state is logic 0 and decrement on PCLK input.
TIMCTL*n*	12C0_0803h	Configure 16-bit counter using FXIO_D[8] input (PCLK), with FXIO_D[9] input (HREF) as the inverted trigger.
SHIFTBUF*n*...*n*+3	Data to receive	Received data can be read from SHIFTBUF*n*...*n*+3. Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

1. *n* = 0 or 4

67.6.10 Motorola 68K and Intel 8080 bus interface

Motorola 68K and Intel 8080 bus are two parallel interfaces commonly used by smart and asynchronous LCD controllers. With GPIO, FLEXIO can drive these interfaces using one timer and one shifter. Additional shifters can be used to support large transfers via the DMA controller.

Example configurations are provided for 1-beat and 16-beat transfers. Other lengths are also supported.

The following table provides an example of how to drive a 16-bit 68K or 8080 bus. For an 8080 bus, two GPIOs are used to drive the nCS and RS pins. For a 68K bus, an additional GPIO is required to drive the RDWR pin.

Table 634. Motorola 68K and Intel 8080 write configuration

Register	Value	Configuration
SHIFTCFG0...7	000F_0100h	Configure 16-bit parallel shift in from adjacent shifter.
SHIFTCTL0	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with data output to FXIO_D[15:0].

Table continues on the next page...

Table 634. Motorola 68K and Intel 8080 write configuration (continued)

Register	Value	Configuration
SHIFTCTL1...7	0000_0002h	Configure transmit using timer 0 on the positive edge of clock.
TIMCMP0	0000_0101h (1 beat) 0000_1F01h (16 beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of beats × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_2200h	Configure enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	01C3_1001h (Motorola 68K, 1 beat) 1DC3_1001h (Motorola 68K, 16 beats) 01C3_1081h (Intel 8080, 1 beat) 1DC3_1081h (Intel 8080, 16 beats)	Configure dual 8-bit counter using FXIO_D[16] output (EN pin for 68K, nWR pin for 8080), with shifter 0 (1-beat) or shifter 7 (16 beats) flag as the inverted trigger.
SHIFTBUF0...7	Data to transmit	Transmit data can be written to SHIFTBUF0 (1 beat) or SHIFTBUF0...7 (16-beats) to initiate a transfer; use the shifter status flag to indicate when data can be written using interrupt or DMA request.

Table 635. Motorola 68K and Intel 8080 read configuration

Register	Value	Configuration
SHIFTCFG0...6	000F_0100h	Configure 16-bit parallel shift in from the adjacent shifter.
SHIFTCFG7	000F_0000h	Configure 16-bit parallel shift in from pin.
SHIFTCTL0...7	0080_0001h	Configure receive using timer 0 on the negative edge of clock with data input from FXIO_D[15:0].
TIMCMP0	0000_0101h (1 beat) 0000_1F01h (16 beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] = (number of beats × 2) - 1, and set TIMCMP[7:0] = (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_2220h	Configure stop_bit. Enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	1DC3_1001h (Motorola 68K, 1 beat) 01C3_1001h (Motorola 68K, 16 beats)	Configure dual 8-bit counter using either FXIO_D[16] output (EN pin for 68K) or FXIO_D[17] output (nRD pin for 8080),

Table continues on the next page...

Table 635. Motorola 68K and Intel 8080 read configuration (continued)

Register	Value	Configuration
	1DC3_1181h (Intel 8080, 1 beat) 01C3_1181h (Intel 8080, 16 beats)	with shifter 7 flag (1-beat) or shifter 0 flag (16 beats) as the inverted trigger.
SHIFTBUF0...7	Data received	Received data can be read from SHIFTBUF0 (1 beat) or SHIFTBUF0...7 (16 beats). Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

In general, any operation to a 68K or 8080 bus target begins with a register write cycle followed by one or more data read or write cycles. To accomplish this, perform the following procedure:

1. Configure FLEXIO with 1-beat write configuration.
2. Configure GPIO to assert the nCS and RS pins (and deassert the RDWR pin for 68K).
3. Write register index data to SHIFTBUF0[15:0].
4. Configure GPIO to deassert the RS pin (and assert the RDWR pin for 68K data read).
5. Configure FLEXIO with the desired read or write configuration (1 or 16 beats).
6. Use the shifter status flag to trigger interrupt or DMA driven data transfers to and from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7).
7. Configure GPIO to deassert the nCS pin.

67.6.11 Low-power state machine

Table 636 shows an example of a hypothetical state machine to illustrate the flexibility allowed in Shifter State mode.

In this example, FLEXIO waits for the FXIO_D[2] pin to assert and then drives a complementary square wave output at a frequency of FLEXIO_CLK/131072 on the FXIO_D[1:0] pins while the comparator output is asserted. This assumes that the comparator is connected to external trigger 15. See the chip-specific FLEXIO information for actual FLEXIO trigger mappings. Throughout this operation, the CPU can be kept in a stop or VLPS mode by writing 0 to CTRL[DOZEN] and ensuring that FLEXIO_CLK is enabled. Figure 280 shows the states and transitions implemented by this example.

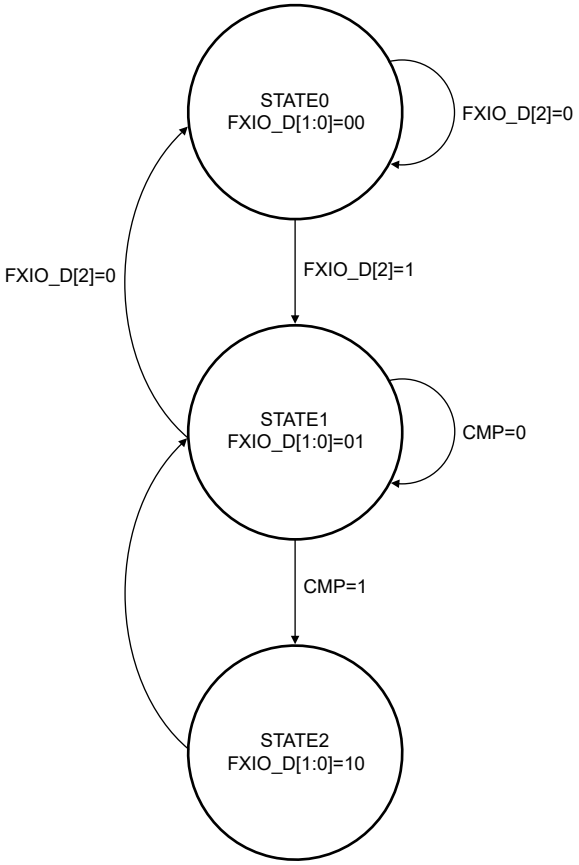


Figure 280. State diagram

Table 636. State machine configuration

Register	Value	Configuration
SHIFTCFG0...2	0000_0003h	Enable FXIO_D[1:0] as outputs.
SHIFTCTL0	0080_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output low to trigger next state.
SHIFTBUF0	0020_8208h	State0: Drive FXIO_D[1:0] = 00; transition to state0 if FXIO_D[2] = 0, state1 if FXIO_D[2] = 1.
SHIFTCTL1	0000_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output high to trigger next state.
SHIFTBUF1	0140_8408h	State1: Drive FXIO_D[1:0]=01; transition to state0 if FXIO_D[2]=0, state1 if CMP = 0, state2 if CMP = 1 (FXIO_D[3]=1).

Table continues on the next page...

Table 636. State machine configuration (continued)

Register	Value	Configuration
SHIFTCTL2	0080_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output low to trigger next state.
SHIFTBUF2	0224_9249h	State2: Drive FXIO_D[1:0] = 10, transition to state1 with timer 0 output low.
TIMCMP0	0000_FFFFh	Configure baud rate of divide by 131072 of the FLEXIO clock. Set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_0000h	Configure timer always enabled.
TIMCTL0	0000_0003h	Configure single 16-bit counter.
TIMCFG1	0010_7600h	Configure timer enabled on trigger rising edge, disabled on trigger falling edge, decrement on trigger edge.
TIMCTL1	0F03_0303h	Configure timer output enabled on FXIO_D[3], external trigger 15 (comparator output) selected.

67.6.12 Keypad interface

The keypad interface can support a 3 × 4 keypad matrix using three timers and three shifters, although a larger matrix can be supported using additional shifters. The configuration is designed for four columns configured as active low open-drain pins and three rows configured as input pins with pull-up resistors enabled.

One shifter is configured in Logic mode to assert its output when any of the row inputs are low, indicating a key is pressed. Use a timer to filter the shifter output to ensure that the key is pressed for a minimum amount of time before performing the column scan.

A different shifter is configured for parallel transmit. Use this shifter to scan each column when a keypress is detected. When not scanning, the shifter output is configured to assert all active low open-drain column outputs to detect any keypress. Use a dedicated timer to control the transmit shifter.

The last shifter is configured for parallel receive. Use this shifter to capture the result of the column scanning so that you can decode which key (or keys) was pressed. This configuration captures the state of both row and column pins for each scan, although the row state can also be deduced by the shift order. Use a dedicated timer to control the receive shifter, which shifts at half the frequency of the transmit shifter.

When the result of the key scan is available in the receive shifter register, FLEXIO continues to monitor the row inputs and can trigger multiple scans from a single keypress. To support debouncing, you can decide how many consecutive scans must be considered as a single keypress.

NOTE

Because the pins used in Logic mode are fixed per shifter and the shifters that support parallel shifts are limited, this configuration is restricted to what pins and shifters it can use. Increasing the matrix beyond four-row inputs requires multiple shifters in Logic mode. Increasing beyond four-column outputs requires concatenating transmit shifters to create a larger shift register.

Table 637. Keypad interface configuration

Register	Value	Configuration
SHIFTCFG0	0003_0032h	Start bit enabled (logic 0) and stop bit enabled (logic 1), configured for 4-bit shift.
SHIFTCTL0	0101_0402h	Configure transmit using timer 1 on the rising edge of clock generating open-drain output on pins[7:4] (column outputs).
SHIFTBUF0	0804_0201h	Static data containing the column scan pattern; each column is scanned one-hot with dead time in between.
SHIFTCFG1	0000_0020h	In Logic mode, mask input 3.
SHIFTCTL1	0000_0007h	Configure Logic mode.
SHIFTBUF1	07FF_07FFh	Static data configuring Logic mode LUT. Output asserts if pins [3:1] are logic 0.
SHIFTCFG3	0007_0000h	Configured for 8-bit shift.
SHIFTCTL3	0280_0001h	Configure receive using timer 2 on falling edge of clock with input data on pins [7:0] (both rows and columns; pin 0 is don't care).
TIMCMP0	0000_00ffh	Configure prescanning glitch filter to 256 FLEXIO clock cycles. For different filter cycles, configure TIMCMP[15:0] as (filter cycles) - 1.
TIMCFG0	0103_6600h	Configure enable on trigger rising edge and disable on trigger falling edge. Initial clock state is logic 0 and is not affected by reset.
TIMCTL0	0540_0003h	Configure 16-bit counter using the shifter 1 flag (logic state) as trigger.
TIMCMP1	0000_0F3Fh	Configure eight shifts (twice for each column) at column scan rate of divide by 128. For a different scan frequency, define TIMCMP[7:0] as (scan divider ÷ 2) - 1.
TIMCFG1	0020_2622h	Enable on trigger rising edge, disable on compare. Start and stop bits are enabled.
TIMCTL1	0340_0001h	Configure dual 8-bit counter using timer 0 output as the trigger.
TIMCMP2	0000_0801h	Configure four shifts at half the frequency of the timer 1 trigger.

Table continues on the next page...

Table 637. Keypad interface configuration (continued)

Register	Value	Configuration
TIMCFG2	0110_2100h	Enable on timer 1 enable, disable on compare, decrement on trigger input with output initially negated, and not affected by reset.
TIMCTL2	0740_0001h	Configure dual 8-bit counter using timer 1 output as the trigger.
SHIFTBUF3	Keypad scan result	Keypad scan result can be read from SHIFTBUF3. Each byte is the result of one scan with both row and column pin state from the scan (pin 0 is not used). Use the shifter status flag to indicate when data can be written using interrupt or DMA request.

67.7 Memory map and registers

67.7.1 FLEXIO register descriptions

NOTE

Invalid register accesses, which include reading a write-only register, writing to a read-only register, or accessing an invalid address, result in a bus error.

67.7.1.1 FLEXIO memory map

FLEXIO base address: 4041_6000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0201_0003h
4h	Parameter (PARAM)	32	R	0410_0808h
8h	FLEXIO Control (CTRL)	32	RW	0000_0000h
Ch	Pin State (PIN)	32	R	0000_0000h
10h	Shifter Status (SHIFTSTAT)	32	RW	0000_0000h
14h	Shifter Error (SHIFTERR)	32	RW	0000_0000h
18h	Timer Status Flag (TIMSTAT)	32	RW	0000_0000h
20h	Shifter Status Interrupt Enable (SHIFTSIEN)	32	RW	0000_0000h
24h	Shifter Error Interrupt Enable (SHIFTEIEN)	32	RW	0000_0000h
28h	Timer Interrupt Enable (TIMIEN)	32	RW	0000_0000h
30h	Shifter Status DMA Enable (SHIFTSDEN)	32	RW	0000_0000h
38h	Timer Status DMA Enable (TIMERSDEN)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
40h	Shifter State (SHIFTSTATE)	32	RW	0000_0000h
48h	Trigger Status (TRGSTAT)	32	RW	0000_0000h
4Ch	External Trigger Interrupt Enable (TRIGIEN)	32	RW	0000_0000h
50h	Pin Status (PINSTAT)	32	RW	0000_0000h
54h	Pin Interrupt Enable (PINIEN)	32	RW	0000_0000h
58h	Pin Rising Edge Enable (PINREN)	32	RW	0000_0000h
5Ch	Pin Falling Edge Enable (PINFEN)	32	RW	0000_0000h
60h	Pin Output Data (PINOUTD)	32	RW	0000_0000h
64h	Pin Output Enable (PINOUTE)	32	RW	0000_0000h
68h	Pin Output Disable (PINOUTDIS)	32	RW	0000_0000h
6Ch	Pin Output Clear (PINOUTCLR)	32	RW	0000_0000h
70h	Pin Output Set (PINOUTSET)	32	RW	0000_0000h
74h	Pin Output Toggle (PINOUTTOG)	32	RW	0000_0000h
80h - 9Ch	Shifter Control (SHIFTCTL0 - SHIFTCTL7)	32	RW	0000_0000h
100h - 11Ch	Shifter Configuration (SHIFTCFG0 - SHIFTCFG7)	32	RW	0000_0000h
200h - 21Ch	Shifter Buffer (SHIFTBUF0 - SHIFTBUF7)	32	RW	0000_0000h
280h - 29Ch	Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7)	32	RW	0000_0000h
300h - 31Ch	Shifter Buffer Byte Swapped (SHIFTBUFBYS0 - SHIFTBUFBYS7)	32	RW	0000_0000h
380h - 39Ch	Shifter Buffer Bit Byte Swapped (SHIFTBUFBBS0 - SHIFTBUFBBS7)	32	RW	0000_0000h
400h - 41Ch	Timer Control (TIMCTL0 - TIMCTL7)	32	RW	0000_0000h
480h - 49Ch	Timer Configuration (TIMCFG0 - TIMCFG7)	32	RW	0000_0000h
500h - 51Ch	Timer Compare (TIMCMP0 - TIMCMP7)	32	RW	0000_0000h
680h - 69Ch	Shifter Buffer Nibble Byte Swapped (SHIFTBUFNBS0 - SHIFTBUFNBS7)	32	RW	0000_0000h
700h - 71Ch	Shifter Buffer Halfword Swapped (SHIFTBUFHWS0 - SHIFTBUFHWS7)	32	RW	0000_0000h
780h - 79Ch	Shifter Buffer Nibble Swapped (SHIFTBUFNIS0 - SHIFTBUFNIS7)	32	RW	0000_0000h
800h - 81Ch	Shifter Buffer Odd Even Swapped (SHIFTBUFOES0 - SHIFTBUFOES7)	32	RW	0000_0000h
880h - 89Ch	Shifter Buffer Even Odd Swapped (SHIFTBUFEOS0 - SHIFTBUFEOS7)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
900h - 91Ch	Shifter Buffer Halfword Byte Swapped (SHIFTBUFHBS0 - SHIFTBUFHBS7)	32	RW	0000_0000h

67.7.1.2 Version ID (VERID)

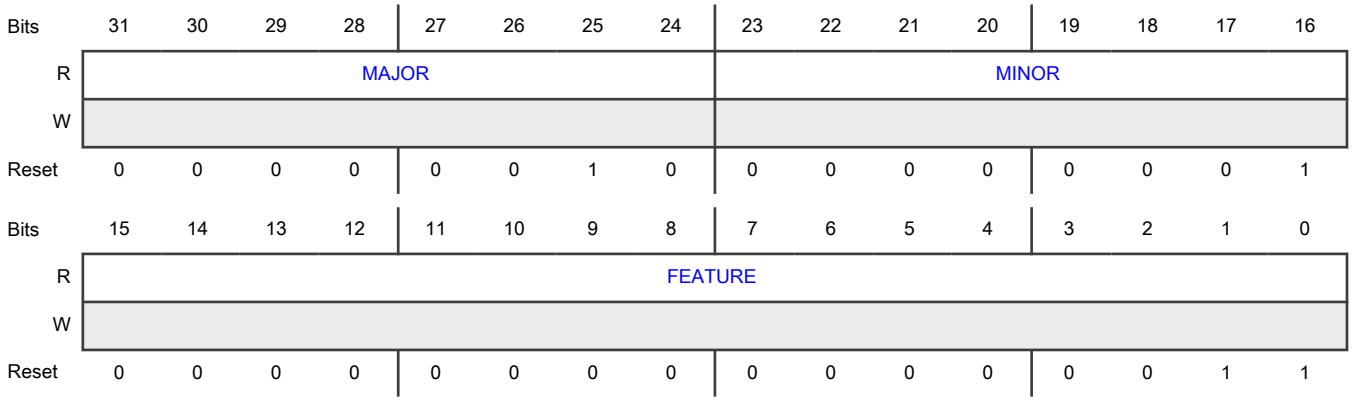
Offset

Register	Offset
VERID	0h

Function

Indicates the version of FLEXIO.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number of the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number of the module specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Standard features implemented

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000_0000_0000_0001b - State, logic, and parallel modes supported 0000_0000_0000_0010b - Pin control registers supported 0000_0000_0000_0011b - State, logic, and parallel modes, plus pin control registers supported

67.7.1.3 Parameter (PARAM)

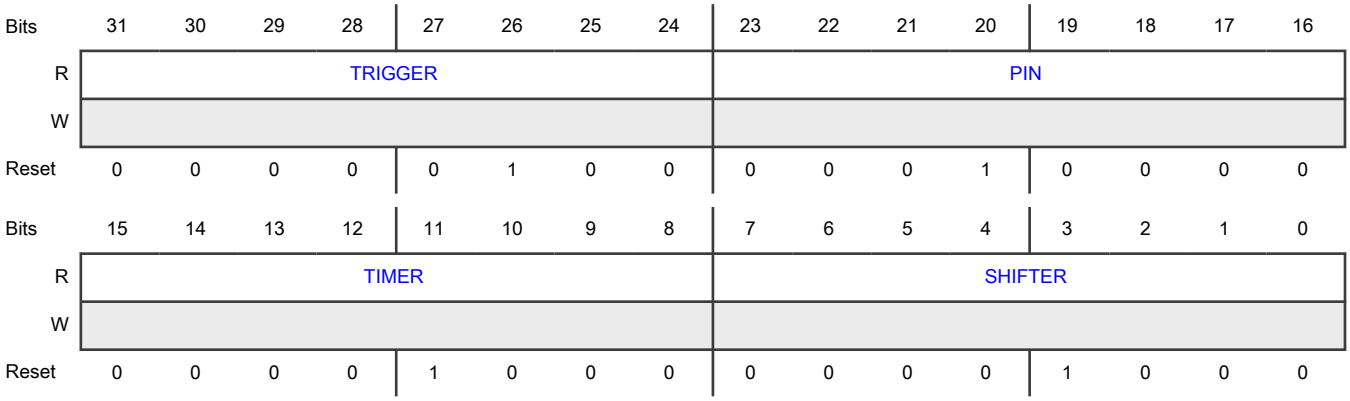
Offset

Register	Offset
PARAM	4h

Function

Contains the number of shifters, timers, pins, and triggers.

Diagram



Fields

Field	Function
31-24 TRIGGER	Trigger Number Indicates the number of external triggers implemented.
23-16 PIN	Pin Number Indicates the number of pins implemented.
15-8 TIMER	Timer Number Indicates the number of timers implemented.
7-0 SHIFTER	Shifter Number Indicates the number of shifters implemented.

67.7.1.4 FLEXIO Control (CTRL)

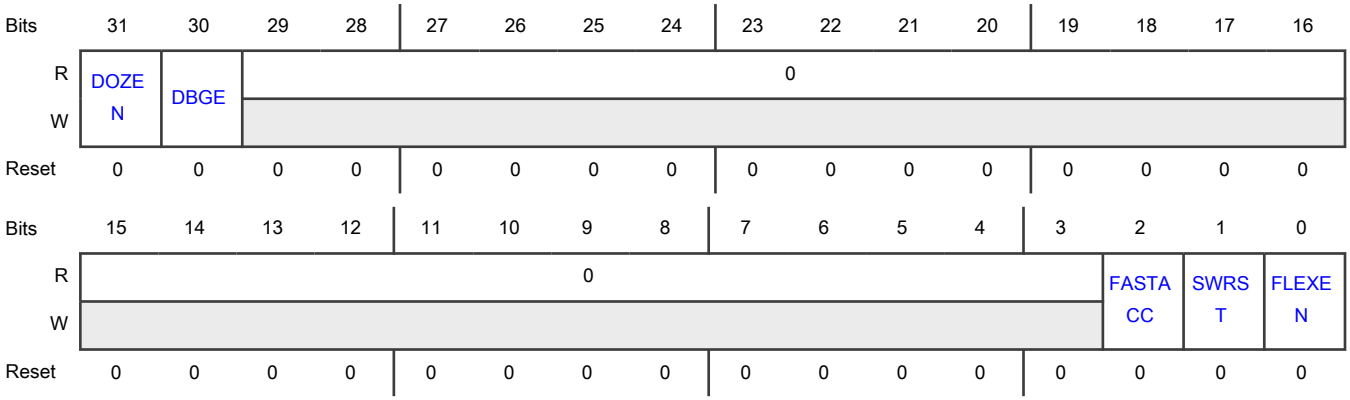
Offset

Register	Offset
CTRL	8h

Function

Controls various aspects of the FLEXIO operation.

Diagram



Fields

Field	Function
31 DOZEN	Doze Enable Disables FLEXIO operation in Doze modes. 0b - Enable 1b - Disable
30 DBGE	Debug Enable Enables the FLEXIO operation in Debug mode. 0b - Disable 1b - Enable
29-3 —	Reserved
2 FASTACC	Fast Access Configures fast or normal register accesses to FLEXIO registers, but requires the FLEXIO functional clock to be at least equal to the frequency of the bus clock.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Normal 1b - Fast
1 SWRST	Software Reset Specifies whether software reset is enabled. The software reset does not affect this register but it affects all other logic in FLEXIO. All other register accesses are ignored until this field is cleared. The field remains 1 until software clears it and the reset has cleared in the FLEXIO clock domain. If you write 1 to this field, all FLEXIO registers except the Control register are reset. 0b - Disabled 1b - Enabled
0 FLEXEN	FLEXIO Enable Enables FLEXIO. 0b - Disable 1b - Enable

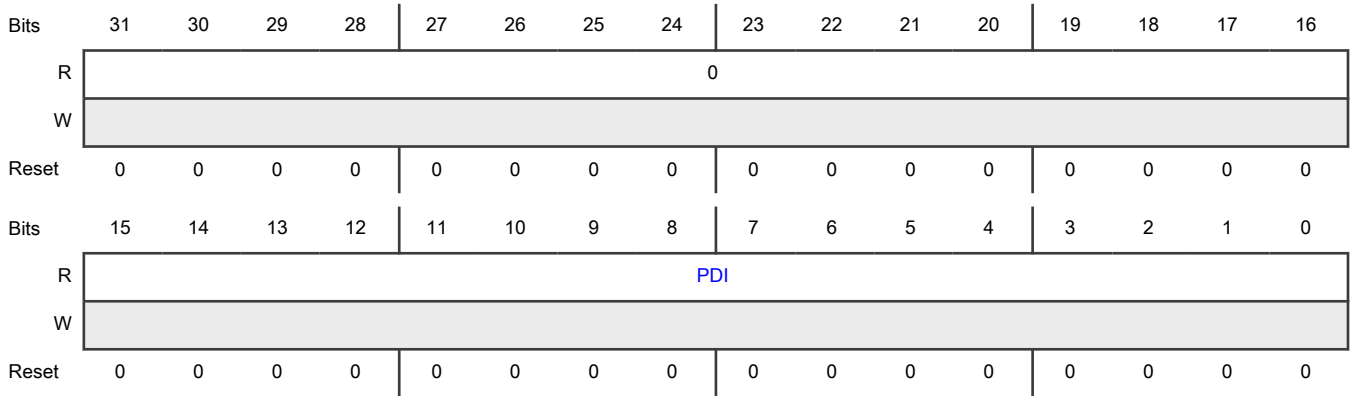
67.7.1.5 Pin State (PIN)

Offset

Register	Offset
PIN	Ch

Function
Indicates the status of the pin data input.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PDI	Pin Data Input Indicates the input data on each of the FLEXIO pins.

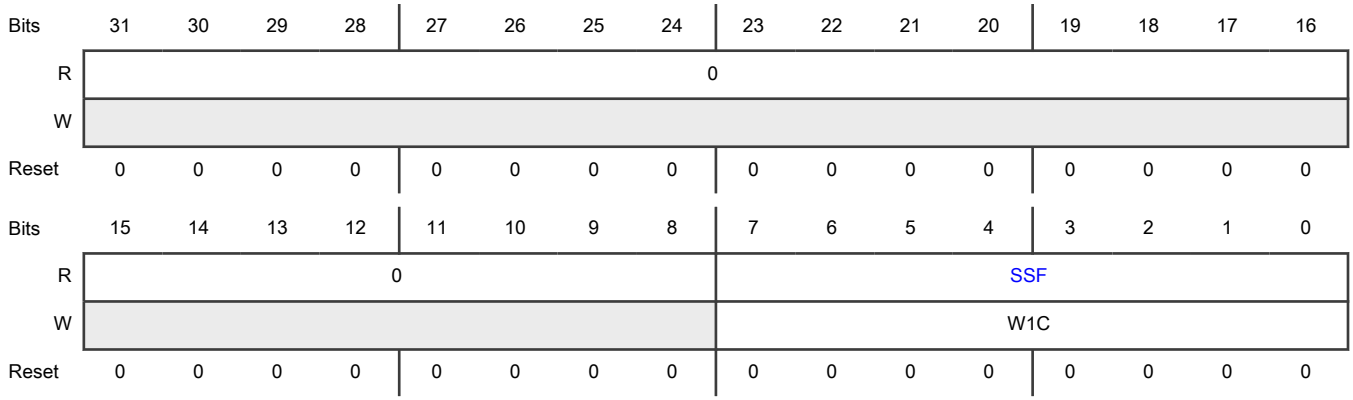
67.7.1.6 Shifter Status (SHIFTSTAT)

Offset

Register	Offset
SHIFTSTAT	10h

Function
Contains shifter status flags.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SSF	Shifter Status Flag Indicates the shifter status. This flag is updated in one of the following cases:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• If SHIFTCTL*n*[SMOD] = 001b (Receive mode), the status flag is set when SHIFTBUF is loaded with data from the shifter (SHIFTBUF is full). The status flag is cleared when you read Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). • If SHIFTCTL*n*[SMOD] = 010b (Transmit mode), the status flag is set when SHIFTBUF data is transferred to the shifter (SHIFTBUF is empty) or when SHIFTCTL*n*[SMOD] is initially configured as 010b (Transmit mode). The status flag is cleared when you write to the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 100b (Match Store mode), the status flag is set when a match occurs between SHIFTBUF and the shifter. The status flag is cleared when you read the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the status flag returns the current match result between SHIFTBUF and the shifter. You cannot clear the status flag by reading the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 110b (State mode), the status flag for a shifter sets when it is selected by the current state pointer. • If SHIFTCTL*n*[SMOD] = 111b (Logic mode), the status flag returns the current value of the programmable logic block output. You can clear this status flag by writing a logic one to the flag for all modes except Match Continuous mode, State mode, and Logic mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000b - Clear 0000_0001b - Set When writing 0000_0000b - No effect 0000_0001b - Clear the flag

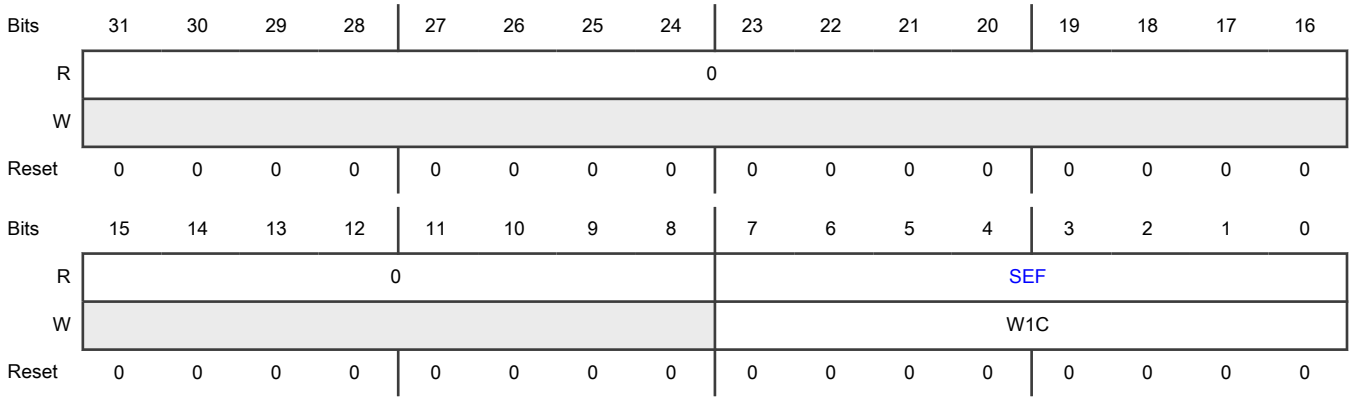
67.7.1.7 Shifter Error (SHIFTERR)

Offset

Register	Offset
SHIFTERR	14h

Function
Reports shifter errors.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SEF	Shifter Error Flag Indicates shifter error flag status. This flag is set when one of the following events occurs: • If SHIFTCTL*n*[SMOD] = 001b (Receive mode), it indicates that either the shifter is ready to store new data into SHIFTBUF before the previous data is read from SHIFTBUF (SHIFTBUF overrun), or the received start or stop bit does not match the expected value. • If SHIFTCTL*n*[SMOD] = 010b (Transmit mode), it indicates that the shifter is ready to load new data from SHIFTBUF before new data is written into SHIFTBUF (SHIFTBUF underrun). • If SHIFTCTL*n*[SMOD] = 100b (Match Store mode), it indicates the occurrence of a match event before the previous match data is read from SHIFTBUF (SHIFTBUF overrun). • If SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the error flag is set when a match occurs between SHIFTBUF and the shifter. • If SHIFTCTL*n*[SMOD] = 111b (Logic mode), the error flag is set when the output of the programmable logic block is asserted. For SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the flag can also be cleared when you read Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000b - Clear 0000_0001b - Set When writing 0000_0000b - No effect 0000_0001b - Clear the flag

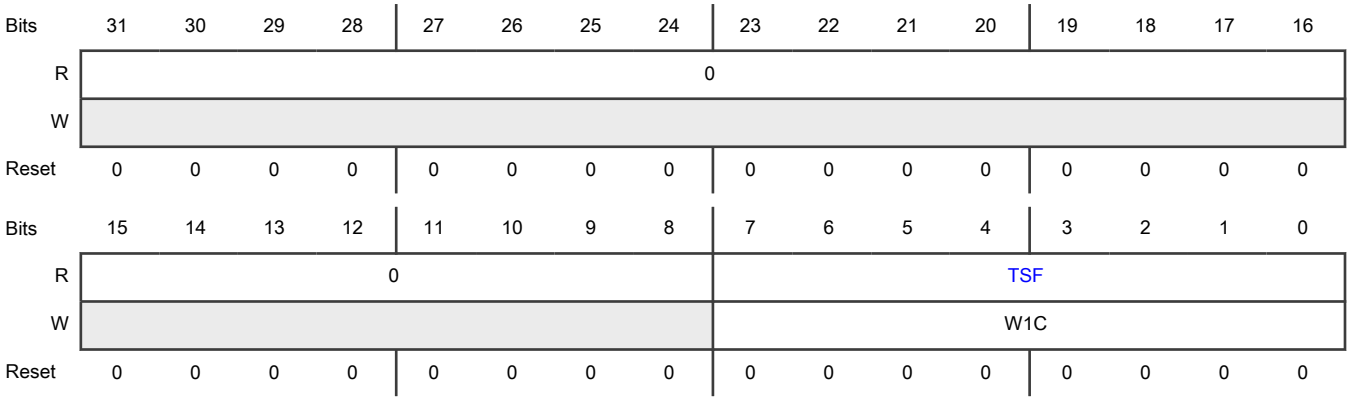
67.7.1.8 Timer Status Flag (TIMSTAT)

Offset

Register	Offset
TIMSTAT	18h

Function
Reports timer status.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 TSF	Timer Status Flag Indicates timer status. This flag is set depending on Timer mode: • In 8-bit baud counter mode, this flag is set when the upper 8-bit counter equals zero and decrements. • In 8-bit high PWM mode, this flag is set when the upper 8-bit counter equals zero and decrements. • In 16-bit counter mode, this flag is set when the 16-bit counter equals zero and decrements. • In 16-bit counter disable mode, TSF is set when a timer disable event is detected. • In 8-bit word counter mode, TSF is set when the upper 8-bit counter equals zero and decrements. • In 8-bit low PWM mode, TSF is set when the upper 8-bit counter equals zero and decrements. • In 16-bit input capture mode, TSF is set when a timer disable event is detected and the flag is clear. In this mode, you must read Timer Control (TIMCTL0 - TIMCTL7) only when TSF is set. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When reading 0000_0000b - Clear 0000_0001b - Set When writing 0000_0000b - No effect 0000_0001b - Clear the flag

67.7.1.9 Shifter Status Interrupt Enable (SHIFTSIEN)

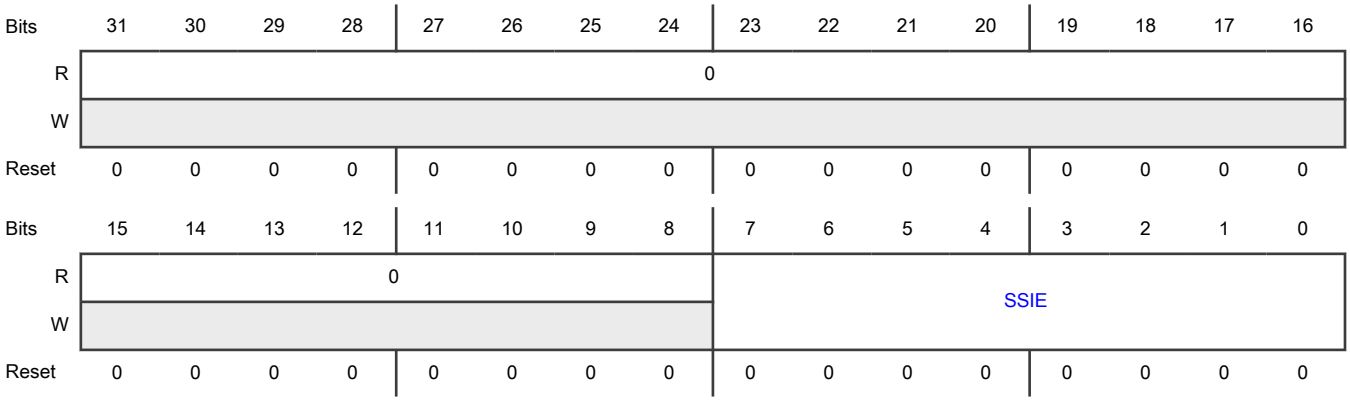
Offset

Register	Offset
SHIFTSIEN	20h

Function

Enables shifter status interrupts.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SSIE	Shifter Status Interrupt Enable Enables interrupt generation when the corresponding SHIFTSTAT[SSF] flag is set. If you write 0 to this field, SHIFTSTAT[SSF] is disabled; and if you write 1 to this field, SHIFTSTAT[SSF] is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable

67.7.1.10 Shifter Error Interrupt Enable (SHIFTEIEN)

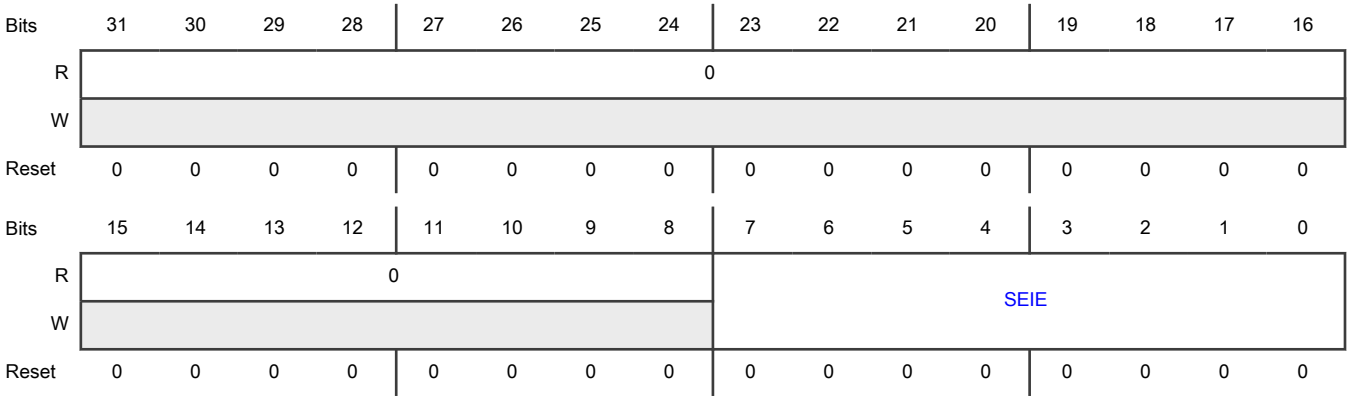
Offset

Register	Offset
SHIFTEIEN	24h

Function

Enables shifter error interrupts.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SEIE	Shifter Error Interrupt Enable Enables interrupt generation when the corresponding SHIFTERR[SEF] flag is set. If you write 0 to this field, SHIFTERR[SEF] is disabled; and if you write 1 to this field, SHIFTERR[SEF] is enabled. 0b - Disable 1b - Enable

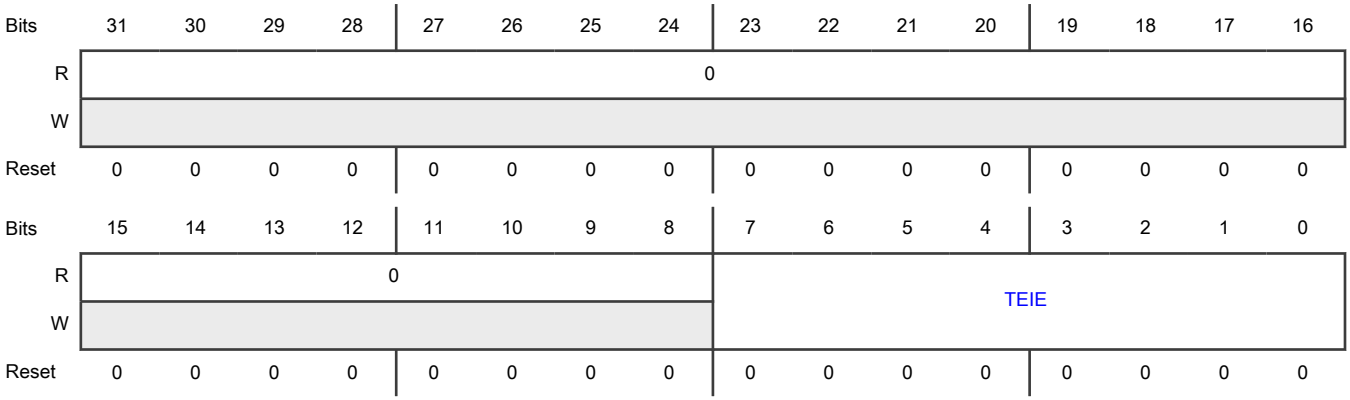
67.7.1.11 Timer Interrupt Enable (TIMIEN)

Offset

Register	Offset
TIMIEN	28h

Function
Enables timer status interrupts.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 TEIE	Timer Status Interrupt Enable Enables interrupt generation when the corresponding TIMSTAT[TSF] flag is set. If you write 0 to this field, TIMSTAT[TSF] is disabled; and if you write 1 to this field, TIMSTAT[TSF] is enabled. 0b - Disable 1b - Enable

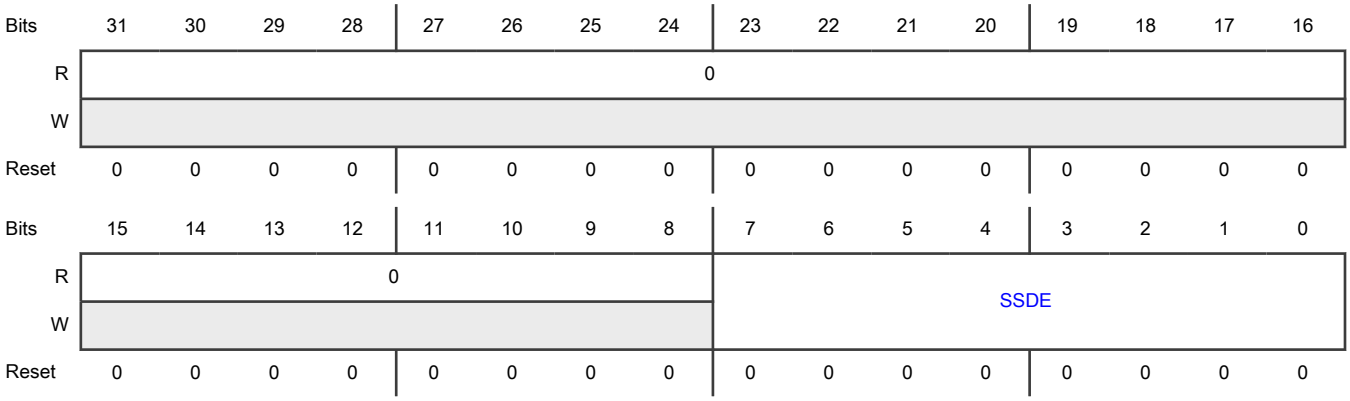
67.7.1.12 Shifter Status DMA Enable (SHIFTSDEN)

Offset

Register	Offset
SHIFTSDEN	30h

Function
Enables shifter DMA requests.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 SSDE	Shifter Status DMA Enable Enables DMA request generation when the corresponding SHIFTSTAT[SSF] flag is set. If you write 0 to this field, SHIFTSTAT[SSF] is disabled; and if you write 1 to this field, SHIFTSTAT[SSF] is enabled. 0b - Disable 1b - Enable

67.7.1.13 Timer Status DMA Enable (TIMERSDEN)

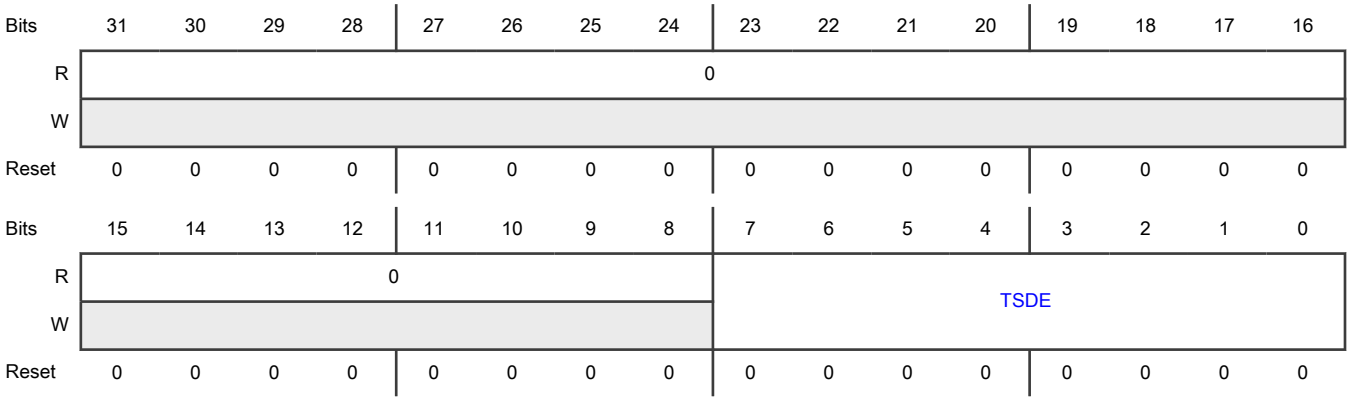
Offset

Register	Offset
TIMERSDEN	38h

Function

Enables DMA requests when the timer status flag is set.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 TSDE	Timer Status DMA Enable Enables DMA request generation when the corresponding TIMSTAT[TSF] flag is set. When the timer status DMA request is enabled, reading or writing to a timer compare register clears the corresponding timer status register. The DMA must therefore read or write to the timer compare register as part of the DMA transfer; otherwise, the DMA request remains asserted. 0b - Disable 1b - Enable

67.7.1.14 Shifter State (SHIFTSTATE)

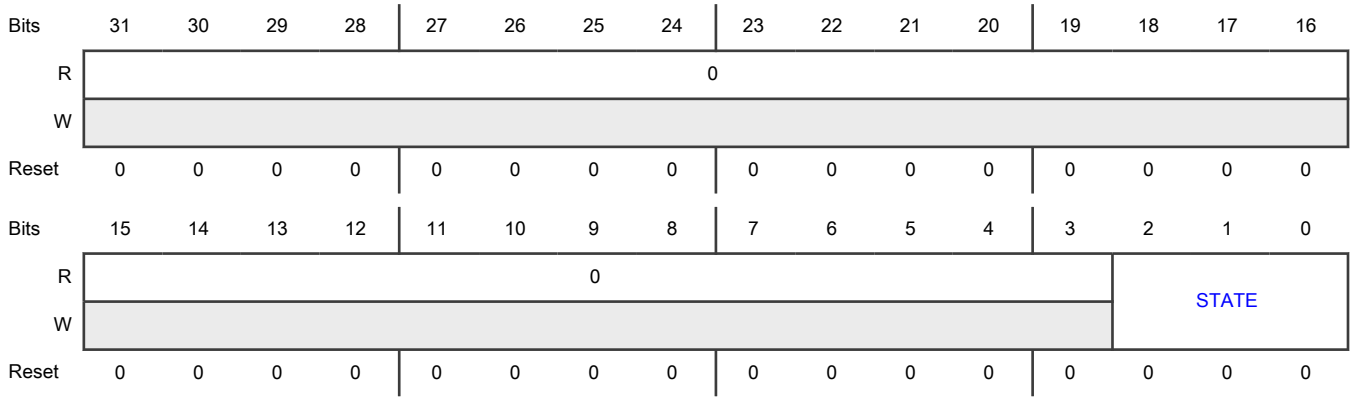
Offset

Register	Offset
SHIFTSTATE	40h

Function

Contains a pointer to track the current shifter.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 STATE	Current State Pointer Maintains a pointer to track the current shifter (configured for State mode) enabled to drive outputs and compute the next state. Reading this register when the state pointer is updating can result in the return of an incorrect state. The value that you write to this field overrides the current state.

67.7.1.15 Trigger Status (TRGSTAT)

Offset

Register	Offset
TRGSTAT	48h

Function

Contains external trigger status flags.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ETSF			
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3-0 ETSF	External Trigger Status Flag Specifies whether the external trigger status flag is set when a rising edge is detected on the corresponding external trigger input. **NOTE** This field behaves differently for register reads and writes. When reading 0000b - Clear 0001b - Set When writing 0000b - No effect 0001b - Clear the flag

67.7.1.16 External Trigger Interrupt Enable (TRIGIEN)

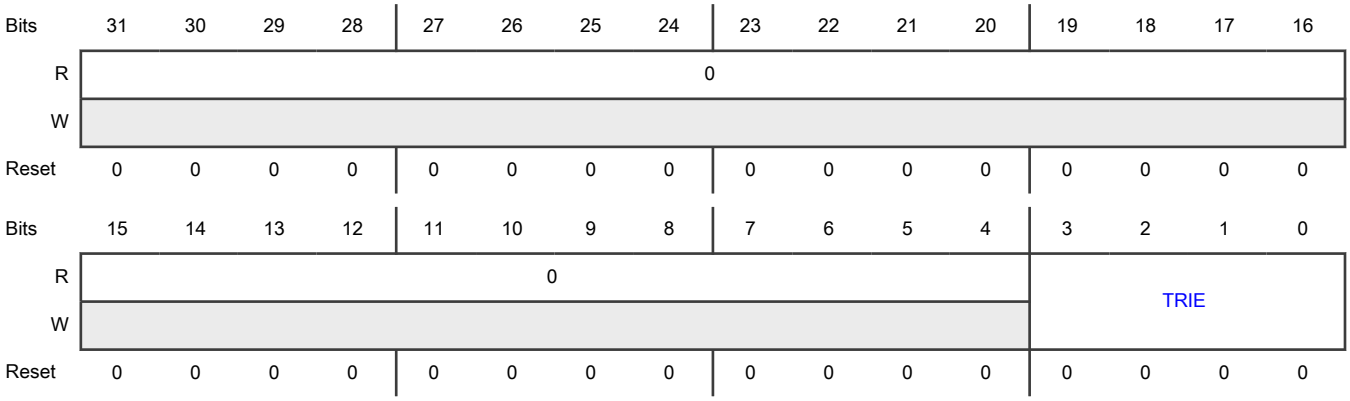
Offset

Register	Offset
TRIGIEN	4Ch

Function

Enables external trigger interrupts.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 TRIE	External Trigger Interrupt Enable Enables interrupt generation when the corresponding TRGSTAT[ETSF] flag is set. If you write 0 to this field, TRGSTAT[ETSF] is disabled, and if you write 1 to this field, TRGSTAT[ETSF] is enabled. 0b - Disable 1b - Enable

67.7.1.17 Pin Status (PINSTAT)

Offset

Register	Offset
PINSTAT	50h

Function

Contains pin status flags.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PSF															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 PSF	Pin Status Flag Indicates whether the pin status flag is set when a rising edge or falling edge (if configured) is detected on the corresponding pin, as configured by the pin. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000_0000_0000b - Clear 0000_0000_0000_0001b - Set When writing 0000_0000_0000_0000b - No effect 0000_0000_0000_0001b - Clear the flag

67.7.1.18 Pin Interrupt Enable (PINIEN)

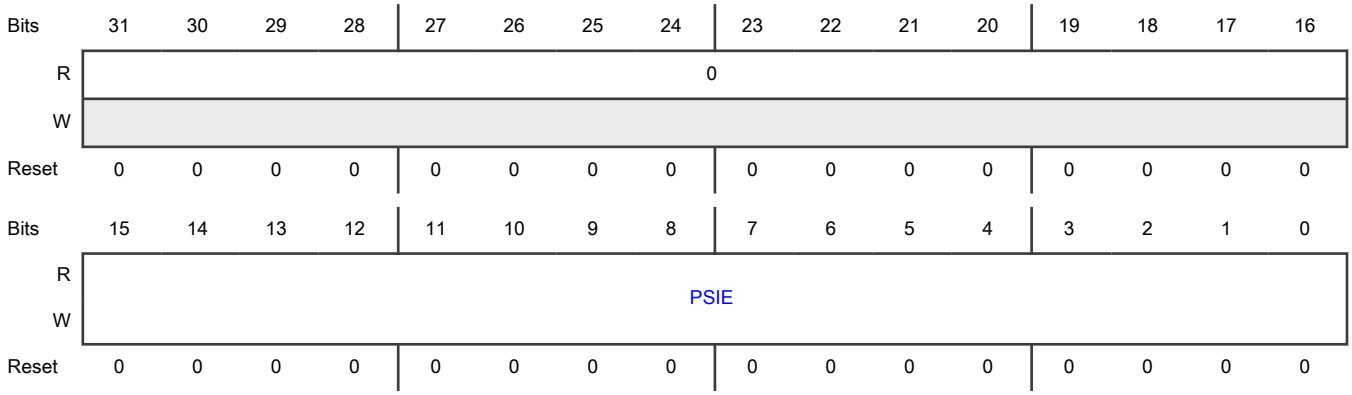
Offset

Register	Offset
PINIEN	54h

Function

Enables pin status interrupts.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PSIE	Pin Status Interrupt Enable Enables interrupt generation when the corresponding PINSTAT[PSF] flag is set. If you write 0 to this field, PINSTAT[PSF] is disabled, and if you write 1 to this field, PINSTAT[PSF] is enabled. 0b - Disable 1b - Enable

67.7.1.19 Pin Rising Edge Enable (PINREN)

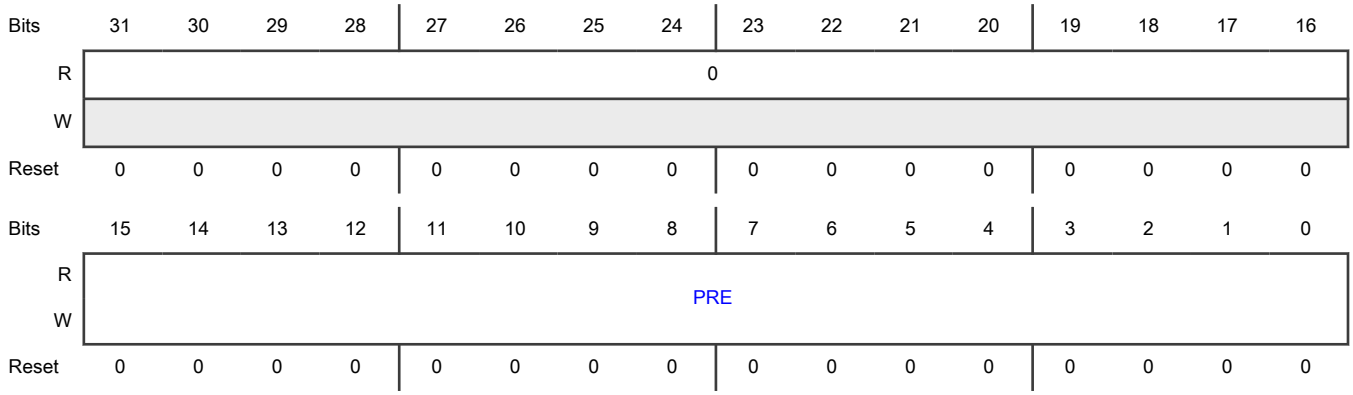
Offset

Register	Offset
PINREN	58h

Function

Enables the pin status flag on a rising edge.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PRE	Pin Rising Edge Specifies whether the pin status flag is set whenever a rising edge is detected on the pin. 0b - Not set 1b - Set

67.7.1.20 Pin Falling Edge Enable (PINFEN)

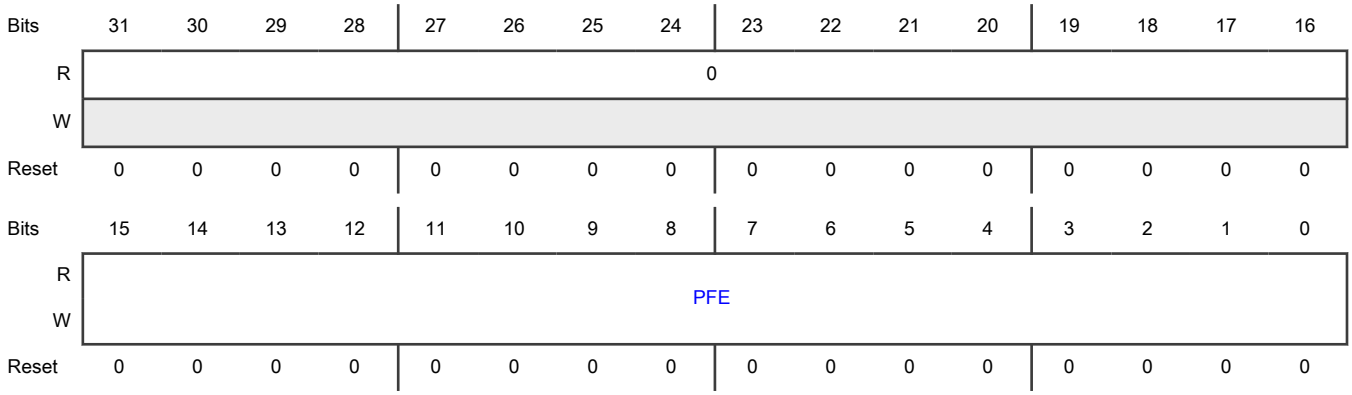
Offset

Register	Offset
PINFEN	5Ch

Function

Enables the pin status flag on a falling edge.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 PFE	Pin Falling Edge Specifies whether the pin status flag is set whenever a falling edge is detected on the pin. 0b - Not set 1b - Set

67.7.1.21 Pin Output Data (PINOUTD)

Offset

Register	Offset
PINOUTD	60h

Function

Contains data output when direct pin output is enabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 OUTD	Output Data Configures the value driven on the corresponding pin when direct pin output is enabled. 0b - Logic zero 1b - Logic one

67.7.1.22 Pin Output Enable (PINOUTE)

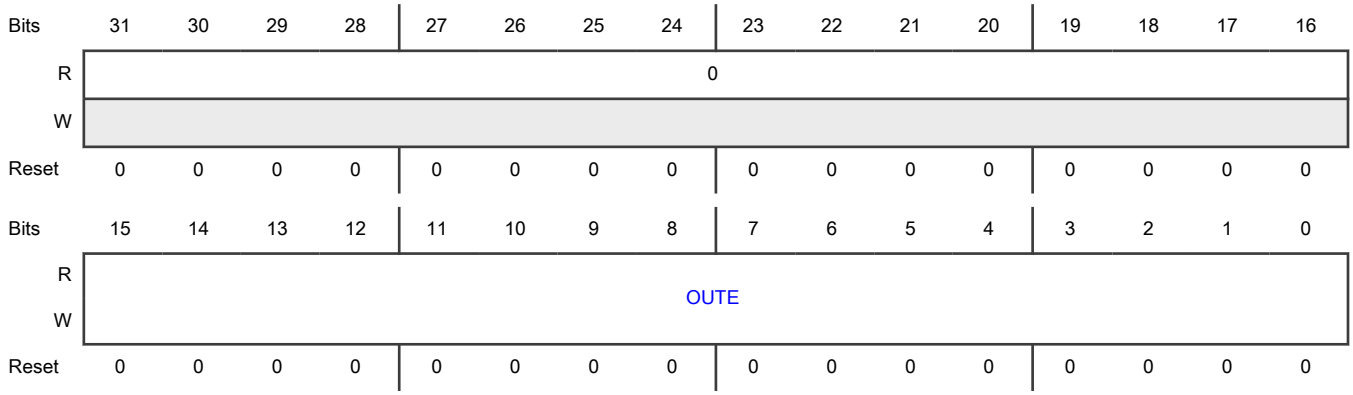
Offset

Register	Offset
PINOUTE	64h

Function

Enables pin output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUTE	Output Enable Enables direct output on the corresponding pin. If this field is 0, the pin is controlled by timer/shifter configuration, and if this field is 1, pin is an output and driven with the value of Pin Output Data (PINOUTD). 0b - Controlled by timer/shifter configuration 1b - Output; driven with value of PINOUTD

67.7.1.23 Pin Output Disable (PINOUTDIS)

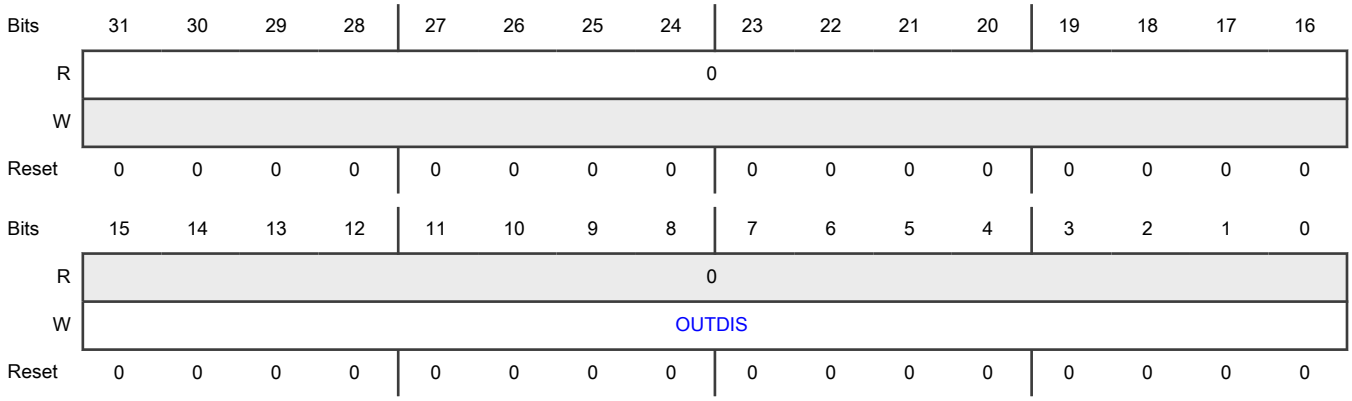
Offset

Register	Offset
PINOUTDIS	68h

Function

Disables pin output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUTDIS	Output Disable Configures the corresponding pins to disable direct output. If this field is 1, the corresponding fields in Pin Output Data (PINOUTD) and Pin Output Enable (PINOUTE) become 0. 0b - No effect 1b - Corresponding fields become 0

67.7.1.24 Pin Output Clear (PINOUTCLR)

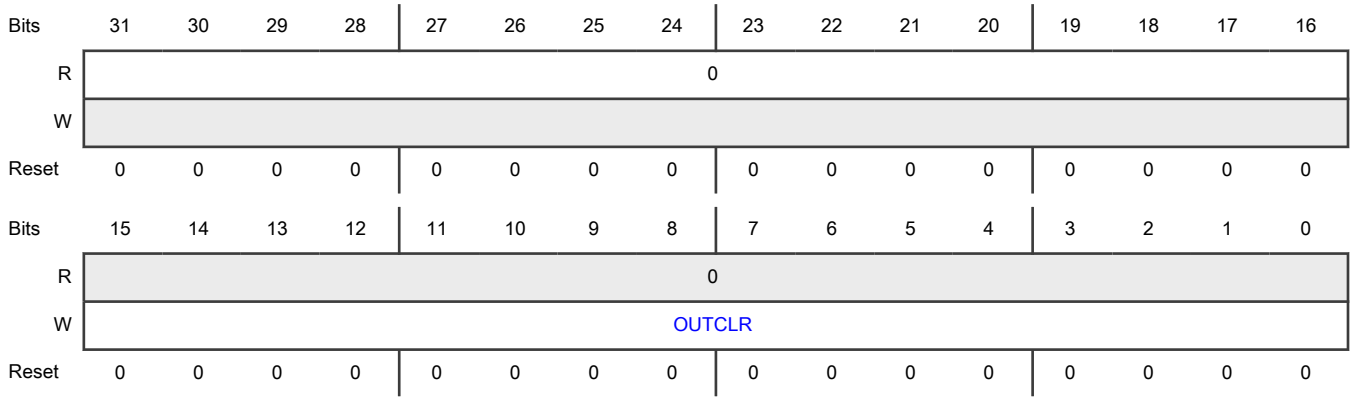
Offset

Register	Offset
PINOUTCLR	6Ch

Function

Clears pin output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUTCLR	Output Clear Configures the corresponding pins to output zero. If this field is 1, the corresponding field in Pin Output Data (PINOUTD) becomes 0 and the one in Pin Output Enable (PINOUTE) becomes 1. 0b - No effect 1b - Corresponding field in PINOUTD becomes 0; corresponding field in PINOUTE becomes 1

67.7.1.25 Pin Output Set (PINOUTSET)

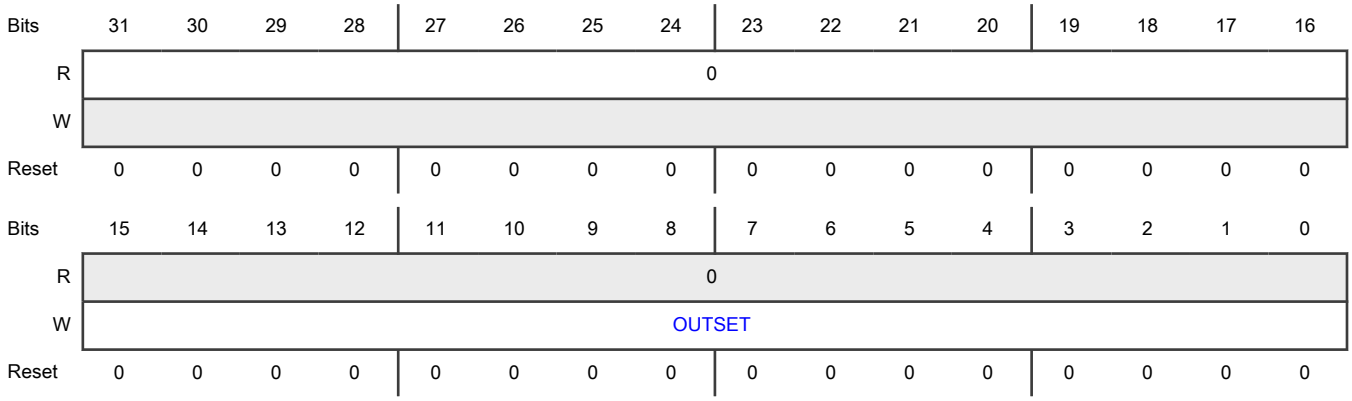
Offset

Register	Offset
PINOUTSET	70h

Function

Sets pin output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUTSET	Output Set Configures the corresponding pins to output logic one. If this field is 1, the corresponding fields in Pin Output Data (PINOUTD) and Pin Output Enable (PINOUTE) become 1. 0b - No effect 1b - Corresponding fields become 1

67.7.1.26 Pin Output Toggle (PINOUTTOG)

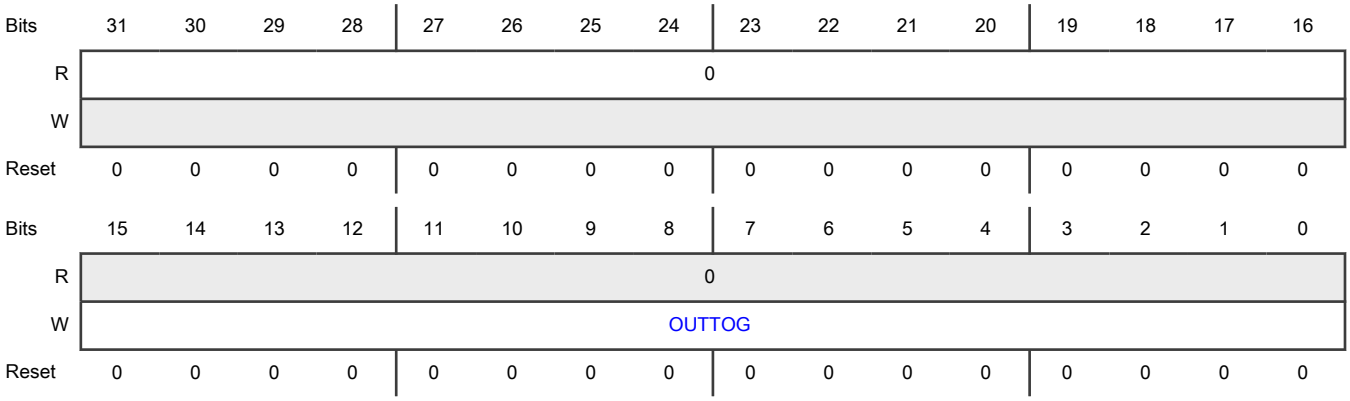
Offset

Register	Offset
PINOUTTOG	74h

Function

Toggles pin output.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 OUTTOG	Output Toggle Configures the corresponding pins to toggle. If this field is 1, the corresponding field in Pin Output Data (PINOUTD) is inverted and the one in Pin Output Enable (PINOUTE) becomes 1. 0b - No effect 1b - Corresponding field in PINOUTD is inverted; corresponding field in PINOUTE becomes 1

67.7.1.27 Shifter Control (SHIFTCTL0 - SHIFTCTL7)

Offset

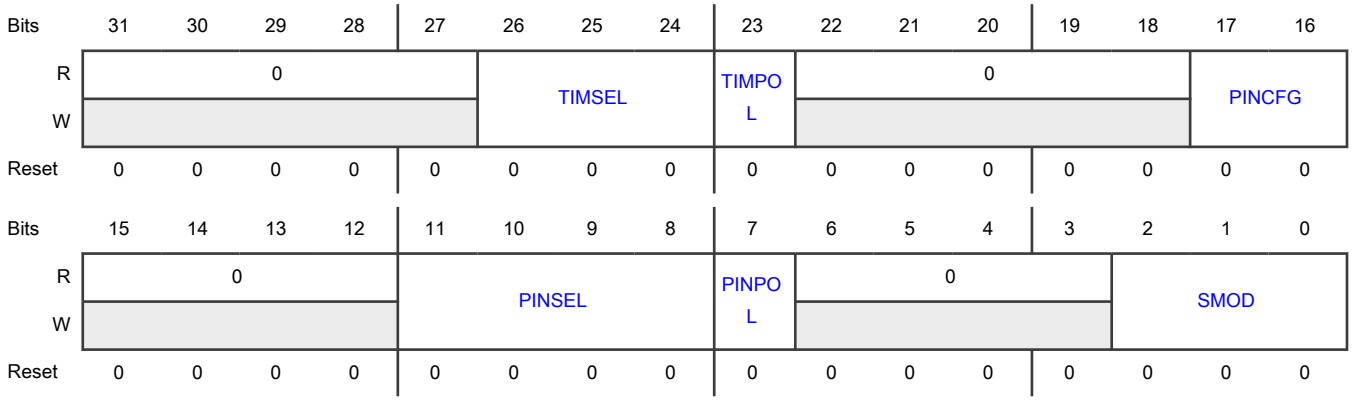
For n = 0 to 7:

Register	Offset
SHIFTCTLn	80h + (n × 4h)

Function

Provides shifter controls.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-24 TIMSEL	Timer Select Selects which timer is used for controlling the logic or shift register and generating the shift clock. TIMSEL = i selects TIMERi.
23 TIMPOL	Timer Polarity Determines whether the shift occurs on the positive edge or negative edge of the shift clock. 0b - Positive edge 1b - Negative edge
22-18 —	Reserved
17-16 PINCFG	Shifter Pin Configuration Specifies shifter pin configuration. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. **NOTE** When initially configuring PINCFG as 11b, FLEXIO may briefly drive the pin low. To avoid this, you can configure PINCFG as 10b along with the rest of the Control register and then perform a subsequent write to set PINCFG as 11b. Likewise, when changing the value of PINCFG from 11b to 00b, you must perform an initial write to set PINCFG as 10b and then perform a subsequent write to update the rest of the Control register with the value of PINCFG as 00b. 00b - Shifter pin output disabled 01b - Shifter pin open-drain or bidirectional output enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Shifter pin bidirectional output data 11b - Shifter pin output
15-12 —	Reserved
11-8 PINSEL	Shifter Pin Select Selects the pin that is used by the shifter input or output. PINSEL = i selects the FXIO_Di pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register.
7 PINPOL	Shifter Pin Polarity Specifies the shifter pin polarity. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to this register. 0b - Active high 1b - Active low
6-3 —	Reserved
2-0 SMOD	Shifter Mode Configures the mode of the shifter. 000b - Disable 001b - Receive mode; capture the current shifter content into SHIFTBUF on expiration of the timer 010b - Transmit mode; load SHIFTBUF contents into the shifter on expiration of the timer 011b - Reserved 100b - Match Store mode; shifter data is compared to SHIFTBUF content on expiration of the timer 101b - Match Continuous mode; shifter data is continuously compared to SHIFTBUF contents 110b - State mode; SHIFTBUF contents store programmable state attributes 111b - Logic mode; SHIFTBUF contents implement programmable logic lookup table

67.7.1.28 Shifter Configuration (SHIFTCFG0 - SHIFTCFG7)

Offset

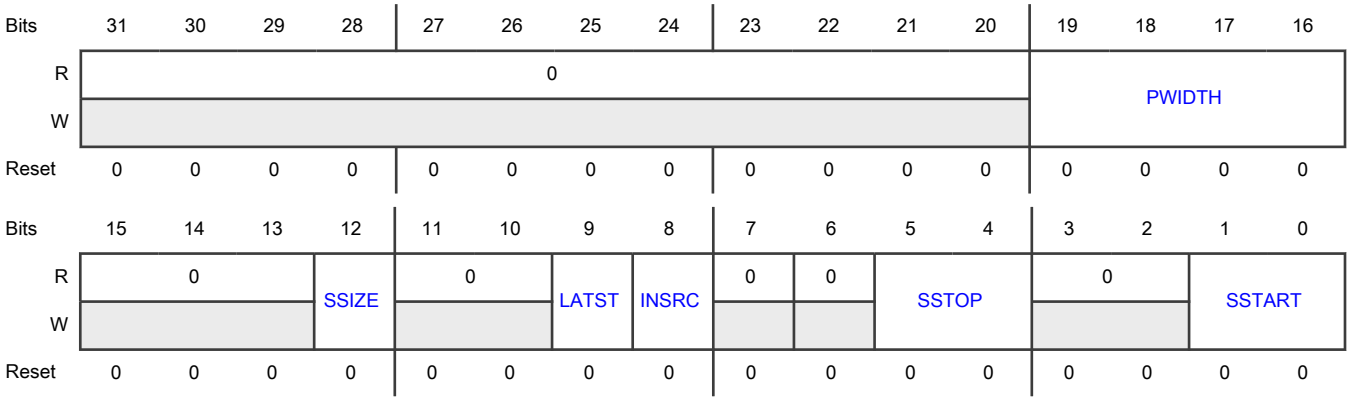
For n = 0 to 7:

Register	Offset
SHIFTCFGn	100h + (n × 4h)

Function

Provides fields for shifter configuration.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-16 PWIDTH	Parallel Width Configures the number of bits to be shifted on each shift clock for all shifters: • 1-bit shift for PWIDTH = 0 • 2-bit shift for PWIDTH = 1 • 4-bit shift for PWIDTH = 2...3 • 8-bit shift for PWIDTH = 4...7 • 16-bit shift for PWIDTH = 8...15 For shifters that support parallel transmit (SHIFTER0, SHIFTER4, ...) or parallel receive (SHIFTER3, SHIFTER7, ...), this field, together with PINSEL, also selects the pins to be driven or sampled on each shift clock: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL]. Shifters that do not support parallel transmit or parallel receive only support parallel shift when SHIFTCFG*n*[INSRC] = 1. If SHIFTCTL*n*[SMOD] = 110b (State mode), use this field to disable state outputs (see State mode).
15-13 —	Reserved
12 SSIZE	Shifter Size Configures the size of the Shift registers. A 24-bit Shift register shifts data only into bits [23:0] and does not update bits [31:24] during shift operations.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When the Shift register is configured for a 24-bit shift, configuring PWIDTH as 8..15 performs a 12-bit shift and PWIDTH as 16..31 performs a 24-bit shift. 0b - 32-bit 1b - 24-bit
11-10 —	Reserved
9 LATST	Late Store Configures what happens when a receive or match Shift register is configured to both shift and store on the same cycle. 0b - Store the pre-shift register state 1b - Store the post-shift register state
8 INSRC	Input Source Selects the input source for the shifter. Configuring this field as 1 is not supported for the last shifter. 0b - Pin 1b - Shifter n+1 output
7 —	Reserved
6 —	Reserved
5-4 SSTOP	Shifter Stop Allows automatic stop bit insertion, if the selected timer has also enabled a stop bit, when SHIFTCTL*n*[SMOD] is 10b (Transmit mode). If SHIFTCTL*n*[SMOD] is 1b or 100b (Receive mode or Match Store mode), this field allows automatic stop bit checking if the selected timer has also enabled a stop bit. If SHIFTCTL*n*[SMOD] is 110b (State mode), this field disables state outputs (see State mode). If SHIFTCTL*n*[SMOD] is 111b (Logic mode), this field masks logic pin inputs (see Logic mode). 00b - Stop bit disabled for Transmitter, Receiver, and Match Store modes 01b - Stop bit disabled for Transmitter, Receiver, and Match Store modes; when timer is in stop condition, Receiver and Match Store modes store receive data on the configured shift edge 10b - Transmitter mode outputs stop bit value 0 in Match Store mode; if stop bit is not 0, Receiver and Match Store modes set error flag (when timer is in stop condition, these modes also store receive data on the configured shift edge)

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Transmitter mode outputs stop bit value 1 in Match Store mode; if stop bit is not 1, Receiver and Match Store modes set error flag (when timer is in stop condition, these modes also store receive data on the configured shift edge)
3-2 ——	Reserved
1-0 SSTART	Shifter Start Allows automatic start bit insertion, if the selected timer has also enabled a start bit, when SHIFTCTL*n*[SMOD] is 10b (Transmit mode). If SHIFTCTL*n*[SMOD] = 1b (Receive mode) or 100b (Match Store mode), this field allows automatic start bit checking if the selected timer has also enabled a start bit. If SHIFTCTL*n*[SMOD] is 110b (State mode), this field disables state outputs (see State mode). If SHIFTCTL*n*[SMOD] = 111b (Logic mode), this field masks logic pin inputs (see Logic mode). 00b - Start bit disabled for Transmitter, Receiver, and Match Store modes; Transmitter mode loads data on enable 01b - Start bit disabled for Transmitter, Receiver, and Match Store modes; Transmitter mode loads data on first shift 10b - Transmitter mode outputs start bit value 0 before loading data on first shift; if start bit is not 0, Receiver and Match Store modes set error flag 11b - Transmitter mode outputs start bit value 1 before loading data on first shift; if start bit is not 1, Receiver and Match Store modes set error flag

67.7.1.29 Shifter Buffer (SHIFTBUF0 - SHIFTBUF7)

Offset

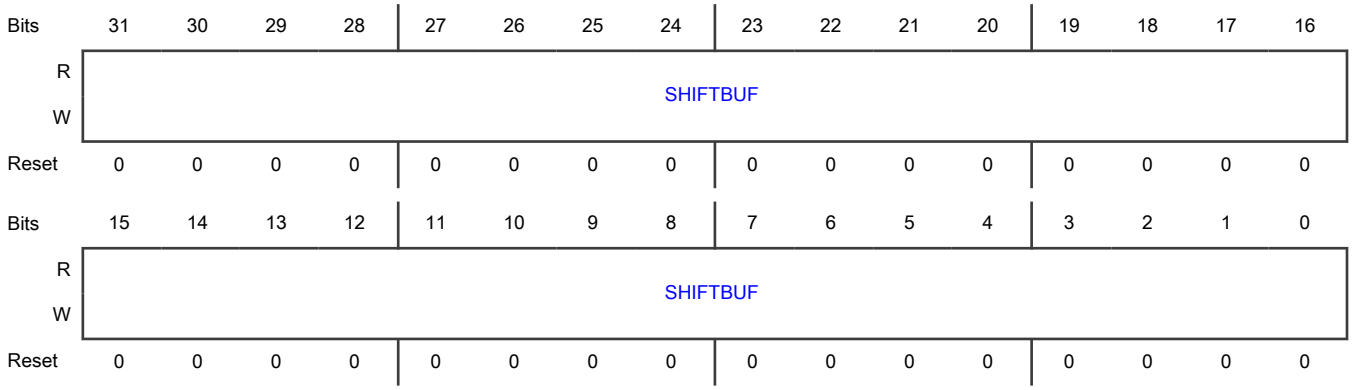
For n = 0 to 7:

Register	Offset
SHIFTBUFn	200h + (n × 4h)

Function

Contains shift buffer data.

Diagram



Fields

Field	Function
31-0 SHIFTBUF	Shift Buffer Contains the data to be matched with the shifter contents and is used for various other functions, depending on the setting of SHIFTCTL*n*[SMOD]: • If SHIFTCTL0[SMOD] is 1b (Receive mode), shifter data is transferred into SHIFTBUF at the expiration of the timer. You must read this register only when the corresponding SHIFTSTAT[SSF] flag is set, indicating that new shifter data is available. • If SHIFTCTL0[SMOD] is 10b (Transmit mode), SHIFTBUF data is transferred into the shifter before the timer begins. • If SHIFTCTL0[SMOD] is 100b (Match Store mode), SHIFTBUF[31:16] contains the data to be matched with the shifter contents and SHIFTBUF[15:0] can be used to mask the match result (1 = mask, 0 = no mask). The match is checked when the timer expires. Shifter data [31:16] is written to SHIFTBUF[31:16] whenever a match event occurs. You must read this register only when the corresponding shifter status flag is set, indicating that new shifter data is available. • If SHIFTCTL0[SMOD] is 101b (Match Continuous mode), SHIFTBUF[31:16] contains the data to be matched with the shifter contents, and SHIFTBUF[15:0] can be used to mask the match result (1 = mask, 0 = no mask). • If SHIFTCTL0[SMOD] is 111b (Logic mode), SHIFTBUF[31:0] implements a 5-input, 32-bit programmable logic lookup table (see Logic mode). • If SHIFTCTL0[SMOD] is 110b (State mode), use SHIFTBUF[31:24] to drive the output value when this shifter is selected by the current state pointer and use SHIFTBUF[23:0] to configure the value of the next state transition (see State mode).

67.7.1.30 Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBISn	280h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is bit-swapped.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFBIS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are bit-swapped. Reads return SHIFTBUF[0:31].

67.7.1.31 Shifter Buffer Byte Swapped (SHIFTBUFBYS0 - SHIFTBUFBYS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBYSn	300h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is byte-swapped.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBYS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBYS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFBYS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are byte-swapped. Reads return {SHIFTBUF[7:0], SHIFTBUF[15:8], SHIFTBUF[23:16], SHIFTBUF[31:24]}.

67.7.1.32 Shifter Buffer Bit Byte Swapped (SHIFTBUFBBS0 - SHIFTBUFBBS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBBSn	380h + (n × 4h)

Function

Contains the register data for Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but it is bit-swapped within each byte.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFBBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), except that reads or writes to this register are bit-swapped within each byte. Reads return {SHIFTBUF[24:31], SHIFTBUF[16:23], SHIFTBUF[8:15], SHIFTBUF[0:7]}.

67.7.1.33 Timer Control (TIMCTL0 - TIMCTL7)

Offset

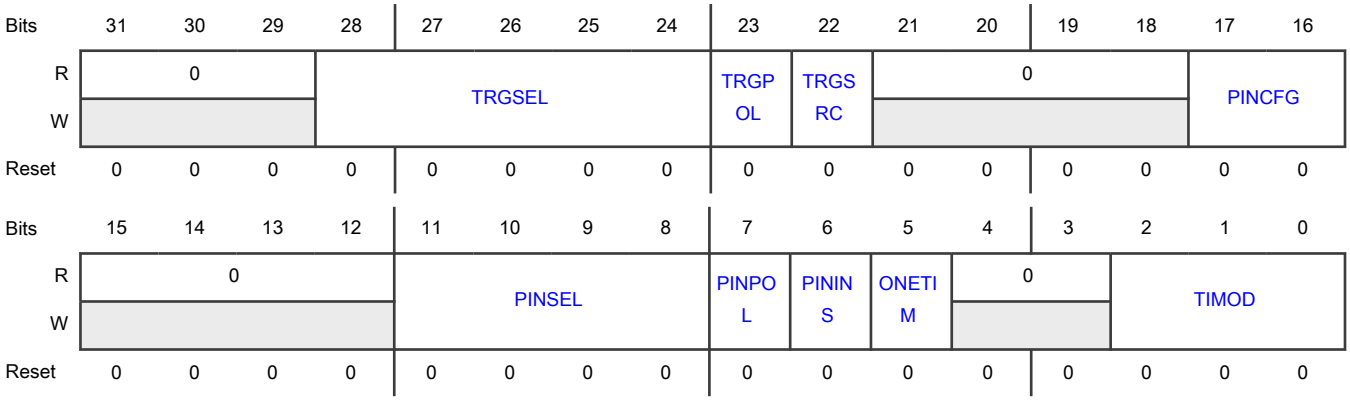
For n = 0 to 7:

Register	Offset
TIMCTLn	400h + (n × 4h)

Function

Controls various settings for timer *n*.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-24 TRGSEL	Trigger Select Selects the trigger. The valid values for TRGSEL depend on the configuration of Parameter (PARAM): • If TRGSRC = 1, the valid values for *n* depend on the settings of PARAM[PIN], PARAM[TIMER], and PARAM[SHIFTER]. • If TRGSRC = 0, the valid values for *n* depend on PARAM[TRIGGER]. See the chip-specific FLEXIO information for external trigger selection. **NOTE** For a pin, *n* = 0 to 15, for a shifter, *n* = 0 to 7, and for a timer, *n* = 0 to 7. If TRGSRC = 0, configure the trigger selection as *n* = external trigger *n* input. If TRGSRC = 1, you can configure the internal trigger to select an input pin as 2×*n* = pin *n* input. If TRGSRC = 1, you can configure the internal trigger to select a shifter or timer signal as: • 4×*n* + 1 = shifter *n* status flag • 4×*n* + 3 = timer *n* trigger output Following are the values for expanded internal trigger selection (TRGSRC = 1):

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• 0000 = Pin 0 • 0001 = Shifter 0 flag • 0010 = Pin 1 • 0011 = Timer 0 trigger • 0100 = Pin 2 • 0101 = Shifter 1 flag • 0110 = Pin 3 • 0111 = Timer 1 trigger • ... • This continues up to pin 15, shifter 7, and timer 7.
23 TRGPOL	Trigger Polarity Specifies whether the trigger is active high or active low. 0b - Active high 1b - Active low
22 TRGSRC	Trigger Source Specifies whether the selected trigger source is external or internal. 0b - External 1b - Internal
21-18 —	Reserved
17-16 PINCFG	Timer Pin Configuration Configures the direction of the timer pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. **NOTE** When you initially configure PINCFG as 11b, FLEXIO may briefly drive the pin low. To avoid this, configure PINCFG as 10b along with the rest of the Control register and then perform a subsequent write to set the value of PINCFG as 11b. Likewise, when changing the value of PINCFG from 11b to 00b, you must perform an initial write to set PINCFG as 10b, and then perform a subsequent write to update the rest of the Control register with PINCFG as 00b. 00b - Timer pin output disabled 01b - Timer pin open-drain or bidirectional output enable 10b - Timer pin bidirectional output data 11b - Timer pin output

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-12 —	Reserved
11-8 PINSEL	Timer Pin Select Selects the pin that is used by the timer input or output. PINSEL = i selects the FXIO_Di pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register.
7 PINPOL	Timer Pin Polarity Specifies the timer pin polarity. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. 0b - Active high 1b - Active low
6 PININS	Timer Pin Input Select Specifies what selects the timer pin input. If this field is 1, the timer input pin is different from the timer output pin. PINSEL must select an even-numbered pin when this field is 1, which means that the output pin is even-numbered and input pin is odd-numbered. 0b - PINSEL selects timer pin input and output 1b - PINSEL + 1 selects the timer pin input; timer pin output remains selected by PINSEL
5 ONETIM	Timer One Time Operation Configures the timer to perform a single enable or disable iteration. Clear the timer status flag for the timer to be enabled again. 0b - Generate the timer enable event as normal 1b - Block the timer enable event unless the timer status flag is clear
4-3 —	Reserved
2-0 TIMOD	Timer Mode Specifies the timer mode: • In 8-bit baud counter mode, the lower 8 bits of the counter and compare register are used to configure the baud rate of the timer shift clock. The upper 8 bits are used to configure the shifter bit count. • In 8-bit PWM high mode, the lower 8 bits of the counter and compare register are used to configure the high period of the timer shift clock. The upper 8 bits are used to configure the low period of the timer shift clock. The shifter bit count is configured using another timer or external signal. • In 16-bit counter mode, the full 16 bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. • In 16-bit counter disable mode, the full 16 bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• In 8-bit word counter mode, the lower 8 bits of the counter and compare register are used to configure the shifter bit count. The upper 8 bits are used to configure the shifter word count. • In 8-bit PWM low mode, the lower 8 bits of the counter and compare register are used to configure the low period of the timer shift clock. The upper 8 bits are used to configure the high period of the timer shift clock. Use another timer or external signal to configure the shifter bit count. • In 16-bit input capture mode, the inverted value of the 16-bit counter is latched into the compare register when a timer counter disable condition is detected (as configured by TIMCFG*n*[TIMDIS]). This also sets the timer status flag. The timer counter is immediately restarted from FFFFh. 000b - Timer disabled 001b - Dual 8-bit counters baud mode 010b - Dual 8-bit counters PWM high mode 011b - Single 16-bit counter mode 100b - Single 16-bit counter disable mode 101b - Dual 8-bit counters word mode 110b - Dual 8-bit counters PWM low mode 111b - Single 16-bit input capture mode

67.7.1.34 Timer Configuration (TIMCFG0 - TIMCFG7)

Offset

For n = 0 to 7:

Register	Offset
TIMCFGn	480h + (n × 4h)

Function

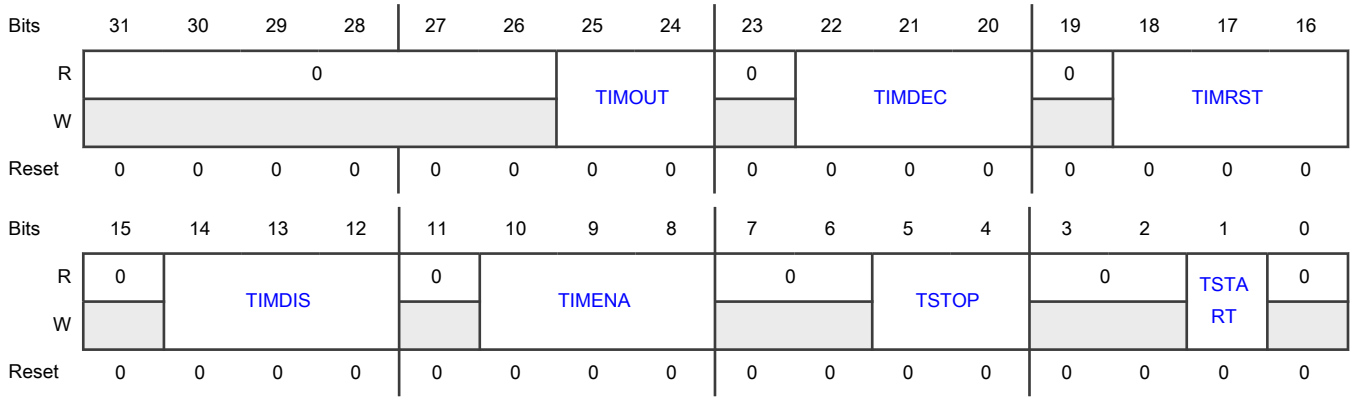
Controls various aspects of timer configuration.

The options to enable or disable the timer using the timer *n* - 1 enable or disable are reserved when *n* is evenly divisible by 4 (timer 0, for example).

NOTE

The pin and trigger level and edges specified in this register refer to the signal state after being modified by the settings of TIMCTL*n*[PINPOL] and TIMCTL*n*[TRGPOL]. For example, "trigger low" means that a trigger is actually at logic level 1 if TIMCTL*n*[TRGPOL] is 1 (active low).

Diagram



Fields

Field	Function
31-26 —	Reserved
25-24 TIMOUT	Timer Output Configures the initial state of the timer output and whether it is affected by the timer reset. 00b - Logic one when enabled; not affected by timer reset 01b - Logic zero when enabled; not affected by timer reset 10b - Logic one when enabled and on timer reset 11b - Logic zero when enabled and on timer reset
23 —	Reserved
22-20 TIMDEC	Timer Decrement Configures the source of the timer decrement and that of the shift clock. 000b - Decrement counter on FLEXIO clock; shift clock equals timer output 001b - Decrement counter on trigger input (both edges); shift clock equals timer output 010b - Decrement counter on pin input (both edges); shift clock equals pin input 011b - Decrement counter on trigger input (both edges); shift clock equals trigger input 100b - Decrement counter on FLEXIO clock divided by 16; shift clock equals timer output 101b - Decrement counter on FLEXIO clock divided by 256; shift clock equals timer output 110b - Decrement counter on pin input (rising edge); shift clock equals pin input 111b - Decrement counter on trigger input (rising edge); shift clock equals trigger input
19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
18-16 TIMRST	Timer Reset Configures the condition that causes the timer counter (and optionally the timer output) to be reset. In 8-bit counter mode, the timer reset only resets the lower 8 bits that configure the baud rate. In all other modes, the timer reset resets full 16 bits of the counter. 000b - Never reset timer 001b - Timer reset on timer output high. 010b - Timer reset on timer pin equal to timer output 011b - Timer reset on timer trigger equal to timer output 100b - Timer reset on timer pin rising edge 101b - Reserved 110b - Timer reset on trigger rising edge 111b - Timer reset on trigger rising or falling edge
15 —	Reserved
14-12 TIMDIS	Timer Disable Configures the condition that causes the timer to be disabled and stop decrementing. 000b - Timer never disabled 001b - Timer disabled on timer n-1 disable 010b - Timer disabled on timer compare (upper 8 bits match and decrement) 011b - Timer disabled on timer compare (upper 8 bits match and decrement) and trigger low 100b - Timer disabled on pin rising or falling edge 101b - Timer disabled on pin rising or falling edge provided trigger is high 110b - Timer disabled on trigger falling edge 111b - Reserved
11 —	Reserved
10-8 TIMENA	Timer Enable Configures the condition that causes the timer to be enabled and start decrementing. 000b - Timer always enabled 001b - Timer enabled on timer n-1 enable 010b - Timer enabled on trigger high 011b - Timer enabled on trigger high and pin high

Table continues on the next page...

Table continued from the previous page...

Field	Function
	100b - Timer enabled on pin rising edge 101b - Timer enabled on pin rising edge and trigger high 110b - Timer enabled on trigger rising edge 111b - Timer enabled on trigger rising or falling edge
7-6 —	Reserved
5-4 TSTOP	Timer Stop Specifies whether the stop bit is enabled. The stop bit can be added on a timer compare (between each word) or on a timer disable. When stop bit is enabled, configured shifters output the contents of the stop bit when the timer is disabled. When stop bit is enabled on timer disable, the timer remains disabled until the next rising edge of the shift clock. If configured for both timer compare and timer disable, only one stop bit is inserted on timer disable. 00b - Disabled 01b - Enabled on timer compare 10b - Enabled on timer disable 11b - Enabled on timer compare and timer disable
3-2 —	Reserved
1 TSTART	Timer Start Specifies whether the start bit is enabled. If it is enabled, configured shifters output the contents of the start bit when the timer is enabled. The timer counter reloads from the compare register on the first rising edge of the shift clock. 0b - Disabled 1b - Enabled
0 —	Reserved

67.7.1.35 Timer Compare (TIMCMP0 - TIMCMP7)

Offset

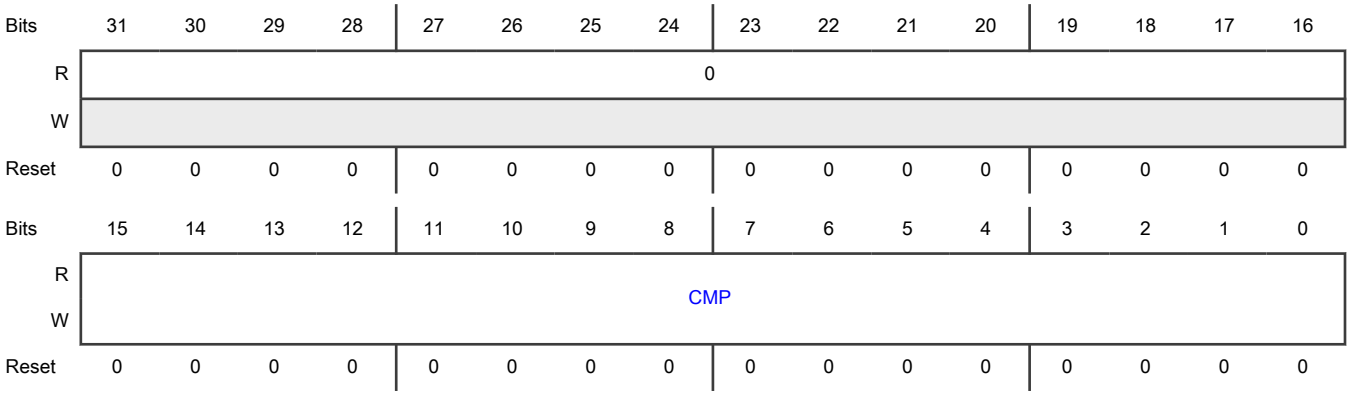
For n = 0 to 7:

Register	Offset
TIMCMPn	500h + (n × 4h)

Function

Contains the timer compare value.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 CMP	Timer Compare Value Loads into the timer counter when the timer is first enabled, when the timer is reset, and when the timer decrements down to zero. In 8-bit baud counter mode, the lower 8 bits configure the baud rate divider as (CMP[7:0] + 1) × 2. The upper 8 bits configure the number of bits in each word as (CMP[15:8] + 1) ÷ 2. In 8-bit PWM high mode, the lower 8 bits configure the high period of the output to (CMP[7:0] + 1) and the upper 8 bits configure the low period of the output to (CMP[15:8] + 1). In 16-bit counter mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) as (CMP[15:0] + 1) × 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word as (CMP[15:0] + 1) ÷ 2. In 16-bit counter disable mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) as (CMP[15:0] + 1) × 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word as (CMP[15:0] + 1) ÷ 2. In 8-bit word counter mode, the lower 8 bits configure the number of bits in each word as (CMP[7:0] + 1) ÷ 2. The upper 8 bits configure the number of words to transfer equal to (CMP[15:8] + 1) ÷ 2. In 8-bit PWM low mode, the lower 8 bits configure the low period of the output to (CMP[7:0] + 1) and the upper 8 bits configure the high period of the output to (CMP[15:8] + 1). In 16-bit input capture mode, the compare register is updated with the inverse of the timer counter value whenever the timer status flag is set. You must read this register only when the timer status flag is set.

67.7.1.36 Shifter Buffer Nibble Byte Swapped (SHIFTBUFNBS0 - SHIFTBUFNBS7)

Offset

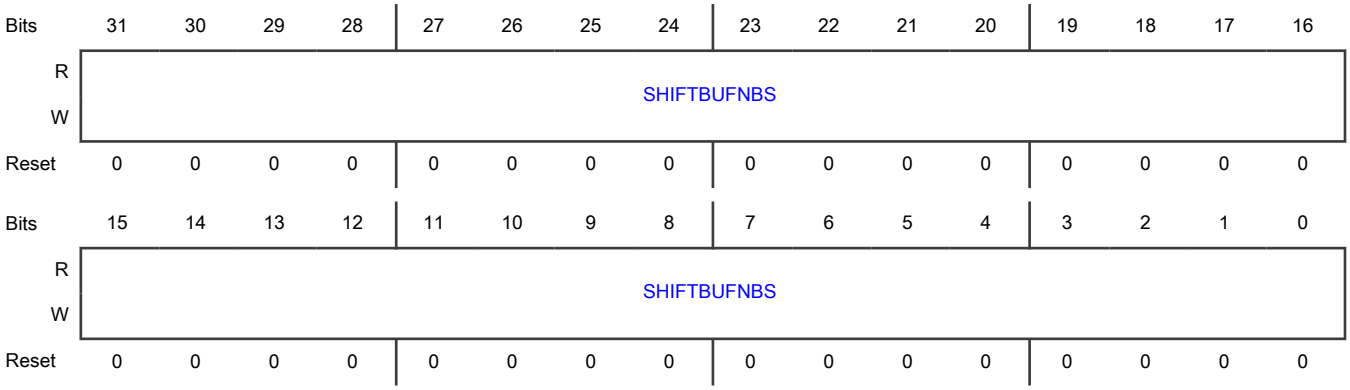
For n = 0 to 7:

Register	Offset
SHIFTBUFNBSn	680h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is nibble-swapped within each byte.

Diagram



Fields

Field	Function
31-0 SHIFTBUFNBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are nibble-swapped within each byte. Reads return {SHIFTBUF[27:24], SHIFTBUF[31:28], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[3:0], SHIFTBUF[7:4]}.

67.7.1.37 Shifter Buffer Halfword Swapped (SHIFTBUFHWS0 - SHIFTBUFHWS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFHWSn	700h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is halfword-swapped.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFHWS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFHWS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFHWS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are halfword-swapped. Reads return {SHIFTBUF[15:0], SHIFTBUF[31:16]}.

67.7.1.38 Shifter Buffer Nibble Swapped (SHIFTBUFNIS0 - SHIFTBUFNIS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFNISn	780h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is nibble-swapped.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFNIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFNIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFNIS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are nibble-swapped. Reads return {SHIFTBUF[3:0], SHIFTBUF[7:4], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[27:24], SHIFTBUF[31:28]}.

67.7.1.39 Shifter Buffer Odd Even Swapped (SHIFTBUFOES0 - SHIFTBUFOES7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFOESn	800h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it has odd and even bits partitioned separately.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFOES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFOES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFOES	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register have the odd and even bits partitioned separately. Only 32-bit accesses are supported for this register. Reads return {SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1], SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0]}.

67.7.1.40 Shifter Buffer Even Odd Swapped (SHIFTBUFEOS0 - SHIFTBUFEOS7)

Offset

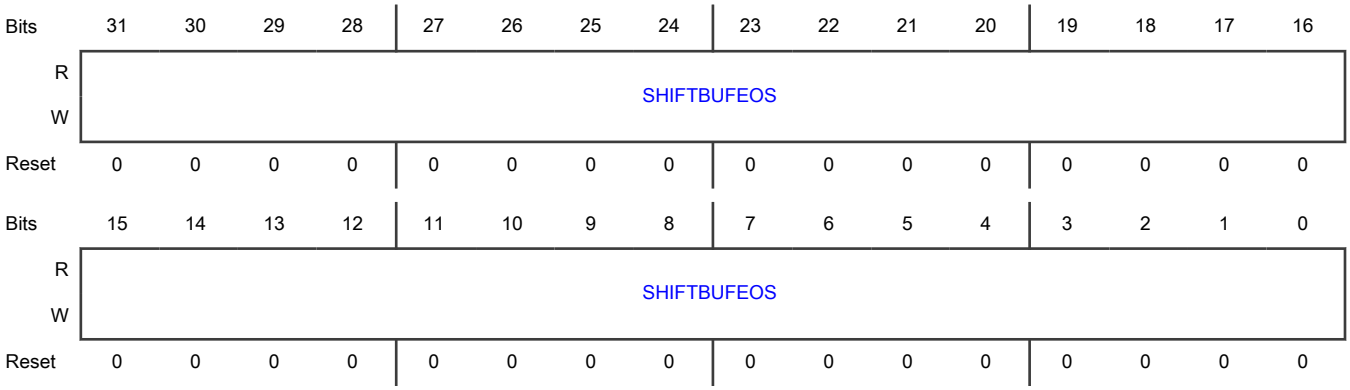
For n = 0 to 7:

Register	Offset
SHIFTBUFEOSn	880h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, with even and odd bits partitioned separately.

Diagram



Fields

Field	Function
31-0 SHIFTBUFEOS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register have the even and odd bits partitioned separately. Only 32-bit accesses are supported for this register. Reads return {SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0], SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1]}.

67.7.1.41 Shifter Buffer Halfword Byte Swapped (SHIFTBUFHBS0 - SHIFTBUFHBS7)

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFHBSn	900h + (n × 4h)

Function

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is halfword byte-swapped.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFHBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFHBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SHIFTBUFHBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are halfword byte-swapped. Reads return {SHIFTBUF[23:16], SHIFTBUF[31:24], SHIFTBUF[7:0], SHIFTBUF[15:8]}.

Chapter 68
Low-Power Flexible Communications Interface (LP_FLEXCOMM)

68.1 Chip-specific LP_FLEXCOMM information

Table 638. Reference links to related information

Topic	Related module	Reference
Full description	LP_FLEXCOMM	LP_FLEXCOMM
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller
Input multiplexing	INPUTMUX	See LP_FLEXCOMMn_TRIG registers in INPUTMUX

68.1.1 Module instances

This device has 21 instances of LP_FLEXCOMM module, numbered LP_FLEXCOMM0 through LP_FLEXCOMM20:

- LP_FLEXCOMM14 and LP_FLEXCOMM16 support only LPSPI functions.
- LP_FLEXCOMM15 supports only LPI2C functions.
- The rest 18 LP_FLEXCOMM instances support LPUART, LPSPI, and LPI2C functions.

Table 639. LP_FLEXCOMM instances

LP_FLEXCOMM number	LPI2C	LPSPI	LPUART	Located in	Access
0	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
1	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
2	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
3	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
4	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

Table continues on the next page...

Table 639. LP_FLEXCOMM instances (continued)

LP_FLEXCOMM number	LPI2C	LPSPI	LPUART	Located in	Access
5	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
6	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
7	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
8	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
9	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
10	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
11	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
12	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
13	Yes	Yes	Yes	VDD2_COMP	CPU0, HiFi4, EZH-V, eDMA0, eDMA1
14	No	Yes	No	VDD2_MEDIA	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
15	Yes	No	No	VDD1_SENSE	CPU0, CPU1, HiFi4, HiFi1
16	No	Yes	No	VDD2_MEDIA	CPU0, CPU1, HiFi4, EZH-V, eDMA0, eDMA1, eDMA2, eDMA3
17	Yes	Yes	Yes	VDD1_SENSE	CPU1, HiFi1, eDMA2, eDMA3
18	Yes	Yes	Yes	VDD1_SENSE	CPU1, HiFi1, eDMA2, eDMA3
19	Yes	Yes	Yes	VDD1_SENSE	CPU1, HiFi1, eDMA2, eDMA3
20	Yes	Yes	Yes	VDD1_SENSE	CPU1, HiFi1, eDMA2, eDMA3

NOTE

Not all LP_FLEXCOMM instance pins are available for each package. You need to configure the module functions according to the package pin availability listed in External signals and pin multiplexing.

Both CPU0 and CPU1 can operate LP_FLEXCOMM14 and LP_FLEXCOMM16. You need to use CPU0 to configure LP_FLEXCOMM14 and LP_FLEXCOMM16 ports by IOPCTL0 registers before either CPU to operate them.

68.1.2 LP_FLEXCOMM configuration

LP_FLEXCOMM supports 8 words FIFO, which is shared between the different configurations.

When configured for LPSPI, LP_FLEXCOMM supports 8x32 bits FIFOs. When configured for LPI2C and LPUART, two 8x16 bits FIFOs are supported, one for each module. In any one of the instances, except for LP_FLEXCOMM14 to LP_FLEXCOMM16, UART and I2C can be used simultaneously with limited functionality. All LP_FLEXCOMM instances, except for LP_FLEXCOMM14 to LP_FLEXCOMM16, support modbus and DMA End of Packet (EOP). There is one interrupt for each instance, see System interrupts. The DMA requests and the trigger inputs are shared.

All the LPI2C instances do not support HS mode when using as controllers.

68.1.3 Register access

For this chip, the LP_FLEXCOMM interface supports only 32-bit register accesses, ignore all the information about supports 8-bit, aligned 16-bit access.

68.1.4 Output trigger

For this chip, all the LP_FLEXCOMM modules have no output triggers, ignore all the information about output triggers in this chapter.

68.1.5 PINT configuration

When an LP_FLEXCOMM is used as LPUART, and its LP_FLEXCOMMx_TRIG[TRIGIN] in INPUTMUX is configured as 1100b to select PINT Boolean pattern-match output as input trigger, the related PMCFG[CFGy] in PINT must not be configured as 111b (Event).

68.2 LP_FLEXCOMM

68.2.1 Overview

Each LP_FLEXCOMM interface provides one peripheral communications function from a choice of several, chosen by software. This chapter describes the overall LP_FLEXCOMM interface and how to choose the communications function. See the corresponding communications module chapter for details on each function provided.

68.2.1.1 Block diagram

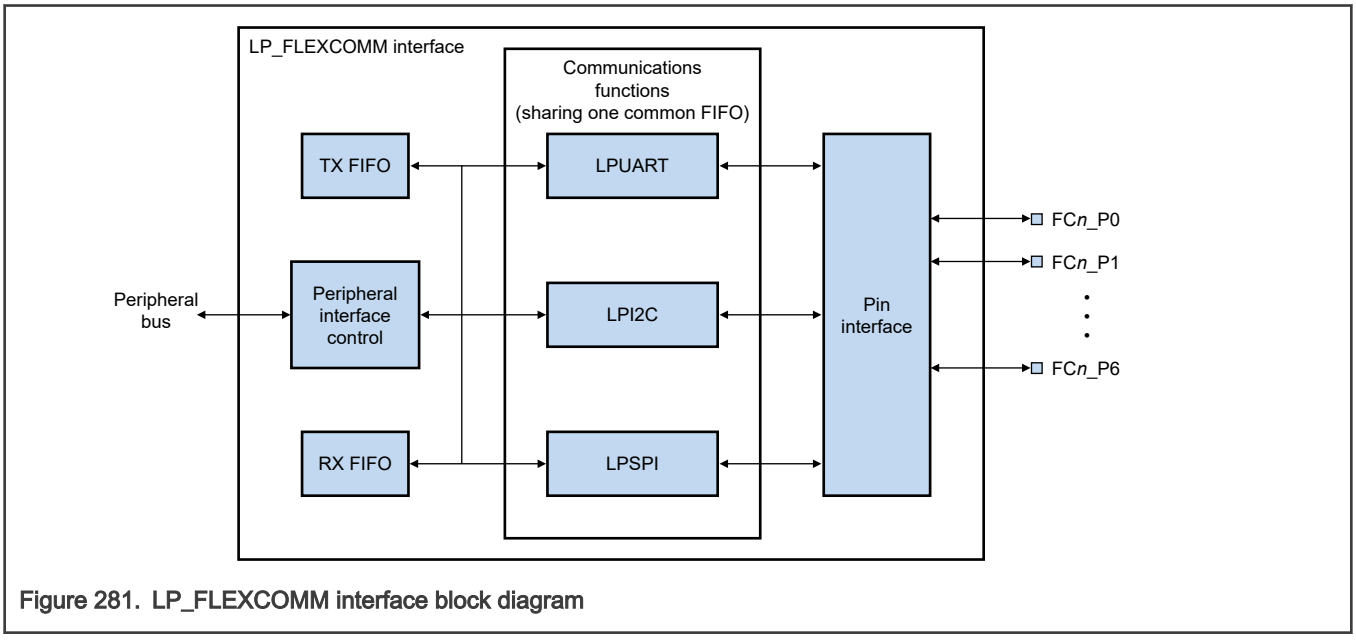


Figure 281. LP_FLEXCOMM interface block diagram

68.2.1.2 Features

Each LP_FLEXCOMM interface provides a choice of peripheral communications functions. You must choose one of them before you can configure and use the function.

- LPUART with asynchronous transmit and receive
- LPSPI controller or subordinate, with up to four peripheral chip selects
- LPI2C with separate controller and subordinate functions

LPUART and LPI2C can be selected at the same time with reduced interface.

A FIFO of 8 words is shared between the different interfaces. The LPI2C subordinate does not support a FIFO.

68.2.2 Functional description

The following sections describe the functional details of LP_FLEXCOMM.

68.2.2.1 Modes

This section describes the modes present in LP_FLEXCOMM.

68.2.2.1.1 Operating modes

Some communication options might not be available on a particular LP_FLEXCOMM interface. Reading PSELID shows which communication functions are available. You can select a specific supported function by writing the appropriate value to PSELID[PERSEL].

68.2.2.1.2 Low-power modes

The LP_FLEXCOMM interface follows the low-power mode operation of the selected communications module.

68.2.2.1.3 Debug mode

When the LP_FLEXCOMM interface is configured as LPSPI or LPI2C, you can configure it not to initiate a new transaction when the CPU is halted.

68.2.2.2 Clocking

The LP_FLEXCOMM interface requires a bus interface clock for the register interface of the selected communications module. It also requires a functional clock which generates the clocking for the peripheral communications interface. The clocks for unselected functions are internally gated.

68.2.2.3 Reset

The LP_FLEXCOMM interface receives a single reset that returns the entire module to its reset state. Out of reset, no peripheral communications function is selected, and the lock bit is clear.

68.2.2.4 Interrupts

The LP_FLEXCOMM interface generates a single interrupt from the configured peripheral communications function(s). When the LP_FLEXCOMM interface is configured for both LPUART and LPI2C functions, you can read the source of the interrupt from ISTAT.

68.2.2.5 DMA

The LP_FLEXCOMM interface generates two DMA requests:

- Transmit DMA Request
- Receive DMA Request (with End Of Packet support)

When the LP_FLEXCOMM interface is configured for both LPUART and LPI2C functions, you must enable DMA requests for only one of them.

68.2.2.6 Triggers

The LP_FLEXCOMM interface receives a single input trigger that is routed to all internal modules. When configured for both LPUART and LPI2C functions, you must enable the input trigger for only one of them.

The LP_FLEXCOMM interface generates 4 output triggers:

Output	LPSPI	LPUART	LPI2C	LPUART+LPI2C
Trigger 0	Frame	TX Word	Controller EOP	Controller EOP
Trigger 1	Word	RX Word	Subordinate EOP	Subordinate EOP
Trigger 2	—	RX Idle	Controller Busy	RX Idle
Trigger 3	—	TX Data	Bus Busy	TX Data

68.2.3 External signals

This table describes the LP_FLEXCOMM interface module signals. Pin usage for a specific communications function is described in the corresponding chapter of each communications module.

Signal	LPSPI	LPUART	LPI2C	LPUART+LPI2C
FC_P0	SDO/DATA[0]	RXD	SDA	SDA
FC_P1	SCK	TXD	SCL	SCL
FC_P2	SDI/DATA[1]	RTS_b	SCLS	TXD
FC_P3	PCS[0]	CTS_b	SDAS	RXD
FC_P4	PCS[3]/DATA[3]	—	HREQ_b	CTS_b
FC_P5	PCS[2]/DATA[2]	—	—	RTS_b
FC_P6	PCS[1]/HREQ	—	—	HREQ_b

68.2.4 Initialization

To initialize a LP_FLEXCOMM interface:

1. Enable the bus interface clock for the LP_FLEXCOMM interface.
2. Select a clock source for the functional clock of the selected communications module.
3. Select the function by writing the appropriate value to PSELID[PERSEL].
4. See the specific communications module chapter to complete the configuration.

68.2.5 Application information

LP_FLEXCOMM interface applications mirror those of the selected communications module.

68.2.6 Memory Map and register definition

This section includes the LP_FLEXCOMM interface module memory map and detailed descriptions of all registers.

The LP_FLEXCOMM interface supports 8-bit, aligned 16-bit, and aligned 32-bit register accesses, unless otherwise noted in the register description of the selected peripheral function.

68.2.6.1 FLEXCOMM register descriptions

68.2.6.1.1 FLEXCOMM memory map

LP_FLEXCOMM0 base address: 4011_0000h

LP_FLEXCOMM1 base address: 4011_1000h

LP_FLEXCOMM2 base address: 4011_2000h

LP_FLEXCOMM3 base address: 4011_3000h

LP_FLEXCOMM4 base address: 4017_1000h

LP_FLEXCOMM5 base address: 4017_2000h

LP_FLEXCOMM6 base address: 4017_3000h

LP_FLEXCOMM7 base address: 4017_4000h

LP_FLEXCOMM8 base address: 4019_9000h

LP_FLEXCOMM9 base address: 4019_A000h

LP_FLEXCOMM10 base address: 4019_B000h

LP_FLEXCOMM11 base address: 4019_C000h

LP_FLEXCOMM12 base address: 4019_D000h

LP_FLEXCOMM13 base address: 4019_E000h

LP_FLEXCOMM17 base address: 4032_6000h

LP_FLEXCOMM18 base address: 4032_7000h

LP_FLEXCOMM19 base address: 4032_8000h

LP_FLEXCOMM20 base address: 4032_9000h

Offset	Register	Width (In bits)	Access	Reset value
FF4h	Interrupt Status (ISTAT)	32	R	0000_0000h
FF8h	Peripheral Select and ID (PSELID)	32	RW	0010_3070h

68.2.6.1.2 Interrupt Status (ISTAT)

Offset

Register	Offset
ISTAT	FF4h

Function

Identifies the source of the LP_FLEXCOMM interface single interrupt.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										I2CS	I2CM	0	SPI	UART RX	UART TX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-6 —	Reserved
5 I2CS	I2C Subordinate Interrupt Indicates if the I2C subordinate has asserted (set) an interrupt or not (remains clear). 0b - Clear 1b - Set
4 I2CM	I2C Controller Interrupt Indicates if the I2C controller has asserted (set) an interrupt or not (remains clear). 0b - Clear 1b - Set
3 —	Reserved
2 SPI	SPI Interrupt Indicates if the SPI has asserted (set) an interrupt or not (remains clear). 0b - Clear 1b - Set
1 UARTRX	UART RX Interrupt Indicates if the UART receiver has asserted (set) an interrupt or not (remains clear).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Clear 1b - Set
0 UARTTX	UART TX Interrupt Indicates if the UART transmitter has asserted (set) an interrupt or not (remains clear). 0b - Clear 1b - Set

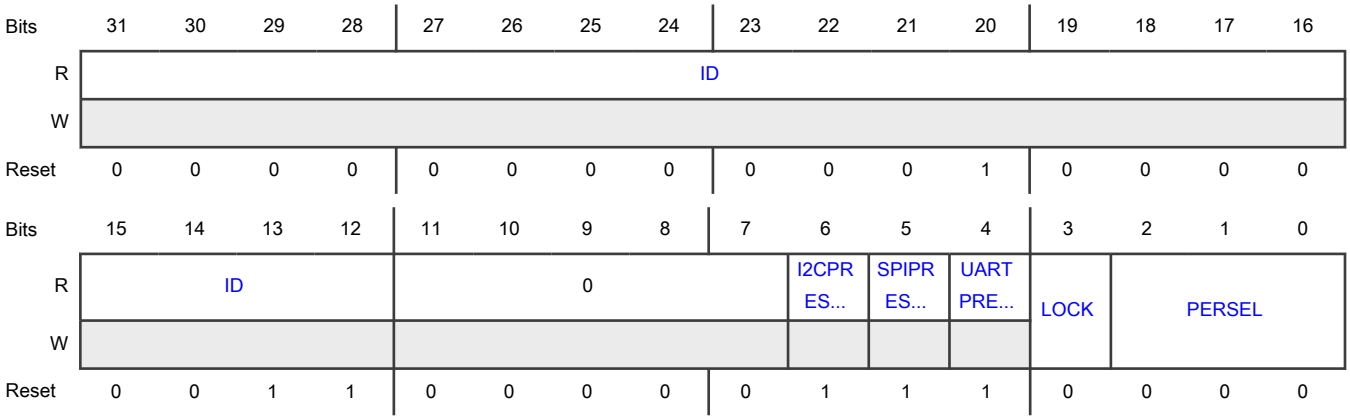
68.2.6.1.3 Peripheral Select and ID (PSELID)

Offset

Register	Offset
PSELID	FF8h

Function
Shows which communications functions are available on a given LP_FLEXCOMM interface and selects the function to be used. It also contains the specific LP_FLEXCOMM interface ID.

Diagram



Fields

Field	Function
31-12 ID	LP_FLEXCOMM interface ID Returns the LP_FLEXCOMM interface ID.
11-7	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
6 I2CPRESENT	I2C Present Indicates if the I2C function is supported in a given LP_FLEXCOMM interface. 0b - Not supported 1b - Supported
5 SPIPRESENT	SPI Present Indicates if the SPI function is supported in a given LP_FLEXCOMM interface. 0b - Not supported 1b - Supported
4 UARTPRESENT	UART Present Indicates if the UART function is supported in a given LP_FLEXCOMM interface. 0b - Not supported 1b - Supported
3 LOCK	Lock Locks the Peripheral Select (PERSEL) field until the next reset. 0b - PERSEL is writable 1b - PERSEL is not writable
2-0 PERSEL	Peripheral Select Selects the peripheral communications function to be used. 000b - No peripheral selected 001b - UART 010b - SPI 011b - I2C 111b - UART and I2C

68.3 Low-Power Inter-Integrated Circuit (LPI2C)

68.3.1 Overview

LPI2C supports an efficient interface to an I2C bus as a controller and target:

- Implements logic support for Standard, Fast, Fast+, Ultra Fast, and High-Speed (HS) modes of operation
- Uses little CPU overhead, with DMA offloading of FIFO register accesses
- Continues operating in Stop modes if an appropriate clock is available

LPI2C also complies with the System Management Bus (SMBus) Specification, version 3. The SMBus is a single-ended simple two-wire bus, which is typically used for low-bandwidth communications.

The Inter-Integrated Circuit (I^2C) serial bus is multi-controller, multi-target, packet-switched, and single-ended, and is often used to attach microcontroller ICs to lower-speed peripheral ICs.

NOTE

Terminology in this chapter has been updated to align with I^2C-bus specification, Rev. 7.0, as shown in Table 640.

Table 640. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

68.3.1.1 Block diagram

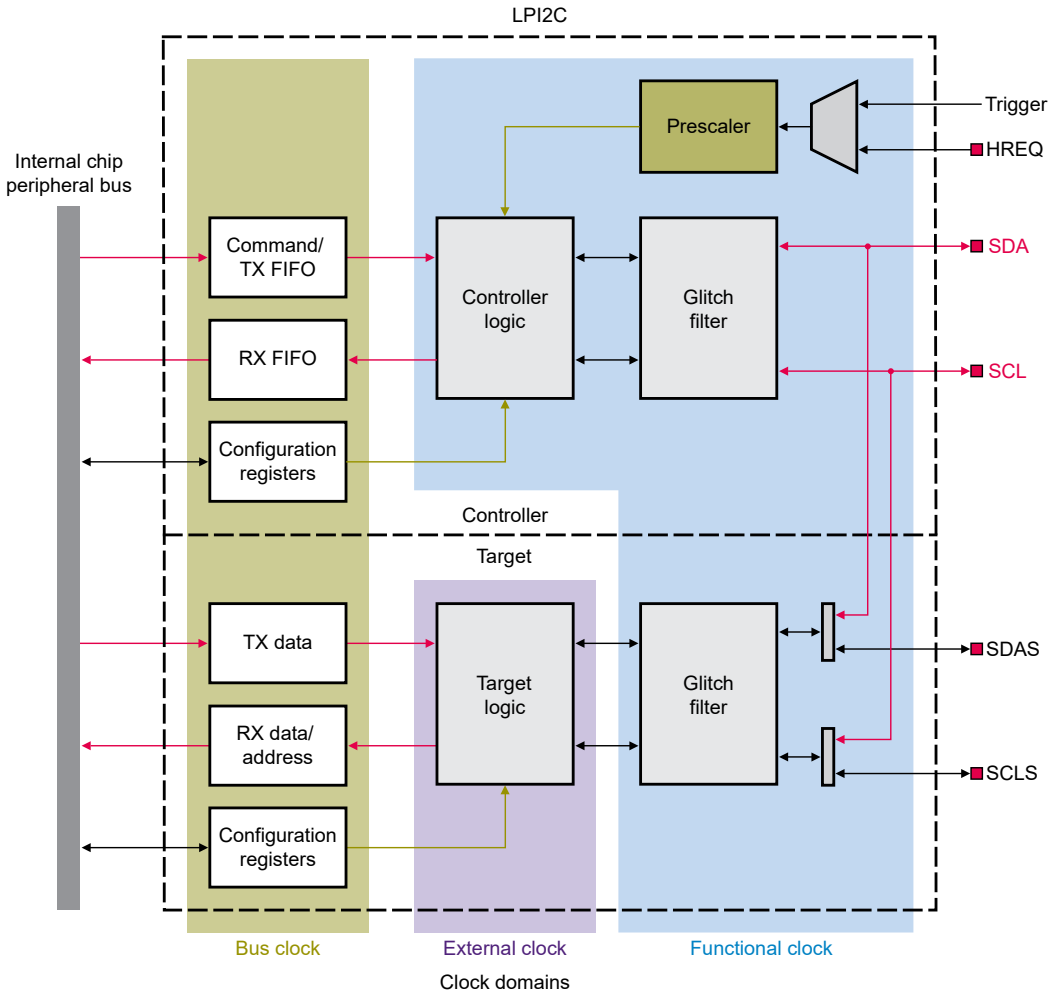


Figure 282. Block diagram

68.3.1.2 Features

LPI2C supports:

- Standard, Fast, Fast+ and Ultra Fast modes

- HS mode in target mode and controller mode
- Multicontroller, including synchronization and arbitration, means that any number of controller nodes can be present. Also, controller and target roles can be changed between messages (after a Stop signal is sent).
- Clock stretching. Used on the SCL line, as an I2C flow control mechanism.
- Arbitration for when the system has more than one controller. When used on the SDA line, ensures that there is only one I2C transmitter at a time.
- General call, seven-bit addressing, and ten-bit addressing
- Software reset, Start byte, and device ID (also require software support)

The LPI2C controller supports:

- Command and transmit FIFO of 8 words (8-bit transmit data + 3-bit command)
- Receive FIFO of 8 words (8-bit receive data).
- Command FIFO waiting for an I2C idle bus before initiating a transfer
- Initiation of repeated Start and Stop conditions and one or more controller-receiver transfers by command FIFO
- Stop condition generation from command FIFO, or automatic generation of Stop condition when the transmit FIFO is empty
- Host request input to control the start time of an I2C bus transfer
- Interrupt generation on data match and unwanted data rejection, via flexible receive data match
- Flags and optional interrupt signals at repeated Start condition, Stop condition, loss of arbitration, unexpected NACK, and command word errors
- Configurable bus idle timeout and pin-stuck-low timeout

The LPI2C target supports:

- Separate I2C target registers to minimize software overhead because of controller or target switching
- 7-bit or 10-bit addressing, address range, SMBus alert, and general call address.
- Transmit data register that supports interrupt or DMA requests
- Receive data register that supports interrupt or DMA requests
- Software-controllable ACK or NACK, with optional clock stretching on ACK or NACK field
- Configurable clock stretching, to avoid transmit-FIFO-underrun and receive-FIFO-overrun errors
- Flags and optional interrupt at end of packet, Stop condition, or bit error detection

68.3.2 Functional description

68.3.2.1 Controller mode

The LPI2C controller logic operates independently from the target logic to perform all controller-mode transfers on the I2C bus.

68.3.2.1.1 Transmit and Command FIFO commands

The transmit FIFO stores command data to initiate various I2C operations. The following operations can be initiated through commands in the transmit FIFO:

- Start or repeated Start condition with address byte, expecting ACK or NACK.
- Transmit data. This operation is the default for zero-extended-byte writes to the transmit FIFO.
- Receive 1-256 bytes of data. You can configure this operation to discard received data and not to store it in the receive FIFO.

- Stop condition. You can configure this operation to send a Stop condition when the transmit FIFO is empty.

Multiple transmit and receive commands can be inserted between the Start and Stop conditions. To comply with the I2C specification, transmit and receive commands must not be interleaved. The receive data command and the receive data and discard commands can be interleaved. This interleaving ensures that only the desired received data is stored in the receive FIFO (or compared with the data match logic).

The LPI2C controller automatically transmits a NACK on the last byte of a receive data command. It transmits the NACK unless the next command in the FIFO is also a receive data command. If the transmit FIFO is empty when a receive data command completes, a NACK is also automatically transmitted.

The LPI2C controller supports 10-bit addressing via a (repeated) Start condition, followed by a transmit data byte containing the second address byte, followed by any number of data bytes with the controller transmit data.

A Start or repeated Start condition expecting a NACK (for example, HS mode controller code) must be followed by a Stop or (repeated) Start condition.

68.3.2.1.2 Controller operations

When LPI2C is enabled, it monitors the I2C bus to detect when the I2C is idle (MSR[BBF]). If either SCL or SDA are low, the I2C bus is no longer considered idle. The bus becomes idle if a Stop condition is detected or if a bus idle timeout is detected (as configured by MCFGR2[BUSIDLE]). After the bus is idle, if the transmit FIFO is not empty and the host request is asserted or disabled, the LPI2C controller initiates a transfer on the bus. This transfer involves the following steps:

1. Wait the bus idle time equal to (MCCR0[CLKLO] + 1) multiplied by the prescaler (MCFGR1[PRESCALE]).
2. Transmit a Start condition and address byte using the timing configuration in Controller Clock Configuration 0 (MCCR0). If an HS mode transfer is configured, the timing configuration from Controller Clock Configuration 1 (MCCR1) is used instead.
3. Perform controller transmit or controller receive transfers, as configured by the transmit FIFO.
4. Transmit NACK on the last byte of a controller receive transfer. This action is performed unless the next command in the transmit FIFO is also a receive data command and the transmit FIFO is not empty.
5. Transmit a repeated Start or Stop condition as configured by the transmit FIFO or MCFGR1[AUTOSTOP]. A repeated Start can change which timing configuration register is used.

When MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. It also relaxes some restrictions on the order of FIFO commands. For example, it allows a Stop command when it is idle to generate a Start condition, followed by one SCL clock pulse, and then a Stop condition.

When the LPI2C controller is disabled, LPI2C continues emptying the transmit FIFO until a Stop condition is transmitted. (The controller could be disabled due to MCR[MEN] being 0, or automatically due to mode entry.) However, LPI2C no longer stalls the I2C bus by waiting for the transmit or receive FIFO. After the transmit FIFO is empty, LPI2C generates a Stop condition automatically.

The LPI2C controller can stall the I2C bus under certain conditions. This stalling results in SCL pulled low continuously on the first bit of a byte, until these conditions change:

- The LPI2C controller is enabled and busy, the transmit FIFO is empty, and MCFGR1[AUTOSTOP] is 0. The LPI2C controller continues to stall the bus until the transmit FIFO is loaded with more data.
- The LPI2C controller is enabled and receiving data, receive data is not being discarded (due to command or receive data match), and the receive FIFO is full. The LPI2C controller continues to stall the I2C bus until the receive FIFO is emptied.

The LPI2C controller aborts the existing transfer when MCFGR0[ABORT] becomes 1. This action causes the LPI2C controller to complete the existing word and then transmit a Stop condition. If the receive FIFO is full when the receive operation is aborted, the last received data is discarded. A new transfer does not start while MCFGR0[ABORT] is 1.

68.3.2.1.3 Receive FIFO and data matching

The receive FIFO stores receive data during controller-receiver transfers. You can configure the LPI2C controller to discard received data instead of storing it in the receive FIFO. This option is configured via the command word in the transmit FIFO.

Received data supports a receive data match function that can match received data against one of two bytes, or against a masked data byte. You can configure the data match function to compare only the first one or two data words received since the last (repeated) Start condition. Received data that is already discarded due to the command word cannot cause a data match. It delays the match on the first data word received until after the discarded data is received.

You can configure the receiver match function to discard all received data until a data match is detected, using MCFGR0[RDMO]. Following a data match, write 0 to MCFGR0[RDMO] before writing 0 to MSR[DMF] to allow all subsequent data to be received.

68.3.2.1.4 Timing parameters

The LPI2C controller can configure the following timing parameters. Parameters are configured separately for HS mode (Controller Clock Configuration 1 (MCCR1)) and other modes (Controller Clock Configuration 0 (MCCR0)). This separation allows the HS mode controller code to be sent using regular timing parameters. Then it allows a switch to HS mode timing (following a repeated Start) until the next STOP condition.

Configure the LPI2C controller timing parameters, measured in LPI2C functional clock cycles, as shown in Table 641. You must configure these parameters to meet the I2C timing specification for the required mode.

Table 641. Timing parameters

I2C specification timing parameter	I2C specification timing symbol	LPI2C timing parameter (in LPI2C functional clock cycles)
SCL clock period	tSCL	(CLKHI + CLKLO + 2 + SCL_LATENCY) × (2 ^ PRESCALE)
Hold time (repeated) Start condition	tHD:STA	(SETHOLD +1) × (2 ^ PRESCALE)
Low period of the SCL clock	tLOW	(CLKLO + 1) × (2 ^ PRESCALE)
High period of the SCL clock	tHIGH	(CLKHI + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Setup time for a repeated Start condition or Stop condition	tSU:STA, tSU:STO	(SETHOLD + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Data hold time	tHD:DAT	(DATAVD + 1) × (2 ^ PRESCALE)
Data setup time	tSU:DAT	(SDA_LATENCY + 1) × (2 ^ PRESCALE)
Bus free time between a Stop and Start condition	tBUF	(CLKLO + 1 + SDA_LATENCY) × (2 ^ PRESCALE)
Data valid time, data valid acknowledge time	tVD:DAT, tVD:ACK	(DATAVD + 1) × (2 ^ PRESCALE)

Table 642 defines the latency parameters. These parameters assume that the risetime is less than one LPI2C functional clock cycle. The risetime depends on a number of factors, including the I/O propagation delay, the I2C bus loading, and the external pullup resistor sizing. A larger risetime increases the number of cycles that the signal takes to propagate through the synchronizer (and glitch filter), which increases the latency.

Table 642. Synchronization latency

Timing parameter	Timing definition
SCL_LATENCY	ROUNDDOWN ((2 + FILTSCL + SCL_RISETIME) ÷ (2 ^ PRESCALE))
SDA_LATENCY	ROUNDDOWN ((2 + FILTSDA + SDA_RISETIME) ÷ (2 ^ PRESCALE))

The following timing restrictions must be enforced to avoid unexpected Start or Stop conditions on the I2C bus. These restrictions also avoid unexpected Start or Stop conditions detected by the LPI2C controller. The timing restrictions can be summarized as "SDA cannot change when SCL is high outside a transmitted (repeated) Start or Stop condition."

Table 643. LPI2C timing parameter restrictions

Timing parameter	Minimum	Maximum	Comment
CLKLO	03h	—	CLKLO x (2 ^ PRESCALE) > SCL_LATENCY
CLKHI	01h	—	Configure CLKHI to meet the duty cycle requirements in the I2C specification
SETHOLD	02h	—	SETHOLD × (2 ^ PRESCALE) > SDA_LATENCY
DATAVD	01h	CLKLO – SDA_LATENCY – 1	Configure DATAVD to meet the data hold requirement in the I2C specification
FILTSCL	00h	[CLKLO × (2 ^ PRESCALE)] – 3	FILTSCL and FILTSDA are the only parameters not multiplied by (2 ^ PRESCALE)
FILTSDA	FILTSCL	[CLKLO × (2 ^ PRESCALE)] – 3	Configuring FILTSDA greater than FILTSCL can delay the SDA input to compensate for board level skew
BUSIDLE	(CLKLO + SETHOLD + 2) × 2	—	Must also be greater than (CLKHI + 1)

See the UM10204, *I2C-bus specification and user manual*.

See Application information for example LPI2C timing configurations.

68.3.2.1.5 Error conditions

The LPI2C controller monitors errors while it is active. The following conditions generate an error flag and block a new Start condition from being sent, until the flag is cleared by software:

- A Start or Stop condition is detected and is not generated by the LPI2C controller (MSR[ALF] becomes 1).
- Transmitting data on SDA and different values are received (MSR[ALF] becomes 1). LPI2C controller stops driving SDA immediately, but continues to drive SCL for the remainder of the byte.
- NACK is detected when transmitting data, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- NACK is detected and is expecting ACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- ACK is detected and is expecting NACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- Transmit FIFO is requesting to transmit or receive data without a Start condition (MSR[FEF] becomes 1). This restriction is ignored when MCFGR0[RELAX] is 1.
- SCL (or SDA if MCFGR1[TIMECFG] is 1) is low for (MCFGR2[TIMELOW] × 256) prescaler cycles without a pin transition (MSR[PLTF] becomes 1).

You must respond to MSR[PLTF] to terminate the existing command. You can terminate the command cleanly by writing 0 to MCR[MEN], or you can terminate it abruptly by writing 1 to MCR[RST].

You can use MCFGR0[RELAX] to attempt to recover a target with SDA stuck low. If MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. Initiating a Start command with address FFh, then the Stop condition, should generate sufficient SCL clock edges to cause the target to release SDA.

You can use MCFGR2[BUSIDLE] to force the I2C bus to be considered idle when SCL and SDA remain high for (BUSIDLE + 1) prescaler cycles. The bus is considered idle when the LPI2C controller is first enabled. When BUSIDLE is configured greater than zero, then SCL or SDA must be high for (BUSIDLE + 1) prescaler cycles before the I2C bus is considered idle.

68.3.2.1.6 DMA support

For efficient DMA transfers to the transmit and command FIFO, two alias regions support incrementing 8-bit, 16-bit, or 32-bit write accesses to that FIFO.

- Controller Transmit Command Burst (MTCBR0 - MTCBR127) is a 512-byte region that supports pushing CMD and DATA into the transmit FIFO.
- Transmit Data Burst (MTDBR0 - MTDBR252) is a 1024-byte region that supports pushing zero extended DATA into the transmit FIFO.

The two regions are contiguous. A DMA transfer can start in the MTCBR region to initialize the transfer including START condition with address. The same transfer can complete in the MTDBR region with the data to transmit without changing the transfer size.

The transmit FIFO prevents writes that overflow the FIFO, but this prevention does not signal an error. Do not perform 32-bit writes to the MTDBR unless there are four empty slots in the transmit FIFO. Do not perform 16-bit writes to the MTDBR unless there are two empty slots in the transmit FIFO.

68.3.2.1.7 Pin configuration

Configuration	Description
Open-drain support	The LPI2C controller defaults to open-drain configuration of the SDA and SCL pins. Support for true open drain depends on the specific device, and requires the pins where LPI2C pins are muxed to support true open drain.
HS mode support	Support for HS mode depends on the specific device. This mode requires the SCL pin to support the current source pullup required in the I2C specification.
Ultra-Fast mode support	The LPI2C controller supports the output-only push-pull function required for I2C Ultra-Fast mode using the SDA and SCL pins. Support for Ultra-Fast mode also requires MCFGR1[IGNACK] to be 1.
Push-pull two-wire support	A push-pull two-wire configuration is available to the LPI2C controller. If LPI2C is the only controller and all I2C pins on the bus are at the same voltage, this configuration may support a partial HS mode. A partial HS mode, not a full HS mode, because this configuration actively drives high rather than enabling a current service pull-up. This configuration sets the SCL pin as push-pull for every clock except the first clock pulse, to allow HS-mode-compatible targets to perform clock stretching. In this mode, the SDA pin is tristated for controller-receive data bits and controller-transmit ACK/NACK bits, and is configured as push-pull at other times. To avoid the risk of contention when SDA is push-pull, configure the pin for open-drain operation, as part of the device-specific configuration.
Push-pull four-wire support	The push-pull four-wire configuration separates the SCL input data and output data into separate pins. It also separates the SDA input data and output data into separate pins. The SCL/SDA pins are used for input data; the SCLS/SDAS pins are used for output data, with configurable polarity. This configuration simplifies external connections when connecting the LPI2C to the I2C bus through external level shifters or discrete components. When using this four-wire configuration, the LPI2C controller logic and LPI2C target logic cannot connect to separate I2C buses.

68.3.2.2 Target mode

To perform all target mode transfers on the I2C bus, the LPI2C target logic operates independently from the LPI2C controller logic.

68.3.2.2.1 Address matching

You can configure the LPI2C target:

- To match one of two addresses, using either 7-bit or 10-bit addressing modes for each address.
- To match a range of addresses in either 7-bit or 10-bit addressing modes.
- To match the general call address and generate appropriate flags.
- To match the SMBus alert address and generate appropriate flags.
- To detect the HS mode controller code address, and to disable the digital filters and output valid delay time until the next Stop condition is detected.

After a valid address is matched, the LPI2C target automatically performs target-transmit or target-receive transfers until:

- A NACK is detected (unless SCFGR1[IGNACK] becomes 1).
- A bit error is detected (the LPI2C target is driving SDA, but a different value is sampled).
- A (repeated) Start or Stop condition is detected.

When SCFGR1[RXALL] is 1, the LPI2C target receives all addresses and data on the I2C bus. Unless the address is matched, it never drives the SDA pin. Configure the LPI2C target to match only on 7-bit address mode when this feature is enabled. Software can support 10-bit address mode. This option is intended for debugging the contents of an I2C transfer between another controller and target. Likewise, SCFGR1[SDCFG] and SCFGR1[RSCFG] configure the Stop or repeated Start flags to assert on all Stop or repeated Start conditions, even when there is no address match.

68.3.2.2.2 Transmit and receive data

Target Transmit Data (STDR) and Target Receive Data (SRDR) are double-buffered and only update during a target-transmit and target-receive transfer, respectively.

You can configure the target address that was received to be read from SRDR (for example, when using DMA to transfer data) or from Target Address Status (SASR).

You can configure STDR to request data only after a target-transmit transfer is detected. You can also configure it to request new data whenever STDR is empty.

Write to STDR only when SSR[TDF] is set.

Read SRDR only when SSR[RDF] is set, or when SSR[AVF] is set and SCFGR1[RXCFG] = 1.

Read SASR only when SSR[AVF] is set.

68.3.2.2.3 Read request

The LPI2C target generates a read request when the I2C bus is idle and you write 1 to SCFGR0[RDREQ]. This occurrence causes the LPI2C target to pull the SDA pin low until either the first SCL falling edge or SCFGR0[RDREQ] becomes 0. Following the next Stop condition or a software timeout, you can proceed as follows:

- If the LPI2C target is accessed at the correct address, the read request is acknowledged, and you must write 0 to SCFGR0[RDREQ].
- If SCFGR0[RDACK] is 1, the read request is acknowledged by a Start followed by one SCL pulse followed by a STOP condition. You must write 0 to SCFGR0[RDREQ].
- If neither of the first two cases occurred, the read request is not acknowledged. You must write 0 and then 1 to SCFGR0[RDREQ] after an appropriate delay, provided the I2C bus is idle.
- If a software timeout occurs, the read request is not acknowledged, and you should write 0 to SCFGR0[RDREQ]. Another request can be generated after an appropriate delay.

Writing to SCFGR0[RDREQ] when the I2C bus is busy results in a logic 0 being written. Write 1 to SCFGR0[RDREQ], then confirm that the register is written to correctly. If the register does not update correctly, the I2C bus is no longer idle and the read request must be attempted later.

68.3.2.2.4 Clock stretching

The LPI2C target supports many configurable options for clock stretching. You can configure these conditions to perform clock stretching:

- SSR[AVF] is set during the ninth clock pulse of the address byte.
- SSR[TDF] is set during the ninth clock pulse of a target-transmit transfer.
- SSR[RDF] is set during the ninth clock pulse of a target-receive transfer.
- SSR[TAF] is set during the eighth clock pulse of an address byte or a target-receive transfer. In HS mode, this option is disabled.
- Clock stretching can be extended for a number of cycles equal to the value of SCFGR2[CLKHOLD] cycles. This stretching allows additional setup time to sample the SDA pin externally. In HS mode, this option is disabled.

When clock stretching is enabled, clock stretching extends for one peripheral bus clock cycle after SDA updates, unless extended by the SCFGR2[CLKHOLD] configuration.

68.3.2.2.5 Timing parameters

The LPI2C target can configure the following timing parameters:

- SDA data valid time from SCL negation to SDA update
- SCL hold time when clock stretching is enabled to increase setup time when sampling SDA externally
- SCL glitch filter time
- SDA glitch filter time

These parameters are disabled when SCR[FILTEN] is 0, when SCR[FILTDZ] is 1 in Doze mode, and when LPI2C target detects HS mode. When disabled, the LPI2C target is clocked directly from the I2C bus. In this case, the target may not satisfy all timing requirements of the I2C specification (such as SDA minimum hold time in Standard/Fast mode).

The LPI2C target places the following restrictions on the timing parameters:

- You must configure SCFGR2[FILTSDA] to be greater than or equal to SCFGR2[FILTSCL] (unless compensating for board level skew between SDA and SCL).
- You must configure SCFGR2[DATAVD] to be less than the minimum SCL low period.

68.3.2.2.6 Error conditions

The LPI2C target can flag the following error conditions:

- SSR[BEF] is set when the LPI2C target is driving SDA but it samples a different value than what is expected.
- SSR[FEF] is set due to a transmit data underrun or a receive data overrun. To eliminate the possibility of underrun and overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on a receive data overrun.
- SSR[FEF] is also set due to an address overrun, but only when SCFGR1[RXCFG] is 1. To eliminate the possibility of overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on an address overrun regardless of the value of SCFGR1[RXCFG].

The LPI2C target does not implement a timeout due to SCL or SDA being stuck low. If this detection is required, use the LPI2C controller logic so you can reset the LPI2C target when this condition is detected.

68.3.2.3 Low-power modes

LPI2C remains functional during low-power modes, if MCR[DOZEN] = 0 and LPI2C uses an external or internal clock source that remains enabled. LPI2C can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPI2C to be disabled in low-power modes when MCR[DOZEN] = 1. In this case, LPI2C waits for the current transfer to complete any pending operation.

NOTE

See the chip-specific information for low-power modes available on your chip.

68.3.2.4 Debug mode

Table 644. Debug mode

Mode	LPI2C operation
Debug	If MCR[DBGEN] = 1, can continue operating in Debug mode.

68.3.2.5 Peripheral triggers

The connection of the LPI2C peripheral triggers to other peripherals depends upon the specific device being used.

Table 645. LPI2C triggers

Trigger	Description
Controller output trigger	Generates an output trigger that can be connected to other peripherals on the device. The controller output trigger asserts on either a repeated Start or a Stop condition. The trigger remains asserted for one cycle of the LPI2C functional clock divided by MCFGR1[PRESCALE].
Target output trigger	Generates an output trigger that can be connected to other peripherals on the device. The target output trigger asserts on either a repeated Start or a Stop condition that occurs after a target address match. The target output trigger remains asserted until the next target SCL pin negation.
Input trigger	To control the start of a LPI2C bus transfer, the LPI2C input trigger can be selected instead of the HREQ input. The input trigger is synchronized. To be detected, the input trigger must assert for at least two cycles of the LPI2C functional clock divided by the value of MCFGR1[PRESCALE]. When LPI2C is busy, the HREQ input (and therefore the input trigger) is ignored.

68.3.2.6 Clocking

Table 646. LPI2C clocks

Clock	Description
LPI2C functional clock	The LPI2C functional clock is asynchronous to the bus clock. It can remain enabled in low-power modes to support I2C bus transfers by the LPI2C controller. The functional clock is also used by the LPI2C target to support digital filter and data hold time configurations. The LPI2C controller divides the functional clock by a prescaler (MCFGR1[PRESCALE]) and the resulting frequency must be at least eight times faster than the I2C bus bandwidth.
External clock	The LPI2C target logic is clocked directly from the external pins. These pins are SCL and SDA, or SCLS and SDAS if the controller and target are implemented on separate pins). This clocking allows the LPI2C target to remain operational, even when the LPI2C functional clock is disabled. **NOTE** If the LPI2C functional clock is disabled, the LPI2C target digital filter must be disabled. This condition can affect compliance with some timing parameters of the I2C specification, such as data hold time.

Table continues on the next page...

Table 646. LPI2C clocks (continued)

Clock	Description
Bus clock	The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPI2C controller and target registers.

For chip-specific clocking information, see the Clocking chapter.

68.3.2.7 Reset

Table 647. LPI2C resets

Reset	Description
Chip reset	The logic and registers for the LPI2C controller and target are reset to their default states after a chip reset.
Software reset	The LPI2C controller implements a software reset field in its control register. MCR[RST] resets all controller logic and registers to their default states, except for Controller Control (MCR) itself. The LPI2C target implements a software reset field in its control register. SCR[RST] resets all target logic and registers to their default states, except for Target Control (SCR) itself.
FIFO reset	The LPI2C controller implements write-only control fields that reset the transmit FIFO (MCR[RTF]) and receive FIFO (MCR[RRF]). After a FIFO is reset, that FIFO is empty. The LPI2C target implements write-only control fields that reset the transmit data register (SCR[RTF]) and receive data register (SCR[RRF]). After a data register is reset, that data register is empty.

68.3.2.8 Interrupts and DMA requests

Depending on the configuration, interrupts and DMA requests can be combined:

- LPI2C controller and target interrupts
- LPI2C controller and target transmit DMA requests
- LPI2C controller and target receive DMA requests

68.3.2.8.1 Controller mode

Table 648 lists the Controller mode sources that can generate LPI2C controller interrupts and LPI2C controller transmit and receive DMA requests.

Table 648. Controller interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (MSR[TDF])	Data can be written to transmit FIFO, as configured by MFCR[TXWATER].	Y	TX	Y
Receive Data Flag (MSR[RDF])	Data can be read from the receive FIFO, as configured by MFCR[RXWATER].	Y	RX	Y

Table continues on the next page...

Table 648. Controller interrupts and DMA requests (continued)

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
End Packet Flag (MSR[EPF])	Controller has transmitted a repeated Start or Stop condition.	Y	N	Y
Stop Detect Flag (MSR[SDF])	Controller has transmitted a Stop condition (and the LPI2C controller is idle, if MCFGR1[STOPCFG] = 1) .	Y	N	Y
NACK Detect Flag (MSR[NDF])	During an address byte, the controller expects an ACK but detects a NACK. During an address byte, the controller expects a NACK but detects an ACK. During a controller-transmitter data byte, the controller detects a NACK.	Y	N	Y
Arbitration Lost Flag (MSR[ALF])	The controller lost arbitration due to a Start or Stop condition detected at the wrong time, or the controller was transmitting data but received data different from the data that was transmitted.	Y	N	Y
FIFO Error Flag (MSR[FEF])	The controller expects a Start condition in the command FIFO, but the next entry in the command FIFO is not a Start condition.	Y	N	Y
Pin Low Timeout Flag (MSR[PLTF])	Pin low timeout is enabled and SCL (or SDA, if configured) is low for longer than the configured timeout.	Y	N	Y
Data Match Flag (MSR[DMF])	The received data matches the configured data match, but the received data is not discarded due to a command FIFO entry.	Y	N	Y
Start Detect Flag (MSR[STF])	A Start condition is detected on the I2C bus (and the LPI2C controller is idle, if MCFGR1[STARTCFG] = 1).	Y	N	Y
Controller Busy Flag (MSR[MBF])	LPI2C controller is busy transmitting or receiving data.	N	N	N
Bus Busy Flag (MSR[BBF])	LPI2C controller is enabled and activity is detected on the I2C bus, but no Stop condition is detected and no bus idle timeout (if enabled) occurred.	N	N	N

68.3.2.8.2 Target mode

Table 649 lists the target mode sources that can generate LPI2C target interrupts and LPI2C target transmit and receive DMA requests.

Table 649. Target interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (SSR[TDF])	Data can be written to Target Transmit Data (STDR).	Y	TX	Y
Receive Data Flag (SSR[RDF])	Data can be read from Target Receive Data (SRDR).	Y	RX	Y
Address Valid Flag (SSR[AVF])	Address can be read from Target Address Status (SASR).	Y	RX	Y
Transmit ACK Flag (SSR[TAF])	ACK or NACK can be written to Target Transmit ACK (STAR).	Y	N	Y
Repeated Start Flag (SSR[RSF])	Target has detected an address match followed by a repeated Start condition.	Y	RX	Y
Stop Detect Flag (SSR[SDF])	Target has detected an address match followed by a Stop condition.	Y	RX	Y
Bit Error Flag (SSR[BEF])	Target was transmitting data, but received data is different from what was transmitted.	Y	N	Y
FIFO Error Flag (SSR[FEF])	This flag is set by: • Transmit data underrun • Receive data overrun • Address status overrun when SCFGR1[RXCFG] = 1 This flag can only be set when clock stretching is disabled.	Y	N	Y
Address Match 0 Flag (SSR[AM0F])	Target detected an address match SAMR[ADDR0].	Y	N	N
Address Match 1 Flag (SSR[AM1F])	Target detected an address match with SAMR[ADDR1] or using an address range.	Y	N	N
General Call Flag (SSR[GCF])	Target detected an address match with the general call address.	Y	N	N
SMBus Alert Response Flag (SSR[SARF])	Target detected an address match with the SMBus alert address.	Y	N	N
Target Busy Flag (SSR[SBF])	LPI2C target is busy receiving an address byte or is transmitting or receiving data.	N	N	N
Bus Busy Flag (SSR[BBF])	LPI2C target is enabled and a Start condition is detected on I2C bus, but no Stop condition detected.	N	N	N

68.3.2.8.3 End-of-packet DMA transfer

End-of-packet functionality serves serial interfaces where the transfer size may not be known in advance and the data is pushed by an external device. Examples include UART receive, I2C Target mode, and SPI Target mode. End-of-packet processing

is intended to ensure that data does not become stranded in either the receive FIFO or DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet differs for each serial interface but is processed by the serial peripheral and DMA in the same way. For example:

- UART end of packet is signaled by an idle line condition.
- I2C end of packet is signaled by Stop or repeated Start condition.
- SPI end of packet is signaled by PCS negation.

When the serial peripheral is configured to signal an end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO regardless of the watermark configuration. For larger watermark configurations, this behavior ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO starts a new packet, data is not pulled from the receive FIFO. Also, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO does not start a new packet, the DMA transfers data as normal.

Because the DMA may transfer multiple words per request, the end-of-packet condition persists until the DMA minor loop finishes and no additional data is pulled from the FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes after the end of packet is signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into system memory. It also saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. The final destination address can be saved to system memory or is available in the destination address register.

Because the DMA terminates the major loop, the receive FIFO is not serviced until the DMA is reconfigured. You can perform this reconfiguration through software or hardware (for example, channel linking or scatter-gather). Minimize this delay to avoid receiver FIFO overrun. Automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. In this case, the DMA spends more time processing the end of packet than transferring the data. For example, to avoid an excessive number of idle conditions, increase the UART idle line length as needed.

68.3.3 External signals

Table 650. External signals

Signal	Name	Two-wire scheme	Four-wire scheme	Direction
SCL	LPI2C clock line	SCL	In Four-Wire mode, this pin is the SCL input pin.	Input or output
SDA	LPI2C data line	SDA	In Four-Wire mode, this pin is the SDA input pin.	Input or output
SCLS	Secondary I2C clock line	Not used	In Four-Wire mode, this pin is the SCLS output pin. If LPI2C controller/target are configured to use separate pins, then this pin is the LPI2C target SCL pin.	Input or output
SDAS	Secondary I2C data line	Not used	In Four-Wire mode, this pin is the SDAS output pin. If LPI2C controller/target are	Input

Table continues on the next page...

Table 650. External signals (continued)

Signal	Name	Two-wire scheme	Four-wire scheme	Direction
			configured to use separate pins, then this pin is the LPI2C target SDA pin.	or output
HREQ	Host request	When configured for controller (input), if host request is asserted and the I2C bus is idle, it initiates a transfer. When configured for target (output), it asserts while valid data is present in Target Transmit Data (STDR). HREQ is an additional pin separate from the two-wire or four-wire scheme.		Input or output

Figure 283 shows the two-signal connection.

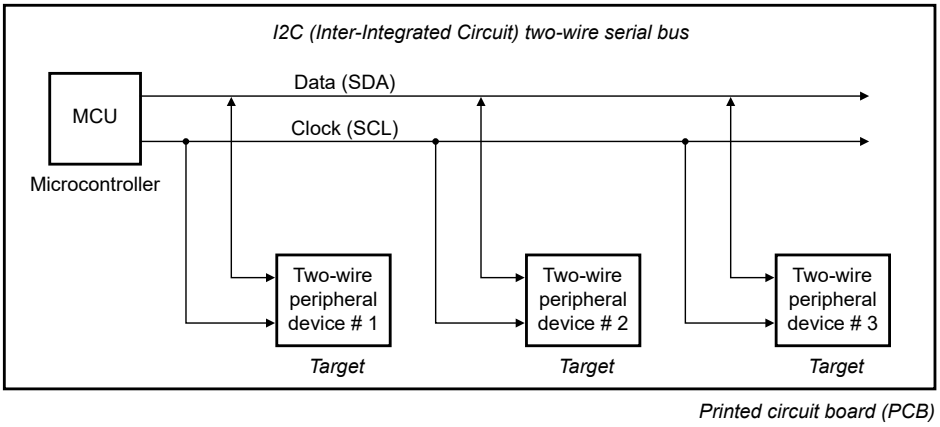


Figure 283. I²C two-wire serial bus

Figure 284 shows a possible four-signal connection.

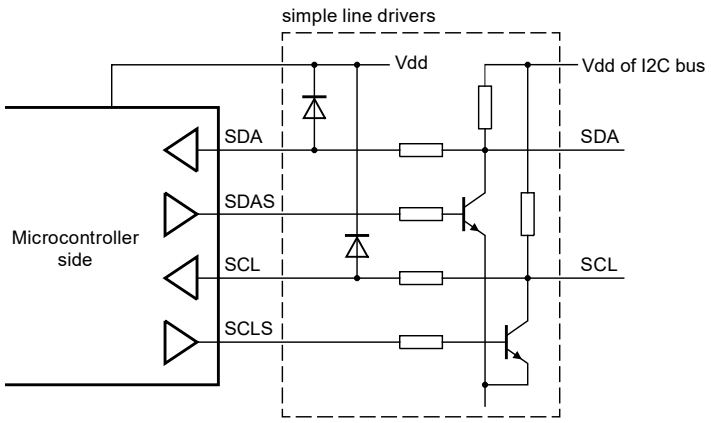


Figure 284. I²C four-wire serial bus

68.3.4 Initialization

To initialize the LPI2C controller:

1. Configure Controller Configuration 0 (MCFGR0)–Controller Configuration 3 (MCFGR3) as required by the application.

2. Configure Controller Clock Configuration 0 (MCCR0) and Controller Clock Configuration 1 (MCCR1) to satisfy the timing requirements of the I2C mode supported by the application.
3. Enable controller interrupts and DMA requests as required by the application.
4. Enable the LPI2C controller by writing 1 to MCR[MEN].

To initialize the LPI2C target:

1. Configure Target Address Match (SAMR) with the I2C address of the target location on the I2C bus.
2. Configure Target Configuration 0 (SCFGR0) and Target Configuration 1 (SCFGR1) as required by the application.
3. Configure Target Configuration 2 (SCFGR2) to satisfy the timing requirements of the I2C mode supported by the application.
4. Enable target interrupts and DMA requests as required by the application.
5. Enable the LPI2C target by writing 1 to SCR[SEN].

68.3.5 Application information

Configure the I2C timing parameters to meet the requirements of the I2C specification. This configuration depends on the supported mode and LPI2C functional clock frequency. When switching between two modes using different clock configuration registers (for example, Fast mode and HS mode), MCFGR1[PRESCALE] must remain constant between the modes.

Table 651. Example timing configurations

I2C mode	Clock frequency	Baud rate	PRESCALE	FILTSCL / FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Standard	8 MHz	100 kbit/s	0h	0h/0h	24h	28h	24h	02h
Standard	48 MHz	100 kbit/s	2h	1h/1h	37h	3Fh	37h	03h
Standard	60 MHz	100 kbit/s	2h	1h/1h	45h	50h	44h	04h
Fast	8 MHz	400 kbit/s	0h	0h/0h	04h	0Bh	05h	02h
Fast+	8 MHz	1 Mbit/s	0h	0h/0h	02h	03h	01h	01h
Fast	48 MHz	400 kbit/s	0h	1h/1h	1Dh	3Eh	35h	0Fh
Fast	48 MHz	400 kbit/s	2h	1h/1h	07h	11h	0Bh	03h
Fast+	48 MHz	1 Mbit/s	2h	1h/1h	03h	06h	04h	04h
HS	48 MHz	3.2 Mbit/s	0h	0h/0h	07h	08h	03h	01h
Fast	60 MHz	400 kbit/s	1h	2h/2h	11h	28h	1Fh	08h
Fast+	60 MHz	1 Mbit/s	1h	2h/2h	07h	0Fh	0Bh	01h
HS	60 MHz	3.33 Mbit/s	1h	0h/0h	04h	04h	02h	01h

68.3.6 Memory map and registers

68.3.6.1 LPI2C register descriptions

Writing to a read-only register or reading from a write-only register can cause bus errors. This module does not check whether programmed values in the registers are correct; you must ensure that valid programmed values are written to the registers.

68.3.6.1.1 LPI2C memory map

LPI2C0 base address: 4011_0800h

LPI2C1 base address: 4011_1800h

LPI2C2 base address: 4011_2800h

LPI2C3 base address: 4011_3800h

LPI2C4 base address: 4017_1800h

LPI2C5 base address: 4017_2800h

LPI2C6 base address: 4017_3800h

LPI2C7 base address: 4017_4800h

LPI2C8 base address: 4019_9800h

LPI2C9 base address: 4019_A800h

LPI2C10 base address: 4019_B800h

LPI2C11 base address: 4019_C800h

LPI2C12 base address: 4019_D800h

LPI2C13 base address: 4019_E800h

LPI2C15 base address: 4021_3000h

LPI2C17 base address: 4032_6800h

LPI2C18 base address: 4032_7800h

LPI2C19 base address: 4032_8800h

LPI2C20 base address: 4032_9800h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	See section
4h	Parameter (PARAM)	32	R	0000_0303h
10h	Controller Control (MCR)	32	RW	0000_0000h
14h	Controller Status (MSR)	32	RW	0000_0001h
18h	Controller Interrupt Enable (MIER)	32	RW	0000_0000h
1Ch	Controller DMA Enable (MDER)	32	RW	0000_0000h
20h	Controller Configuration 0 (MCFGR0)	32	RW	0000_0000h
24h	Controller Configuration 1 (MCFGR1)	32	RW	0000_0000h
28h	Controller Configuration 2 (MCFGR2)	32	RW	0000_0000h
2Ch	Controller Configuration 3 (MCFGR3)	32	RW	0000_0000h
40h	Controller Data Match (MDMR)	32	RW	0000_0000h
48h	Controller Clock Configuration 0 (MCCR0)	32	RW	0000_0000h
50h	Controller Clock Configuration 1 (MCCR1)	32	RW	0000_0000h
58h	Controller FIFO Control (MFCR)	32	RW	0000_0000h
5Ch	Controller FIFO Status (MFSR)	32	R	0000_0000h
60h	Controller Transmit Data (MTDR)	32	W	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
70h	Controller Receive Data (MRDR)	32	R	0000_4000h
78h	Controller Receive Data Read Only (MRDROR)	32	R	0000_4000h
110h	Target Control (SCR)	32	RW	0000_0000h
114h	Target Status (SSR)	32	RW	0000_0000h
118h	Target Interrupt Enable (SIER)	32	RW	0000_0000h
11Ch	Target DMA Enable (SDER)	32	RW	0000_0000h
120h	Target Configuration 0 (SCFGR0)	32	RW	0000_0000h
124h	Target Configuration 1 (SCFGR1)	32	RW	0000_0000h
128h	Target Configuration 2 (SCFGR2)	32	RW	0000_0000h
140h	Target Address Match (SAMR)	32	RW	0000_0000h
150h	Target Address Status (SASR)	32	R	0000_4000h
154h	Target Transmit ACK (STAR)	32	RW	0000_0000h
160h	Target Transmit Data (STDR)	32	W	0000_0000h
170h	Target Receive Data (SRDR)	32	R	0000_4000h
178h	Target Receive Data Read Only (SRDROR)	32	R	0000_4000h
200h - 3FCh	Controller Transmit Command Burst (MTCBR0 - MTCBR127)	32	W	0000_0000h
400h - 7F0h	Transmit Data Burst (MTDBR0 - MTDBR252)	32	W	0000_0000h
7F4h - 7FCh	Transmit Data Burst (MTDBR253 - MTDBR255)	32	W	0000_0000h

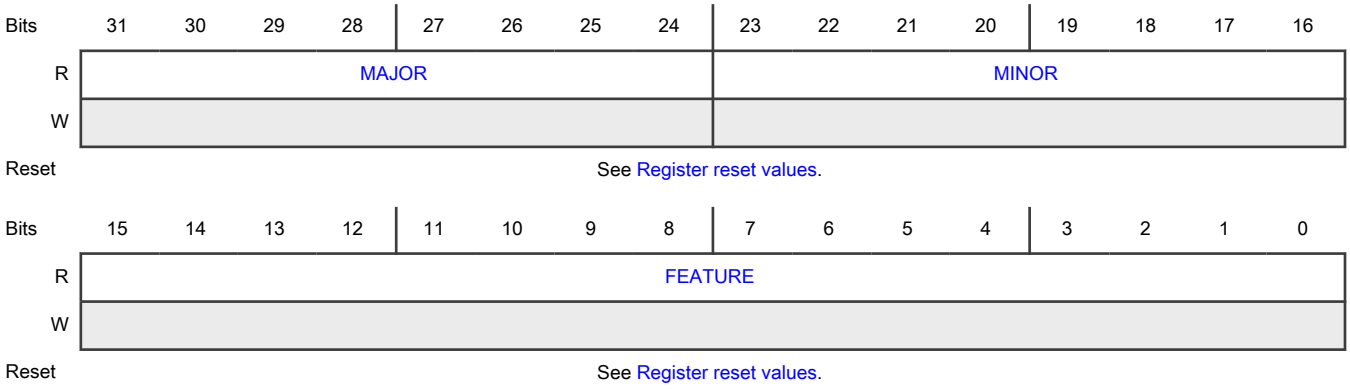
68.3.6.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function
Contains version numbers for the module design and feature set.

Diagram



Register reset values

Register	Reset value
VERID	LPI2C0–LPI2C13: 0103_0003h LPI2C15: 0103_0002h LPI2C17–LPI2C20: 0103_0003h

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module design specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module design specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number. 0000_0000_0000_0010b - Controller only, with standard feature set 0000_0000_0000_0011b - Controller and target, with standard feature set

68.3.6.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains parameter values implemented in the module.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MRXFIFO				0				MTXFIFO			
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Fields

Field	Function
31-16 —	Reserved
15-12 —	Reserved
11-8 MRXFIFO	Controller Receive FIFO Size Configures the number of words in the controller receive FIFO to $2^{MRXFIFO}$.
7-4 —	Reserved
3-0 MTXFIFO	Controller Transmit FIFO Size Configures the number of words in the controller transmit FIFO to $2^{MTXFIFO}$.

68.3.6.1.4 Controller Control (MCR)

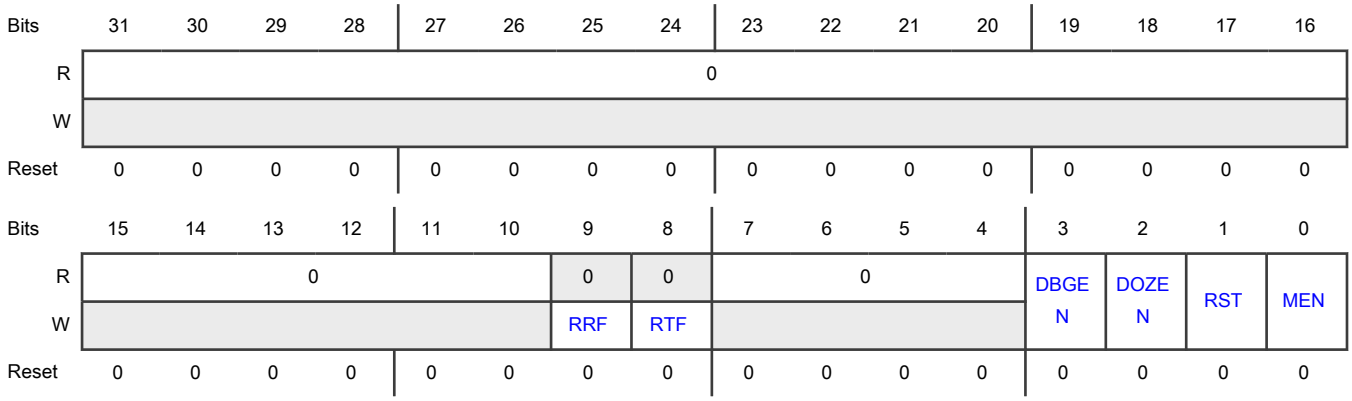
Offset

Register	Offset
MCR	10h

Function

Contains resets, debug enable, and other controller control settings.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Resets the receive FIFO in Controller Receive Data (MRDR). 0b - No effect 1b - Reset receive FIFO
8 RTF	Reset Transmit FIFO Resets the transmit FIFO in Controller Transmit Data (MTDR). 0b - No effect 1b - Reset transmit FIFO
7-4 —	Reserved
3 DBGEN	Debug Enable Enables the controller in Debug mode. 0b - Disable 1b - Enable
2 DOZEN	Doze Mode Enable Enables the controller in Doze mode. 0b - Enable 1b - Disable
1	Software Reset

Table continues on the next page...

Table continued from the previous page...

Field	Function
RST	Resets all internal controller logic and registers except Controller Control (MCR). This field remains 1 (enabled) until you write 0 to it. The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0b - No effect 1b - Reset
0 MEN	Controller Enable Enables the controller logic. 0b - Disable 1b - Enable

68.3.6.1.5 Controller Status (MSR)

Offset

Register	Offset
MSR	14h

Function

Contains status flags for transmit and receive data, for start and stop conditions, and for bus and controller busy or idle status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STF	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-26 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
25 BBF	Bus Busy Flag Specifies whether the I2C bus is busy. 0b - Idle 1b - Busy
24 MBF	Controller Busy Flag Specifies whether the I2C controller is busy. 0b - Idle 1b - Busy
23-16 —	Reserved
15 STF	Start Flag Specifies whether a Start condition is detected on the bus when the bus is idle. When MCFGR1[STARTCFG] is 0, this field becomes 1 only if the LPI2C controller is also idle. When MCFGR1[STARTCFG] is 1, STF becomes 1 for any Start condition when the I2C bus is idle. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Start condition not detected 1b - Start condition detected When writing 0b - No effect 1b - Clear the flag
14 DMF	Data Match Flag Indicates whether the received data matches MDMR[MATCH0] or MDMR[MATCH1] (as configured by MCFGR1[MATCFG]). Received data discarded due to MTDR[CMD] does not cause this flag to set. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Matching data not received 1b - Matching data received When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 PLTF	Pin Low Timeout Flag Indicates whether pin low timeout has occurred. Sets when the SCL or SDA input is low for more than the number of PINLOW cycles defined by MCFGR3[PINLOW], even when the LPI2C controller is idle. You must resolve the pin low condition via software. PLTF cannot be cleared as long as the pin low timeout continues. Before LPI2C can initiate a Start condition, you must clear this flag. See MCFGR1[TIMECFG] for the SCL and/or SDA timeout settings. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Pin low timeout did not occur 1b - Pin low timeout occurred When writing 0b - No effect 1b - Clear the flag
12 FEF	FIFO Error Flag Detects the LPI2C controller's attempt to send or receive data without first generating a (repeated) Start condition. This error can occur when the transmit FIFO underflows when MCFGR1[AUTOSTOP] = 1. When this flag is set, the LPI2C controller sends a Stop condition (if busy). The controller does not initiate a new Start condition until the flag is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No FIFO error 1b - FIFO error When writing 0b - No effect 1b - Clear the flag
11 ALF	Arbitration Lost Flag Indicates whether arbitration is lost. Either of these conditions sets this flag: • The LPI2C controller transmits a logic 1 and detects a logic 0 on the I2C bus. • The LPI2C controller detects a Start or Stop condition when the LPI2C controller is transmitting data. When ALF is set, the LPI2C controller releases the I2C bus (goes idle), and the LPI2C controller does not initiate a new Start condition until the ALF is cleared.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - Controller did not lose arbitration 1b - Controller lost arbitration When writing 0b - No effect 1b - Clear the flag
10 NDF	NACK Detect Flag Indicates whether an unexpected NACK has been detected. This flag is set when the LPI2C controller detects a NACK it was not expecting when transmitting an address or data. When set, the controller does not initiate a new Start condition until this flag is cleared. If a NACK is expected for a given address (as configured by the command word), this flag is set if a NACK is not generated. When this flag is set, the LPI2C controller automatically transmits a Stop condition if MCFGR1[AUTOSTOP] = 1, or if the transmit FIFO is not empty. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No unexpected NACK detected 1b - Unexpected NACK detected When writing 0b - No effect 1b - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C controller has generated a Stop condition. When MCFGR1[STOPCFG] = 0, this flag is set for any Stop condition. When MCFGR1[STOPCFG] = 1, this flag is set only when the LPI2C controller is also idle (transmit FIFO is empty). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stop condition generated 1b - Stop condition generated When writing 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Clear the flag
8 EPF	End Packet Flag Indicates whether the LPI2C controller has generated a repeated Start condition or a Stop condition. When the controller first generates a Start condition, this flag is not set. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stop or repeated Start generated 1b - Stop or repeated Start generated When writing 0b - No effect 1b - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the receive data is ready. This flag is set when the number of words in the receive FIFO is greater than MFCR[RXWATER]. 0b - Receive data not ready 1b - Receive data ready
0 TDF	Transmit Data Flag Indicates whether transmit data is requested. This flag is set when the number of words in the transmit FIFO is equal or less than MFCR[TXWATER]. 0b - Transmit data not requested 1b - Transmit data requested

68.3.6.1.6 Controller Interrupt Enable (MIER)

Offset

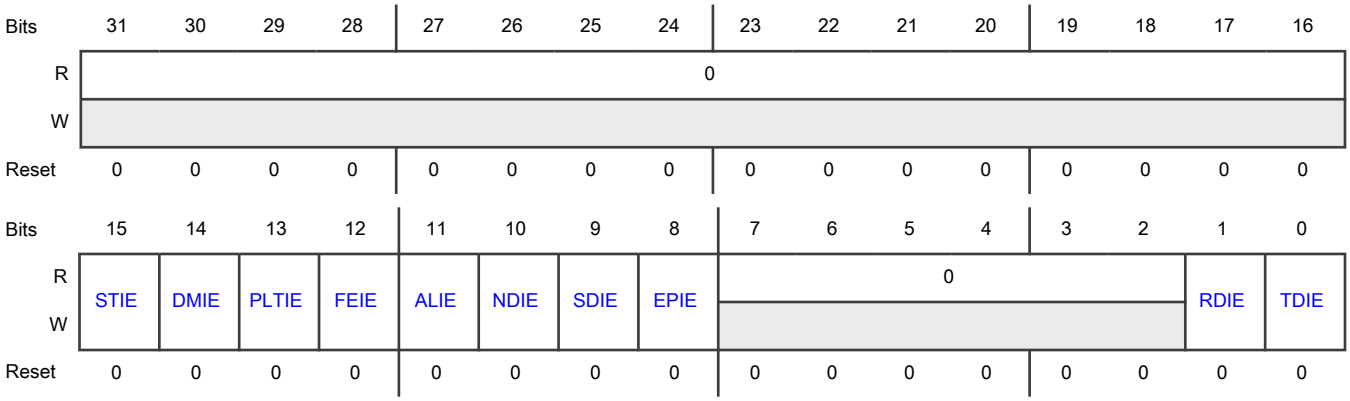
Register	Offset
MIER	18h

Function
Contains enables for:

- Transmit and receive data interrupts

- Start, Stop, and NACK detection interrupts
- DMA interrupts

Diagram



Fields

Field	Function
31-16 —	Reserved
15 STIE	Start Interrupt Enable Enables interrupt for Start condition. 0b - Disable 1b - Enable
14 DMIE	Data Match Interrupt Enable Enables interrupt for data match. 0b - Disable 1b - Enable
13 PLTIE	Pin Low Timeout Interrupt Enable Enables interrupt for pin-low timeout. 0b - Disable 1b - Enable
12 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0b - Disable 1b - Enable
11	Arbitration Lost Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
ALIE	Enables interrupt for arbitration lost. 0b - Disable 1b - Enable
10 NDIE	NACK Detect Interrupt Enable Enables interrupt for NACK detection. 0b - Disable 1b - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0b - Disable 1b - Enable
8 EPIE	End Packet Interrupt Enable Enables interrupt for end packet. 0b - Disable 1b - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0b - Disable 1b - Enable

68.3.6.1.7 Controller DMA Enable (MDER)

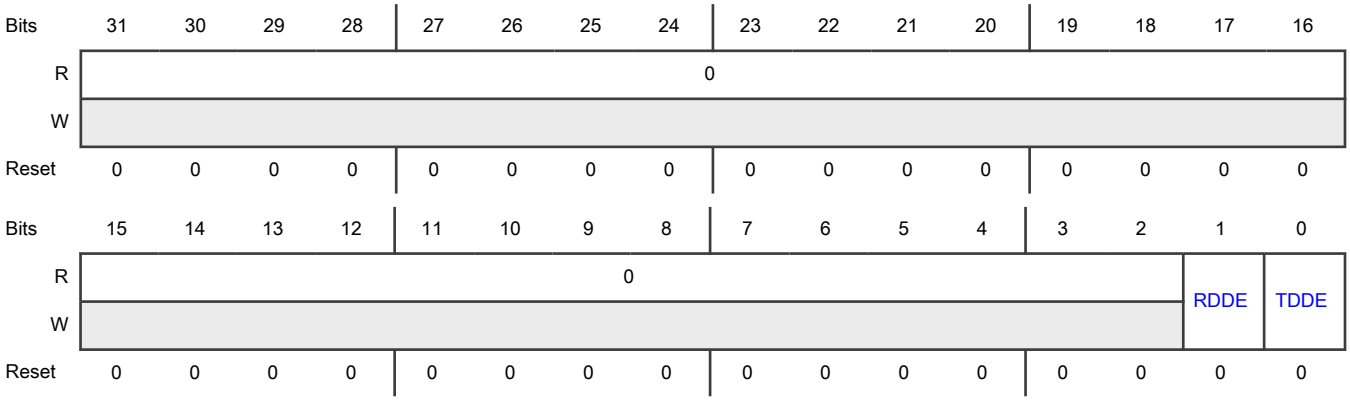
Offset

Register	Offset
MDER	1Ch

Function

Contains DMA transmit, request, and receive enables.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables DMA receive data. 0b - Disable 1b - Enable
0 TDDE	Transmit Data DMA Enable Enables DMA transmit data. 0b - Disable 1b - Enable

68.3.6.1.8 Controller Configuration 0 (MCFGR0)

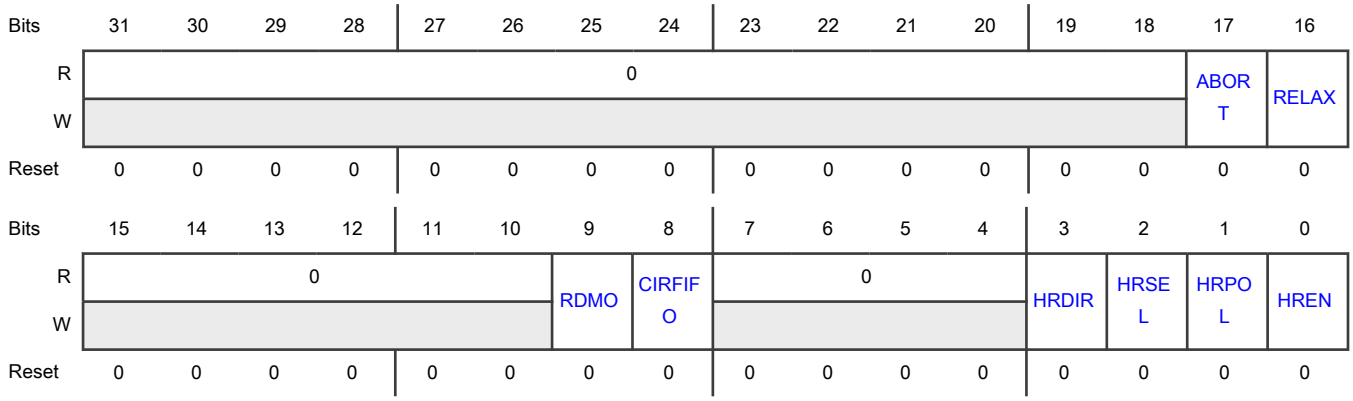
Offset

Register	Offset
MCFGR0	20h

Function

Contains host settings and other receive and transfer settings.

Diagram



Fields

Field	Function
31-18 —	Reserved
17 ABORT	Abort Transfer Determines whether the LPI2C controller completes the existing byte and then sends a Stop condition. A new transfer does not start while this field is 1. 0b - Normal transfer 1b - Abort existing transfer and do not start a new one
16 RELAX	Relaxed Mode Specifies the type of transfer. If this field is 1, the LPI2C controller relaxes several transfer restrictions. Transfers start when the bus is busy and FIFO commands are not checked for errors. You can use this mode to attempt recovery of a stuck I2C target by toggling the SCL pin. 0b - Normal transfer 1b - Relaxed transfer
15-10 —	Reserved
9 RDMO	Receive Data Match Only Determines whether all received data that does not set MSR[DMF] is discarded. After MSR[DMF] is set, the RDMO configuration is ignored. When disabling RDMO, write 0 to this field before writing 0 to MSR[DMF] to ensure that no receive data is lost. 0b - Received data is stored in the receive FIFO 1b - Received data is discarded unless MSR[DMF] is set
8 CIRFIFO	Circular FIFO Enable Enables the transmit FIFO read pointer to be saved to a temporary register. The transmit FIFO empties as normal. After the LPI2C controller is idle and the transmit FIFO is empty, the read pointer value is

Table continues on the next page...

Table continued from the previous page...

Field	Function
	restored from the temporary register. This setting causes the contents of the transmit FIFO to be cycled through repeatedly. If MCFGR1[AUTOSTOP] is 1, then a Stop condition is sent whenever the transmit FIFO is empty and the read pointer is restored. 0b - Disable 1b - Enable
7-4 —	Reserved
3 HRDIR	Host Request Direction Configures the direction of the HREQ pin. When this field is 1, the LPI2C target drives the HREQ output pin and the pin becomes 1 when there is data in the target transmit data register. For proper operation when this field is 1, write 1 to SCFGR1[TXCFG]. 0b - HREQ pin is input (for LPI2C controller) 1b - HREQ pin is output (for LPI2C target)
2 HRSEL	Host Request Select Selects the source of the host request input. When host request is enabled, this field must not change. 0b - Host request input is pin HREQ 1b - Host request input is input trigger
1 HRPOL	Host Request Polarity Configures the polarity of the host request input. When host request is enabled, this field must not change. HRPOL sets the polarity for both the HREQ pin and the input trigger. • When HRPOL=0, the polarity is configured for active low, so host request is asserted if the HREQ pin or input trigger are logic 0. • When HRPOL=1, the polarity is configured for active high, so host request is asserted if the HREQ pin or input trigger are logic 1. 0b - Active low 1b - Active high
0 HREN	Host Request Enable Enables host request. When enabled, the LPI2C controller only initiates a Start condition if the host request input is asserted and the bus is idle. A repeated Start condition is not affected by the host request. 0b - Disable 1b - Enable

68.3.6.1.9 Controller Configuration 1 (MCFGR1)

Offset

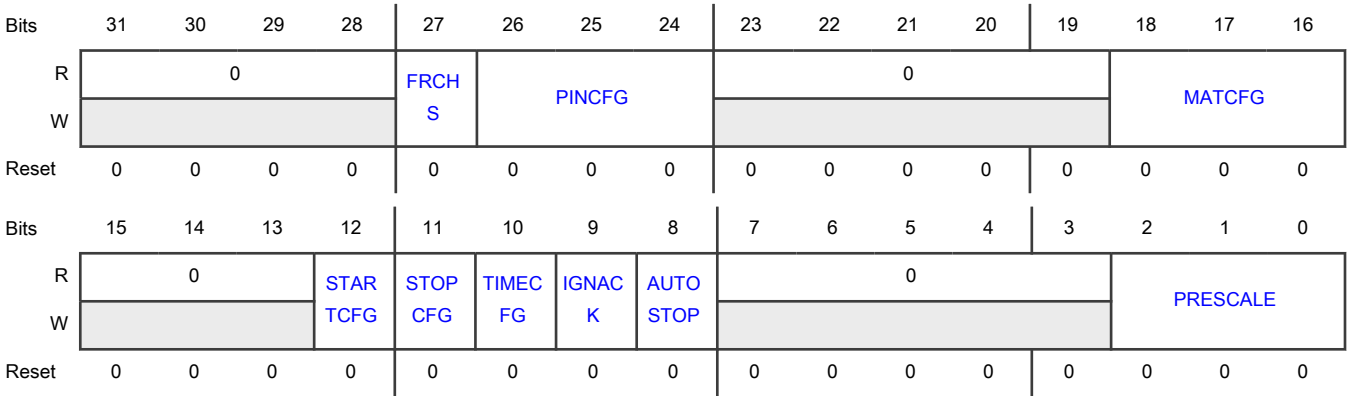
Register	Offset
MCFGR1	24h

Function

Contains controls for pin configuration, clock prescaler, and various other control settings.

Write to this register only when the I2C controller is disabled.

Diagram



Fields

Field	Function
31-28 —	Reserved
27 FRCHS	Force HS Mode Forces the LPI2C pin state into HS mode. Writing 1 to this field does not impact the digital filter configuration or digital timing parameters. 0b - No effect 1b - LPI2C pin state forced into HS mode
26-24 PINCFG	Pin Configuration Configures the pin mode for LPI2C. 000b - Two-pin open drain mode. SCL/SDA pins: Bidirectional open drain for controller and target. SCLS/SDAS pins: Not used. 001b - Two-pin output only mode (Ultra-Fast mode). SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller and target. SCLS/SDAS pins: Not used.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010b - Two-pin push-pull mode. SCL/SDA pins: Bidirectional push-pull for controller and target. SCLS/SDAS pins: Not used. 011b - Four-pin push-pull mode. SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Output-only push-pull for controller and target. 100b - Two-pin open-drain mode with separate LPI2C target. SCL/SDA pins: Bidirectional open drain for controller. SCLS/SDAS pins: Bidirectional open drain for target. 101b - Two-pin output only mode (Ultra-Fast mode) with separate LPI2C target. SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller. SCLS/SDAS pins: Output-only open drain for target. 110b - Two-pin push-pull mode with separate LPI2C target. SCL/SDA pins: Bidirectional push-pull for controller. SCLS/SDAS pins: Bidirectional push-pull for target. 111b - Four-pin push-pull mode (inverted outputs). SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Inverted output-only push-pull for controller and target.
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that sets MSR[DMF]. See Controller Data Match (MDMR). 000b - Match is disabled 001b - Reserved 010b - Match is enabled: first data word equals MDMR[MATCH0] OR MDMR[MATCH1] 011b - Match is enabled: any data word equals MDMR[MATCH0] OR MDMR[MATCH1] 100b - Match is enabled: (first data word equals MDMR[MATCH0]) AND (second data word equals MDMR[MATCH1) 101b - Match is enabled: (any data word equals MDMR[MATCH0]) AND (next data word equals MDMR[MATCH1) 110b - Match is enabled: (first data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1]) 111b - Match is enabled: (any data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1])
15-13 —	Reserved
12 STARTCFG	Start Configuration Configures the conditions that set MSR[STF] when a Start condition occurs. When this field is 0, MSR[STF] is set on a Start condition when both the I2C bus and LPI2C are idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus except the LPI2C controller.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When this field is 1, MSR[STF] is set on a Start condition when the I2C bus is idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus including the LPI2C controller. 0b - Sets when both I2C bus and LPI2C controller are idle 1b - Sets when I2C bus is idle
11 STOPCFG	Stop Configuration Configures the conditions that set MSR[SDF]. When this field is 0, any Stop condition generated by the LPI2C controller sets SDF. When this field is 1, the last Stop condition before the LPI2C controller is idle sets SDF. In this case, the transmit FIFO is empty at the time of the Stop condition. 0b - Any Stop condition 1b - Last Stop condition
10 TIMECFG	Timeout Configuration Configures which signals must be low for longer than the configured timeout to set MSR[PLTF]. When this field is 0, MSR[PLTF] is set when SCL is low for longer than the configured timeout. 0b - SCL 1b - SCL or SDA
9 IGNACK	Ignore NACK Determines whether the LPI2C controller ignores a received NACK and treats it as an ACK. In this case, MSR[NDF] is never set. This field must be 1 in Ultra-Fast mode. 0b - No effect 1b - Treat a received NACK as an ACK
8 AUTOSTOP	Automatic Stop Generation Determines whether a Stop condition is generated when the LPI2C controller is busy and the transmit FIFO is empty. A Stop condition can also be generated using a transmit FIFO command. When this field is 1, a Stop condition is automatically generated when the transmit FIFO is empty and the LPI2C controller is busy. 0b - No effect 1b - Stop automatically generated
7-3 —	Reserved
2-0 PRESCALE	Prescaler Configures the clock prescaler used for all LPI2C controller logic except the digital glitch filters. 000b - Divide by 1

Table continues on the next page...

Table continued from the previous page...

Field	Function
	001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32 110b - Divide by 64 111b - Divide by 128

68.3.6.1.10 Controller Configuration 2 (MCFGR2)

Offset

Register	Offset
MCFGR2	28h

Function

Contains the configuration for the bus idle timeout and glitch filters for SDA and SCL.

Write to this register only when the I2C controller is disabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FILTSDA				0				FILTSCL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BUSIDLE											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
27-24 FILTSDA	Glitch Filter SDA Configures the I2C controller digital glitch filters for the SDA input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C controller digital glitch filters for SCL input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. These cycles are based on the functional clock. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
15-12 —	Reserved
11-0 BUSIDLE	Bus Idle Timeout Configures the bus idle timeout period, in clock cycles. If both SCL and SDA are high for longer than the number of cycles defined by this field, the I2C bus is assumed to be idle and the controller can generate a Start condition. Writing 0 to this field disables the bus idle timeout.

68.3.6.1.11 Controller Configuration 3 (MCFGR3)

Offset

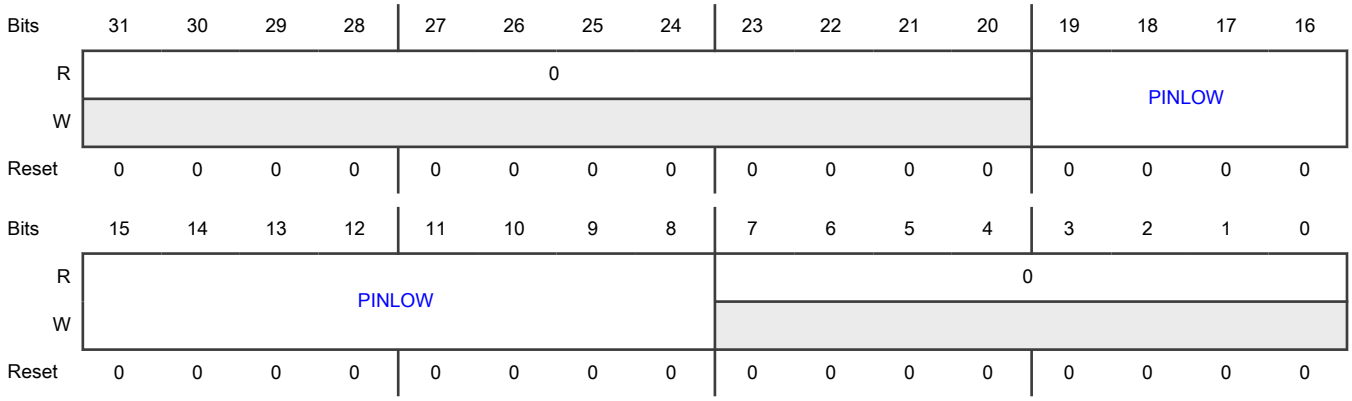
Register	Offset
MCFGR3	2Ch

Function

Configures the threshold value for the pin low timeout flag.

Write to this register only when the I2C controller is disabled.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-8 PINLOW	Pin Low Timeout Configures the threshold value, in clock cycles, that sets MSR[PLTF]. If SCL or SDA (selected by MCFGR1[TIMECFG]) is low for longer than (PINLOW × 256) cycles, MSR[PLTF] is set. When this field is 0, the pin low timeout feature is disabled.
7-0 —	Reserved

68.3.6.1.12 Controller Data Match (MDMR)

Offset

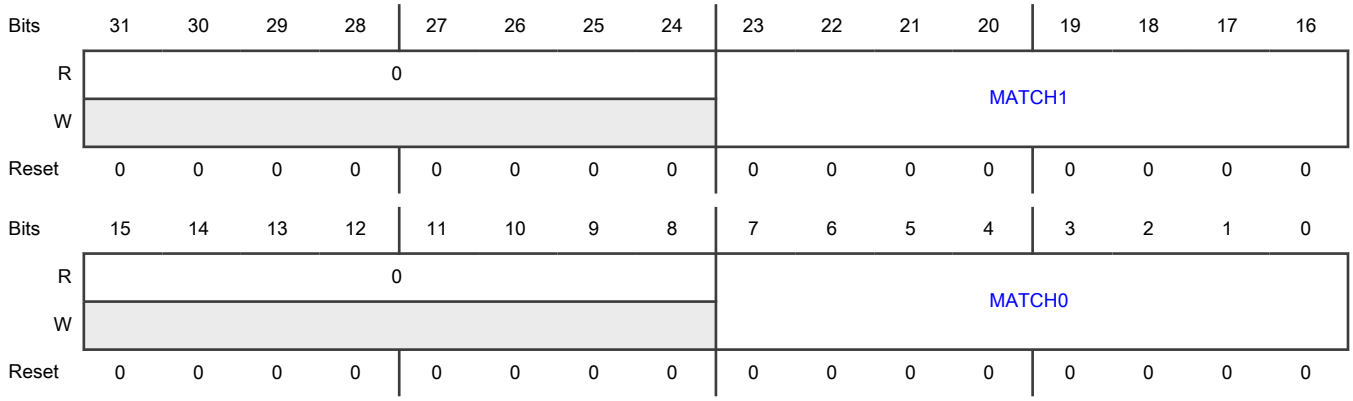
Register	Offset
MDMR	40h

Function

Contains data match values.

Write to this register only when the I2C controller is disabled or idle.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 MATCH1	Match 1 Value Specifies match 1 value that is compared to the received data when receive data match is enabled.
15-8 —	Reserved
7-0 MATCH0	Match 0 Value Specifies match 0 value that is compared to the received data when receive data match is enabled.

68.3.6.1.13 Controller Clock Configuration 0 (MCCR0)

Offset

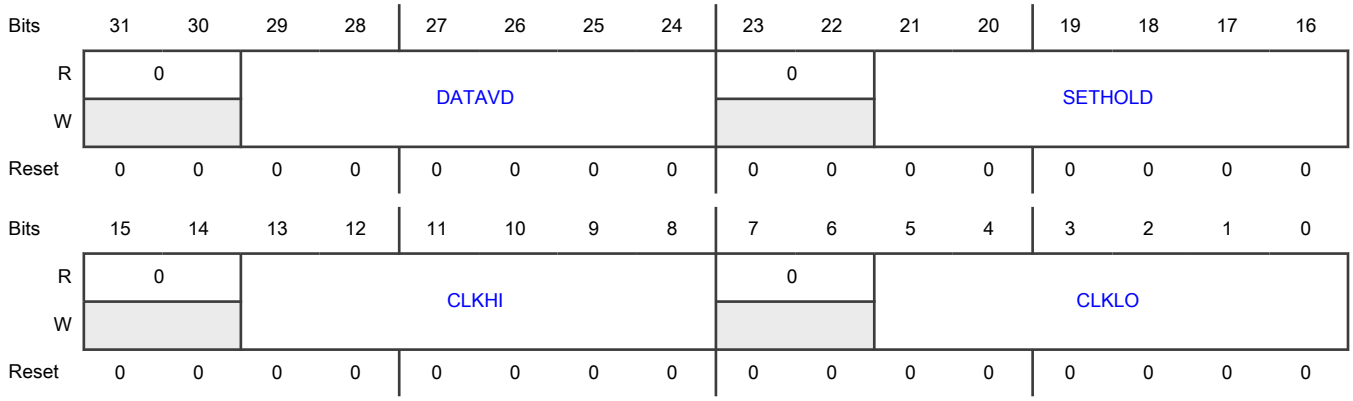
Register	Offset
MCCR0	48h

Function

Configures various clock controls.

You cannot make changes to this register when the I2C controller is enabled and is used for standard, fast, fast-mode plus, and ultra-fast transfers.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start • Setup and hold time for a repeated Start • Setup time for a Stop The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

68.3.6.1.14 Controller Clock Configuration 1 (MCCR1)

Offset

Register	Offset
MCCR1	50h

Function

Configures various clock controls.

You cannot makes changes to this register when the I2C controller is enabled and is used for HS mode transfers. The separate clock configuration for HS mode allows arbitration to take place in Fast mode (with timing configured by Controller Clock Configuration 0 (MCCR0)), before switching to HS mode (with timing configured by MCCR1).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start condition • Setup and hold time for a repeated Start condition • Setup time for a Stop condition The setup time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

68.3.6.1.15 Controller FIFO Control (MFCR)

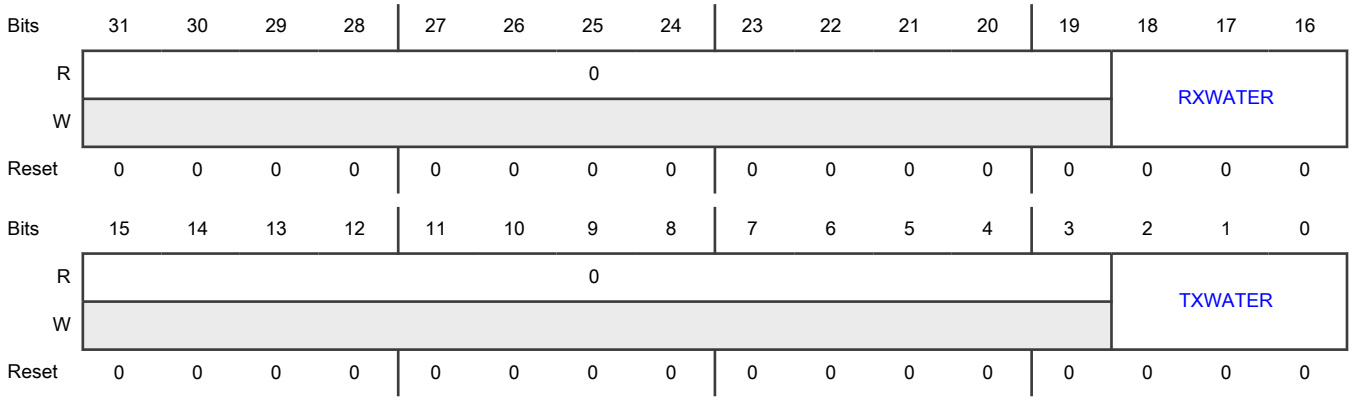
Offset

Register	Offset
MFCR	58h

Function

Controls the receive and transmit FIFO watermark values.

Diagram



Fields

Field	Function
31-19 —	Reserved
18-16 RXWATER	Receive FIFO Watermark Determines the watermark for setting SSR[RDF]. That flag is set when the number of words in the receive FIFO is greater than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.
15-3 —	Reserved
2-0 TXWATER	Transmit FIFO Watermark Determines the watermark for setting SSR[TDF]. That flag is set when the number of words in the transmit FIFO is equal or less than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.

68.3.6.1.16 Controller FIFO Status (MFSR)

Offset

Register	Offset
MFSR	5Ch

Function

Specifies the number of words in the transmit and receive FIFOs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												RXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-16 RXCOUNT	Receive FIFO Count Specifies the number of words in the receive FIFO.
15-4 —	Reserved
3-0 TXCOUNT	Transmit FIFO Count Specifies the number of words in the transmit FIFO.

68.3.6.1.17 Controller Transmit Data (MTDR)

Offset

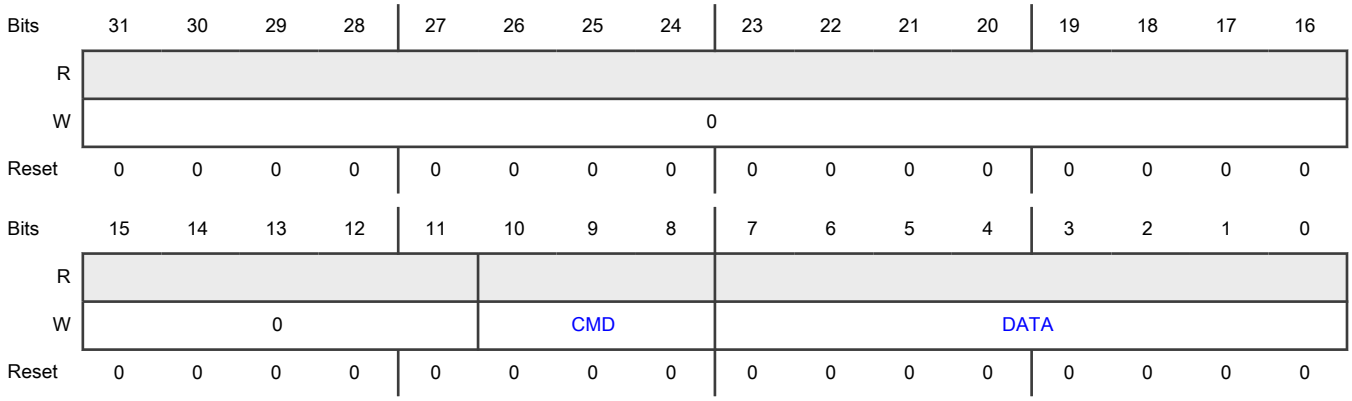
Register	Offset
MTDR	60h

Function

Configures transmit data:

- An 8-bit write to MTDR[CMD] is ignored and does not increment the FIFO write pointer.
- An 8-bit write to MTDR[DATA] zero-extends the value of MTDR[CMD] and increments the FIFO write pointer.
- A 16-bit or 32-bit write operation writes to both MTDR[CMD] and MTDR[DATA] and increments the FIFO write pointer.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-8 CMD	Command Data Selects command transmitted by controller. 000b - Transmit the value in DATA[7:0] 001b - Receive (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes and check each for a data match (if configured) before storing the received data in the receive FIFO. 010b - Generate Stop condition on I2C bus 011b - Receive and discard (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes but do not check for a data match or store those bytes in the receive FIFO. 100b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] 101b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] (this transfer expects a NACK to be returned) 110b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode 111b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode (this transfer expects a NACK to be returned)
7-0 DATA	Transmit Data Contains data used by the commands listed in MTDR[CMD]. Performing an 8-bit write to this field zero-extends the value of MTDR[CMD].

68.3.6.1.18 Controller Receive Data (MRDR)

Offset

Register	Offset
MRDR	70h

Function

Contains the status of the receive FIFO and the data received by the I2C controller that has not been discarded.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RXEMPTY	0						DATA							
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-15 —	Reserved
14 RXEMPTY	Receive Empty Indicates whether the controller receive data FIFO is empty. 0b - Not empty 1b - Empty
13-8 —	Reserved
7-0 DATA	Receive Data Contains data received by the I2C controller that has not been discarded. Received data can be discarded due to the command in MTDR[CMD], or the controller can be configured to discard nonmatching data.

68.3.6.1.19 Controller Receive Data Read Only (MRDROR)

Offset

Register	Offset
MRDROR	78h

Function

Contains the status of the receive FIFO and returns the data received by the I2C controller, but does not pull the data from the FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RXEMPTY	0						DATA							
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-15 —	Reserved
14 RXEMPTY	RX Empty Indicates whether the receive data FIFO is empty. 0b - Not empty 1b - Empty
13-8 —	Reserved
7-0 DATA	Receive Data

68.3.6.1.20 Target Control (SCR)

Offset

Register	Offset
SCR	110h

Function

Contains resets and other target control settings.

NOTE

Each module instance supports a different number of registers.

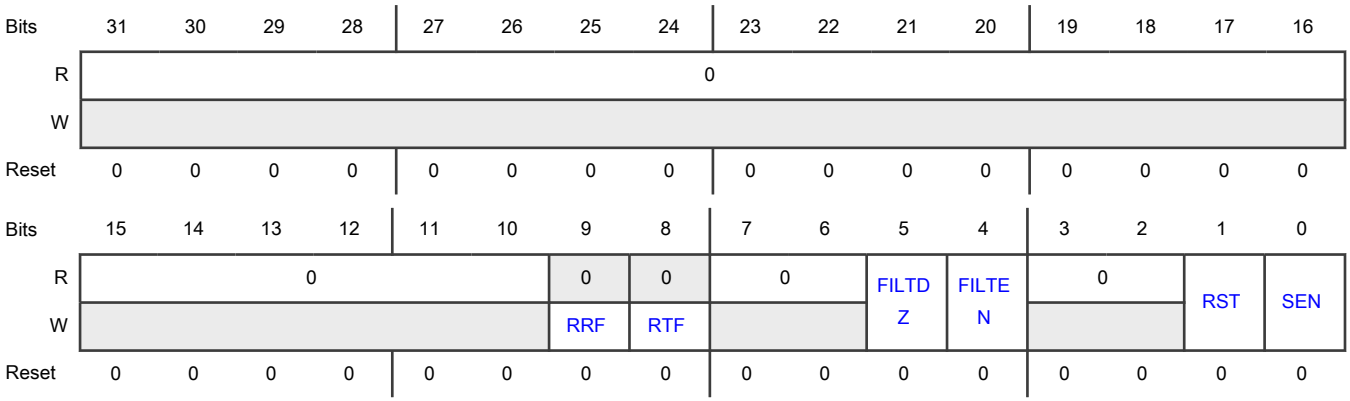
Instance	Register supported	Register not supported
LPI2C0	SCR	—
LPI2C1	SCR	—
LPI2C2	SCR	—
LPI2C3	SCR	—
LPI2C4	SCR	—
LPI2C5	SCR	—
LPI2C6	SCR	—
LPI2C7	SCR	—
LPI2C8	SCR	—
LPI2C9	SCR	—
LPI2C10	SCR	—
LPI2C11	SCR	—
LPI2C12	SCR	—
LPI2C13	SCR	—
LPI2C15	—	SCR
LPI2C17	SCR	—
LPI2C18	SCR	—
LPI2C19	SCR	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C20	SCR	—

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Empties the receive FIFO in Target Receive Data (SRDR). 0b - No effect 1b - SRDR is now empty
8 RTF	Reset Transmit FIFO Empties the transmit FIFO in Target Transmit Data (STDR). 0b - No effect 1b - STDR is now empty
7-6 —	Reserved
5 FILTDZ	Filter Doze Enable Enables filter in Doze mode. Update this field only when the I2C target is disabled. 0b - Enable 1b - Disable
4	Filter Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
FILTEN	Enables digital filter and output delay counter for target mode. Update this field only when the I2C target is disabled. 0b - Disable 1b - Enable
3-2 —	Reserved
1 RST	Software Reset Resets target mode logic. The reset takes effect immediately. The value of this field remains 1 until you write 0 to it. There is no minimum delay required before clearing the software reset. 0b - Not reset 1b - Reset
0 SEN	Target Enable Enables I2C Target mode. 0b - Disable 1b - Enable

68.3.6.1.21 Target Status (SSR)

Offset

Register	Offset
SSR	114h

Function
Contains status flags for transmit and receive data, for error conditions, and for bus and target busy or idle status.

NOTE
Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SSR	—
LPI2C1	SSR	—
LPI2C2	SSR	—
LPI2C3	SSR	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C4	SSR	—
LPI2C5	SSR	—
LPI2C6	SSR	—
LPI2C7	SSR	—
LPI2C8	SSR	—
LPI2C9	SSR	—
LPI2C10	SSR	—
LPI2C11	SSR	—
LPI2C12	SSR	—
LPI2C13	SSR	—
LPI2C15	—	SSR
LPI2C17	SSR	—
LPI2C18	SSR	—
LPI2C19	SSR	—
LPI2C20	SSR	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	SBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SARF	GCF	AM1F	AM0F	FEF	BEF	SDF	RSF	0				TAF	AVF	RDF	TDF
W					W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25 BBF	Bus Busy Flag Indicates whether an I2C bus is idle or busy. 0b - Idle 1b - Busy
24 SBF	Target Busy Flag Indicates whether an I2C target is idle or busy. 0b - Idle 1b - Busy
23-16 —	Reserved
15 SARF	SMBus Alert Response Flag Indicates whether an SMBus alert response has been detected. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - Disabled or not detected 1b - Enabled and detected
14 GCF	General Call Flag Indicates whether a target has detected the general call address. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - General call address disabled or not detected 1b - General call address detected
13 AM1F	Address Match 1 Flag Indicates whether the received address matches the value in ADDR1, or it falls within the ADDR0 to ADDR1 range as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - Matching address not received 1b - Matching address received
12	Address Match 0 Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
AM0F	Indicates whether the received address matches the ADDR0 field, as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - ADDR0 matching address not received 1b - ADDR0 matching address received
11 FEF	FIFO Error Flag Indicates whether there is a FIFO error. This flag can only be set when clock stretching is disabled. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No FIFO error 1b - FIFO error When writing 0b - No effect 1b - Clear the flag
10 BEF	Bit Error Flag Indicates whether the LPI2C target has transmitted a logic 1 and detects a logic 0 on the I2C bus. The target ignores the rest of the transfer until the next (repeated) Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No bit error occurred 1b - Bit error occurred When writing 0b - No effect 1b - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C target detects a Stop condition, and if the LPI2C target matched the last address byte. When SCFGR1[SDCFG] = 1, this flag is set on any Stop condition. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No Stop detected 1b - Stop detected When writing 0b - No effect 1b - Clear the flag
8 RSF	Repeated Start Flag Indicates whether the LPI2C target detects a repeated Start condition and if the LPI2C target matched the last address byte. When SCFGR1[RSCFG] = 1, this flag is set on any repeated Start condition. This flag is not set when the target first detects a Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No repeated Start detected 1b - Repeated Start detected When writing 0b - No effect 1b - Clear the flag
7-4 —	Reserved
3 TAF	Transmit ACK Flag Indicates whether a transmit ACK or NACK is required. You can clear this flag by writing to Target Transmit ACK (STAR). 0b - Not required 1b - Required
2 AVF	Address Valid Flag Indicates whether the contents of Target Address Status (SASR) are valid. You can clear this flag by reading SASR. When SCFGR1[RXCFG] = 1, this flag is also cleared by reading Target Receive Data (SRDR). 0b - Not valid 1b - Valid
1 RDF	Receive Data Flag Indicates whether receive data is ready. You can clear this flag by reading Target Receive Data (SRDR). When SCFGR1[RXCFG] = 1, this flag is not cleared when reading Target Receive Data (SRDR) if SSR[AVF] = 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not ready 1b - Ready
0 TDF	Transmit Data Flag Indicates whether transmit data has been requested. This flag is cleared by writing to Target Transmit Data (STDR). When SCFGR1[TXCFG] = 0, if a NACK, repeated Start, or Stop condition is detected, this flag is also cleared. 0b - Transmit data not requested 1b - Transmit data is requested

68.3.6.1.22 Target Interrupt Enable (SIER)

Offset

Register	Offset
SIER	118h

Function
Contains transmit and receive data interrupt enables, start and stop detect interrupt enables, and other target interrupt enables.

NOTE
Each module instance supports a different number of registers.

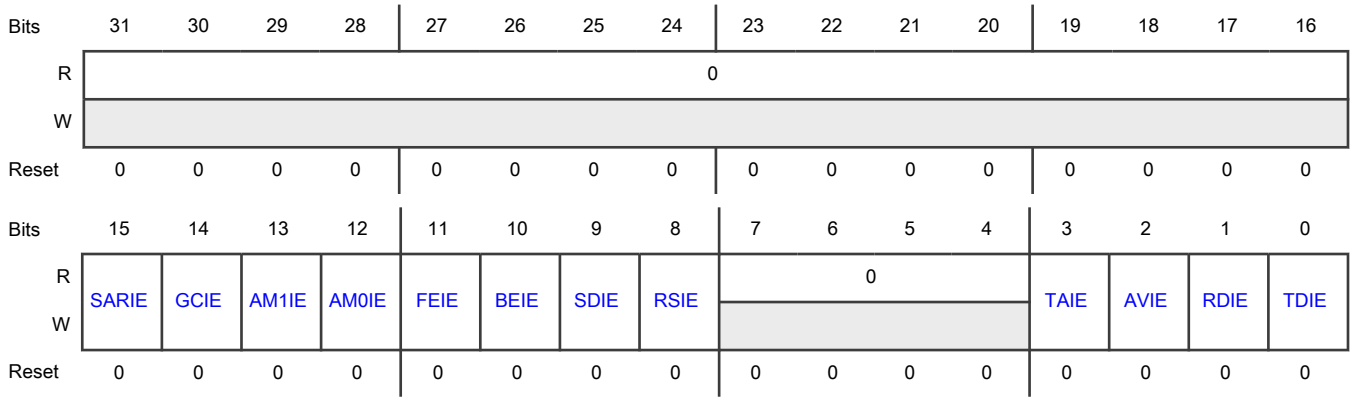
Instance	Register supported	Register not supported
LPI2C0	SIER	—
LPI2C1	SIER	—
LPI2C2	SIER	—
LPI2C3	SIER	—
LPI2C4	SIER	—
LPI2C5	SIER	—
LPI2C6	SIER	—
LPI2C7	SIER	—
LPI2C8	SIER	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C9	SIER	—
LPI2C10	SIER	—
LPI2C11	SIER	—
LPI2C12	SIER	—
LPI2C13	SIER	—
LPI2C15	—	SIER
LPI2C17	SIER	—
LPI2C18	SIER	—
LPI2C19	SIER	—
LPI2C20	SIER	—

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SARIE	SMBus Alert Response Interrupt Enable Enables interrupt for SMBus alert response. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
14 GCIE	General Call Interrupt Enable Enables interrupt for general call. 0b - Disabled 1b - Enabled
13 AM1IE	Address Match 1 Interrupt Enable Enables interrupt for address match 1. 0b - Disable 1b - Enable
12 AM0IE	Address Match 0 Interrupt Enable Enables interrupt for address match 0. 0b - Disable 1b - Enable
11 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0b - Disable 1b - Enable
10 BEIE	Bit Error Interrupt Enable Enables interrupt for bit error. 0b - Disable 1b - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0b - Disable 1b - Enable
8 RSIE	Repeated Start Interrupt Enable Enables interrupt for repeated start. 0b - Disable 1b - Enable
7-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
3 TAIE	Transmit ACK Interrupt Enable Enables interrupt for transmit ACK. 0b - Disable 1b - Enable
2 AVIE	Address Valid Interrupt Enable Enables interrupt for valid address. 0b - Disable 1b - Enable
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0b - Disable 1b - Enable

68.3.6.1.23 Target DMA Enable (SDER)

Offset

Register	Offset
SDER	11Ch

Function
Contains the transmit, request, and receive enables for DMA.

NOTE
Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SDER	—
LPI2C1	SDER	—
LPI2C2	SDER	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C3	SDER	—
LPI2C4	SDER	—
LPI2C5	SDER	—
LPI2C6	SDER	—
LPI2C7	SDER	—
LPI2C8	SDER	—
LPI2C9	SDER	—
LPI2C10	SDER	—
LPI2C11	SDER	—
LPI2C12	SDER	—
LPI2C13	SDER	—
LPI2C15	—	SDER
LPI2C17	SDER	—
LPI2C18	SDER	—
LPI2C19	SDER	—
LPI2C20	SDER	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						SDDE	RSDE	0				AVDE	RDDE	TDDE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-10 —	Reserved
9 SDDE	Stop Detect DMA Enable Enables DMA end-of-packet processing on stop detection. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response. • Returns 0000_40FFh. • Writes 0 to SSR[SDF]. 0b - Disable 1b - Enable
8 RSDE	Repeated Start DMA Enable Enables DMA end-of-packet processing on repeated start. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response. • Returns 0000_40FFh. • Writes 0 to SSR[RDF]. 0b - Disable 1b - Enable
7-3 —	Reserved
2 AVDE	Address Valid DMA Enable Enables address valid DMA request. The address valid DMA request is shared with the receive data DMA request. If both are enabled, write 1 to SCFGR1[RXCFG] to allow the DMA to read the address from Target Receive Data (SRDR). 0b - Disable 1b - Enable
1 RDDE	Receive Data DMA Enable Enables receive data for DMA. 0b - Disable DMA request 1b - Enable DMA request
0 TDDE	Transmit Data DMA Enable Enables transmit data for DMA. 0b - Disable 1b - Enable

68.3.6.1.24 Target Configuration 0 (SCFGR0)

Offset

Register	Offset
SCFGR0	120h

Function
Configures the read request feature.

NOTE
Each module instance supports a different number of registers.

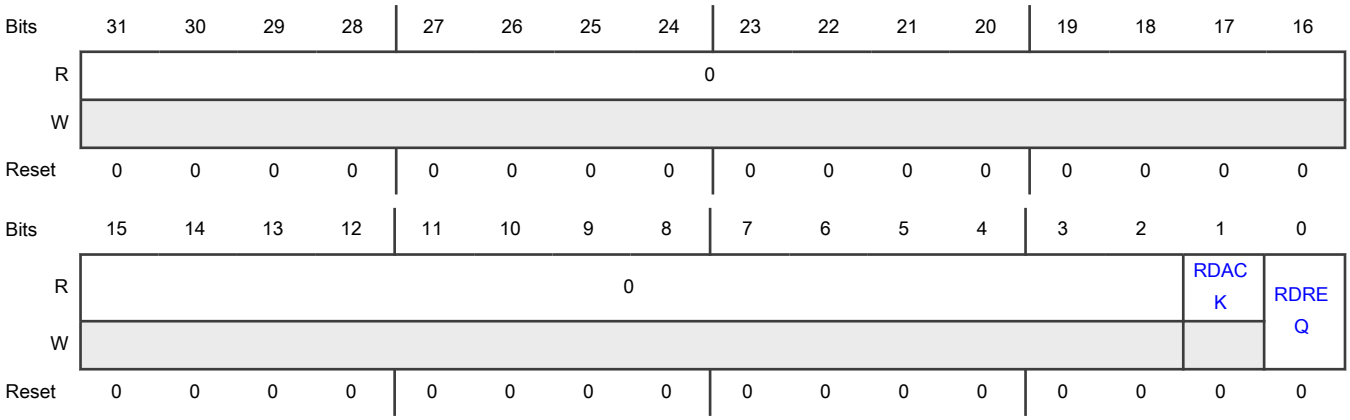
Instance	Register supported	Register not supported
LPI2C0	SCFGR0	—
LPI2C1	SCFGR0	—
LPI2C2	SCFGR0	—
LPI2C3	SCFGR0	—
LPI2C4	SCFGR0	—
LPI2C5	SCFGR0	—
LPI2C6	SCFGR0	—
LPI2C7	SCFGR0	—
LPI2C8	SCFGR0	—
LPI2C9	SCFGR0	—
LPI2C10	SCFGR0	—
LPI2C11	SCFGR0	—
LPI2C12	SCFGR0	—
LPI2C13	SCFGR0	—
LPI2C15	—	SCFGR0
LPI2C17	SCFGR0	—
LPI2C18	SCFGR0	—
LPI2C19	SCFGR0	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C20	SCFGR0	—

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RDACK	Read Acknowledge Flag Indicates whether a Start then Stop sequence with one SCL pulse between them acknowledged the read request while SCFGR0[RDREQ] = 1. You can clear this flag by writing 0 to SCFGR0[RDREQ]. 0b - Read Request not acknowledged 1b - Read Request acknowledged
0 RDREQ	Read Request Enables read request. When the I2C bus is idle, writing 1 to this field causes the LPI2C target to drive SDA low, triggering a Start condition. The LPI2C target releases SDA on the next falling edge of SCL or when you write 0 to this field. To initiate a second read request (for example, following I2C bus activity that did not acknowledge the request), write 0 then 1 to this field. When the I2C bus is busy, writing to this register always writes 0 to this field. 0b - Disable 1b - Enable

68.3.6.1.25 Target Configuration 1 (SCFGR1)

Offset

Register	Offset
SCFGR1	124h

Function

Configures various aspects of the target.

Write to this register only when the I2C target is disabled.

NOTE

Each module instance supports a different number of registers.

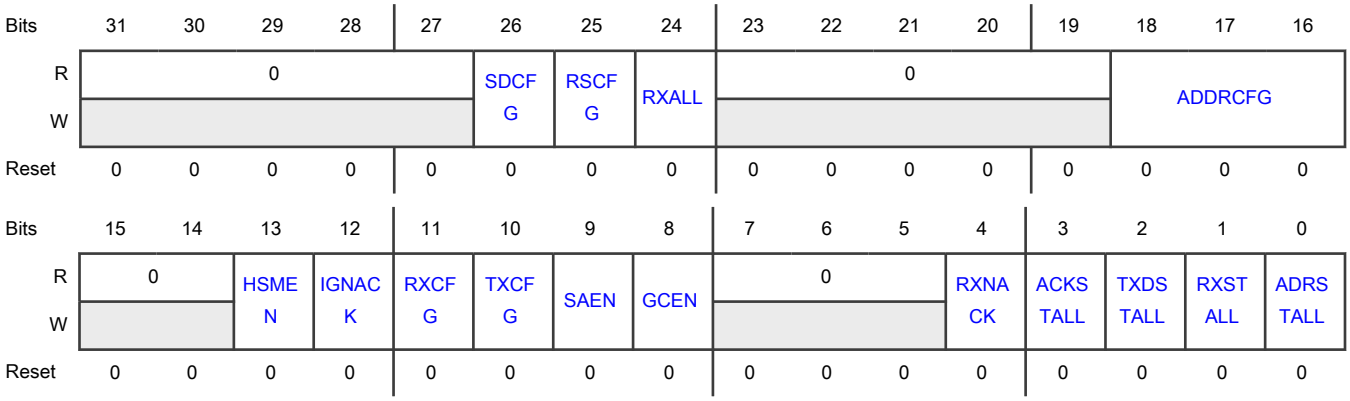
Instance	Register supported	Register not supported
LPI2C0	SCFGR1	—
LPI2C1	SCFGR1	—
LPI2C2	SCFGR1	—
LPI2C3	SCFGR1	—
LPI2C4	SCFGR1	—
LPI2C5	SCFGR1	—
LPI2C6	SCFGR1	—
LPI2C7	SCFGR1	—
LPI2C8	SCFGR1	—
LPI2C9	SCFGR1	—
LPI2C10	SCFGR1	—
LPI2C11	SCFGR1	—
LPI2C12	SCFGR1	—
LPI2C13	SCFGR1	—
LPI2C15	—	SCFGR1
LPI2C17	SCFGR1	—
LPI2C18	SCFGR1	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C19	SCFGR1	—
LPI2C20	SCFGR1	—

Diagram



Fields

Field	Function
31-27 —	Reserved
26 SDCFG	Stop Detect Configuration Configures the conditions that set SSR[SDF]. 0b - Any Stop condition following an address match 1b - Any Stop condition
25 RSCFG	Repeated Start Configuration Configures the conditions that set SSR[RSF]. 0b - Any repeated Start condition following an address match 1b - Any repeated Start condition
24 RXALL	Receive All Enables receive-all functionality. When enabled, the LPI2C target stores all addresses on the I2C bus in Target Address Status (SASR) and all data on the I2C bus in Target Receive Data (SRDR). However, the LPI2C target does not drive SDA (for ACK or target-transmit transfer) unless there is an address match. The LPI2C target can drive SCL if clock stretching is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When this field is 1, the LPI2C target only supports 7-bit addressing modes (you must configure SCFGR1[ADDRCFG] for 7-bit address match). Software can support 10-bit addresses, however. The first byte of a 10-bit address is saved to Target Address Match (SAMR). The second byte is saved as the first data byte in Target Receive Data (SRDR). Use this field to aid debugging of the I2C bus by saving all data on the bus without altering the state of the bus. When this field is 1, it is recommended to also write 1 to SCFGR1[RSCFG] and SCFGR1[SDCFG] so you can track the full state of the I2C bus. 0b - Disable 1b - Enable
23-19 —	Reserved
18-16 ADDRCFG	Address Configuration Configures the condition that causes an address to match. 000b - Address match 0 (7-bit) 001b - Address match 0 (10-bit) 010b - Address match 0 (7-bit) or address match 1 (7-bit) 011b - Address match 0 (10-bit) or address match 1 (10-bit) 100b - Address match 0 (7-bit) or address match 1 (10-bit) 101b - Address match 0 (10-bit) or address match 1 (7-bit) 110b - From address match 0 (7-bit) to address match 1 (7-bit) 111b - From address match 0 (10-bit) to address match 1 (10-bit)
15-14 —	Reserved
13 HSMEN	HS Mode Enable Enables detection of the HS mode controller code of target address 0000_1XX, but does not cause an address match on this code. When this field is 1 and any HS mode controller code is detected, SCR[FILTEN] and SCFGR1[ACKSTALL] are ignored until the next Stop condition is detected. 0b - Disable 1b - Enable
12 IGNACK	Ignore NACK Determines whether the target ends transfer when a NACK condition is detected. When this field is 1, the LPI2C target continues transfers after a NACK is detected. This field is required to be 1 in Ultra-Fast mode. 0b - End transfer on NACK 1b - Do not end transfer on NACK

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 RXCFG	Receive Data Configuration Configures which data is returned and which flags are cleared when reading Target Receive Data (SRDR). When this field is 0, reading SRDR returns received data and clears SSR[RDF]. When this field is 1, reading SRDR: • Returns the value of Target Address Status (SASR) and clears SSR[AVF] when SSR[AVF] is set. • Returns received data and clears SSR[RDF] when SSR[AVF] is not set. 0b - Return received data, clear SSR[RDF] 1b - Return SASR and clear SSR[AVF] when SSR[AVF] is set, return received data and clear SSR[RDF] when SSR[AFV] is not set
10 TXCFG	Transmit Flag Configuration Determines which conditions set SSR[TDF]. This field always becomes 1 before a NACK is detected at the end of a target-transmit transfer. This change can cause an extra word to be written to the transmit data FIFO. When this field is 0, Target Transmit Data (STDR) is automatically emptied when a target-transmit transfer is detected. SSR[TDF] is set when a target-transmit transfer is detected, and SSR[TDF] is cleared at the end of the target-transmit transfer. When this field is 1, SSR[TDF] is set when STDR is empty, and SSR[TDF] is cleared when STDR is full. This setting allows STDR to be filled before a target-transmit transfer is detected. However, it can cause STDR to be written before a NACK is detected on the last byte of a target-transmit transfer. 0b - SSR[TDF] is set only during a target-transmit transfer when STDR is empty 1b - SSR[TDF] is set whenever STDR is empty
9 SAEN	SMBus Alert Enable Enables a match on an SMBus alert. 0b - Disable 1b - Enable
8 GCEN	General Call Enable Enables a general call address. 0b - Disable 1b - Enable
7-5 —	Reserved
4 RXNACK	Receive NACK Determines whether to override the setting of STAR[TXNACK] when the LPI2C receives a matching address during an overrun.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When this field is 1, the LPI2C target responds with a NACK under the following conditions: • SSR[AVF] would be set due to matching an address, but that flag is already 1 (address overrun). • SSR[RDF] would be set due to receiving data, but that flag is already 1 (receive data overrun). 0b - ACK or NACK always determined by STAR[TXNACK] 1b - NACK always generated on address overrun or receive data overrun, otherwise ACK or NACK is determined by STAR[TXNACK]
3 ACKSTALL	ACK SCL Stall Enables SCL clock stretching during target-transmit address bytes and target-receiver address and data bytes, so you can write to Target Transmit ACK (STAR) before the ACK or NACK is transmitted. Clock stretching occurs when transmitting the ninth bit, and is therefore not compatible with HS mode. If this field is 1: • You do not need to write 1 to SCFGR1[RXSTALL] or SCFGR1[ADRSTALL]. • When there is an address match on the first byte of a 10-bit address, SSR[AVF] is set, allowing you to read the received address before writing to Target Transmit ACK (STAR). 0b - Disable 1b - Enable
2 TXDSTALL	Transmit Data SCL Stall Enables SCL clock stretching when SSR[TDF] = 1 during a target-transmit transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable
1 RXSTALL	RX SCL Stall Enables SCL clock stretching when SSR[RDF] = 1 during a target-receive transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable
0 ADRSTALL	Address SCL Stall Enables SCL clock stretching when SSR[AVF] = 1. Clock stretching only occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable

68.3.6.1.26 Target Configuration 2 (SCFGR2)

Offset

Register	Offset
SCFGR2	128h

Function

Configures data valid delay, clock hold time, and glitch filters for SDA and SCL.

Write to this register only when the I2C target is disabled.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SCFGR2	—
LPI2C1	SCFGR2	—
LPI2C2	SCFGR2	—
LPI2C3	SCFGR2	—
LPI2C4	SCFGR2	—
LPI2C5	SCFGR2	—
LPI2C6	SCFGR2	—
LPI2C7	SCFGR2	—
LPI2C8	SCFGR2	—
LPI2C9	SCFGR2	—
LPI2C10	SCFGR2	—
LPI2C11	SCFGR2	—
LPI2C12	SCFGR2	—
LPI2C13	SCFGR2	—
LPI2C15	—	SCFGR2
LPI2C17	SCFGR2	—
LPI2C18	SCFGR2	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C19	SCFGR2	—
LPI2C20	SCFGR2	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FILTSDA				0				FILTSCL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DATAVD						0				CLKHOLD			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C target digital glitch filters for SDA input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C target digital glitch filters for SCL input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
15-14 —	Reserved
13-8 DATAVD	Data Valid Delay Configures the SDA data valid delay time for the I2C target, which is equal to FILTSCL + DATAVD + 3 cycles. The data valid delay must be configured to be less than the minimum SCL low period. MCFGR1[PRESCALE] does not affect the I2C target data valid delay time, and the I2C target data valid delay time is disabled in HS mode.
7-4 —	Reserved
3-0 CLKHOLD	Clock Hold Time Configures the minimum clock hold time for the I2C target, when clock stretching is enabled. The minimum hold time is equal to the number of cycles defined by this field + 3. MCFGR1[PRESCALE] does not affect the I2C target clock hold time, and the I2C target clock hold time is disabled in HS mode.

68.3.6.1.27 Target Address Match (SAMR)

Offset

Register	Offset
SAMR	140h

Function

Contains address values for received target match comparison.

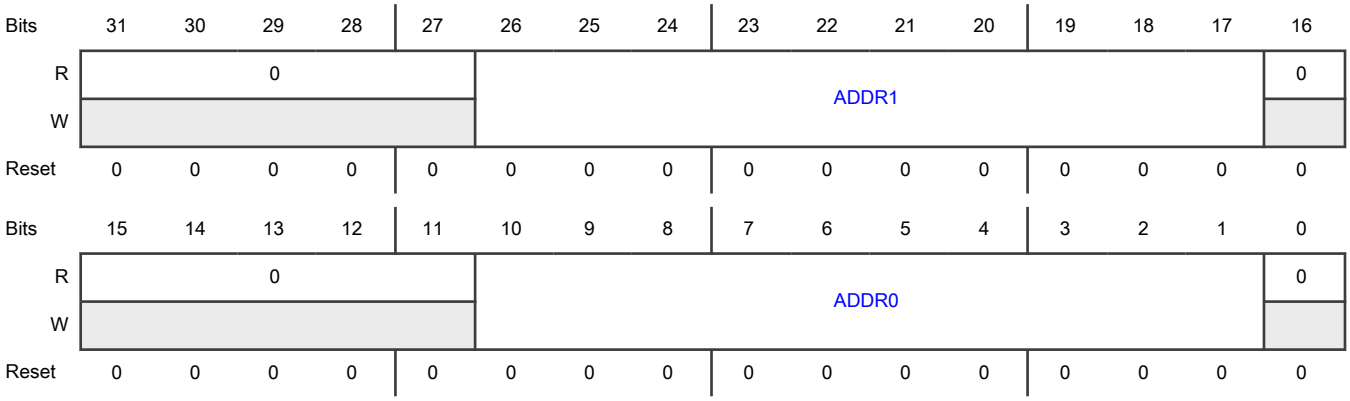
Write to this register only when the I2C target is disabled.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SAMR	—
LPI2C1	SAMR	—
LPI2C2	SAMR	—
LPI2C3	SAMR	—
LPI2C4	SAMR	—
LPI2C5	SAMR	—
LPI2C6	SAMR	—
LPI2C7	SAMR	—
LPI2C8	SAMR	—
LPI2C9	SAMR	—
LPI2C10	SAMR	—
LPI2C11	SAMR	—
LPI2C12	SAMR	—
LPI2C13	SAMR	—
LPI2C15	—	SAMR
LPI2C17	SAMR	—
LPI2C18	SAMR	—
LPI2C19	SAMR	—
LPI2C20	SAMR	—

Diagram



Fields

Field	Function
31-27 —	Reserved
26-17 ADDR1	Address 1 Value Contains the value of address 1, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR1[26:25]} and the second address byte is compared to ADDR1[24:17]. In 7-bit mode, the address is compared to ADDR1[23:17].
16-11 —	Reserved
10-1 ADDR0	Address 0 Value Contains the value of address 0, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR0[10:9]} and the second address byte is compared to ADDR0[8:1]. In 7-bit mode, the address is compared to ADDR0[7:1].
0 —	Reserved

68.3.6.1.28 Target Address Status (SASR)

Offset

Register	Offset
SASR	150h

Function

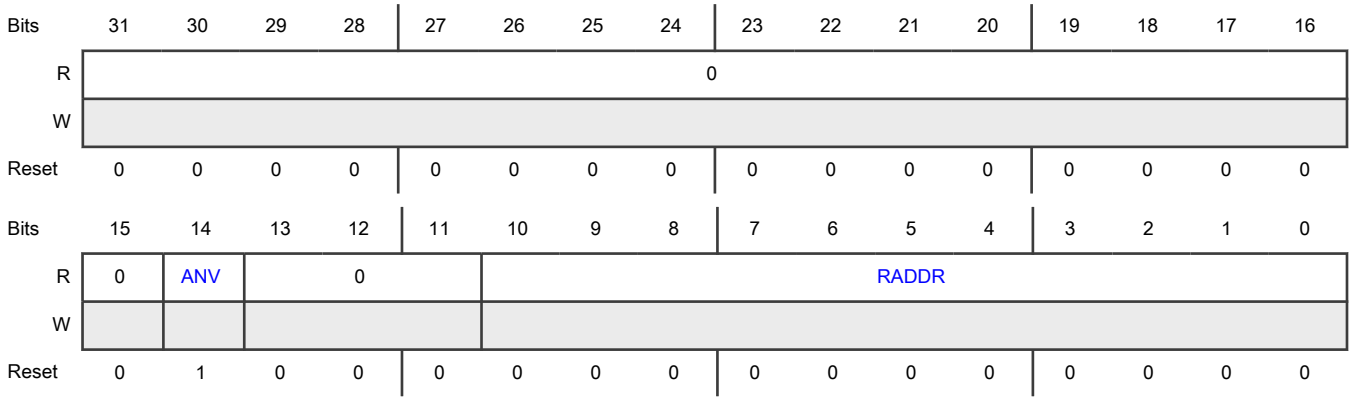
Contains the received address and its validity.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SASR	—
LPI2C1	SASR	—
LPI2C2	SASR	—
LPI2C3	SASR	—
LPI2C4	SASR	—
LPI2C5	SASR	—
LPI2C6	SASR	—
LPI2C7	SASR	—
LPI2C8	SASR	—
LPI2C9	SASR	—
LPI2C10	SASR	—
LPI2C11	SASR	—
LPI2C12	SASR	—
LPI2C13	SASR	—
LPI2C15	—	SASR
LPI2C17	SASR	—
LPI2C18	SASR	—
LPI2C19	SASR	—
LPI2C20	SASR	—

Diagram



Fields

Field	Function
31-15 —	Reserved
14 ANV	Address Not Valid Indicates whether SASR[RADDR] is valid. 0b - Valid 1b - Not valid
13-11 —	Reserved
10-0 RADDR	Received Address Contains the received address. Updates whenever SSR[AM0F] or SSR[AM1F] is set. Reading Target Address Status (SASR) clears SSR[AM0F] and SSR[AM1F]. In 7-bit mode, the address byte is stored in RADDR[7:0]. In 10-bit mode, the first address byte is {11110, RADDR[10:9], RADDR[0]} and the second address byte is RADDR[8:1]. The Read-or-Write bit is therefore always stored in RADDR[0]. When SCFGR1[ACKSTALL] = 1, if the first address byte matches in 10-bit mode, the first address byte is stored in RADDR[7:0] so you can read this field before writing the Transmit ACK. If the second address byte matches, this field is then updated with the full 10-bit address.

68.3.6.1.29 Target Transmit ACK (STAR)

Offset

Register	Offset
STAR	154h

Function

Configures choice of ACK or NACK on each received word.

You can write to this register only when SCFGR1[ACKSTALL] = 1.

SCFGR1[ACKSTALL] enables clock stretching during the ACK-or-NACK bit slot. During this time, you can write to this register.

The logic ensures that the clock stretching continues for at least one bus clock cycle after this register is updated.

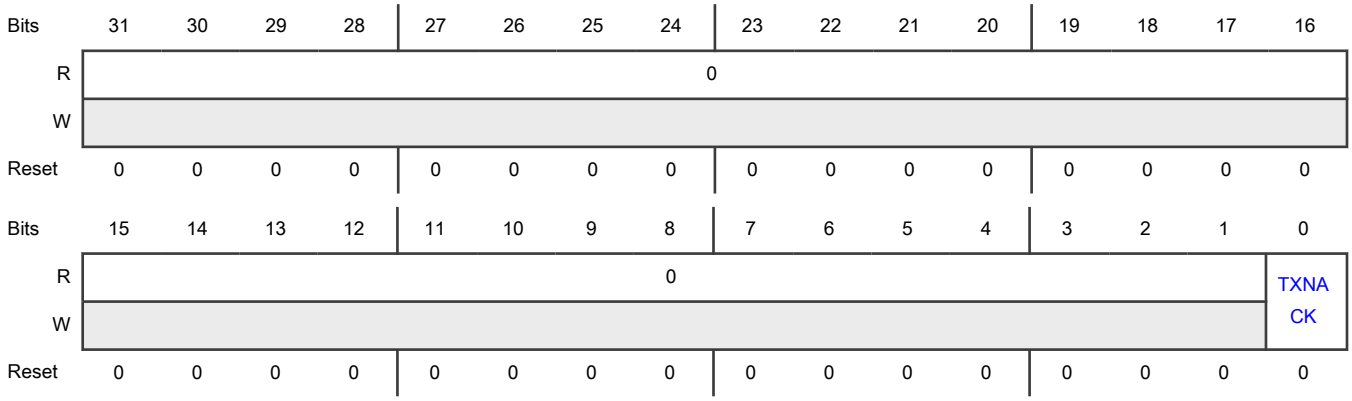
This clock stretching time can be extended via SCFGR2[CLKHOLD].

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	STAR	—
LPI2C1	STAR	—
LPI2C2	STAR	—
LPI2C3	STAR	—
LPI2C4	STAR	—
LPI2C5	STAR	—
LPI2C6	STAR	—
LPI2C7	STAR	—
LPI2C8	STAR	—
LPI2C9	STAR	—
LPI2C10	STAR	—
LPI2C11	STAR	—
LPI2C12	STAR	—
LPI2C13	STAR	—
LPI2C15	—	STAR
LPI2C17	STAR	—
LPI2C18	STAR	—
LPI2C19	STAR	—
LPI2C20	STAR	—

Diagram



Fields

Field	Function
31-1 —	Reserved
0 TXNACK	Transmit NACK Selects whether transmit ACK (logic 0) or NACK (logic 1) is returned on the bus by the I2C target after receiving each word. • When SCFGR1[ACKSTALL] = 1, a transmit NACK signal must be written once for each matching address byte and each received word. SCFGR1[ACKSTALL] must be 1, because that setting stalls the data transfer until software reads the received word (and determines whether to respond with an ACK or NACK). • To configure the default (ACK or NACK), you can write to this field when LPI2C target is disabled or idle. 0b - Transmit ACK 1b - Transmit NACK

68.3.6.1.30 Target Transmit Data (STDR)

Offset

Register	Offset
STDR	160h

Function

Contains the I2C target data to transmit.

Clock stretching (enabled or disabled) affects when the transmit data is transferred. SCFGR1[TXDSTALL] enables clock stretching during the first data bit of a target-transmit transfer.

If clock stretching is enabled (SCFGR1[TXDSTALL] = 1), the transmit data transfer is stalled until this register is updated. Clock stretching is extended by at least 1 bus clock cycle after this register is updated. Clock stretching can be delayed further by using SCFGR2[CLKHOLD].

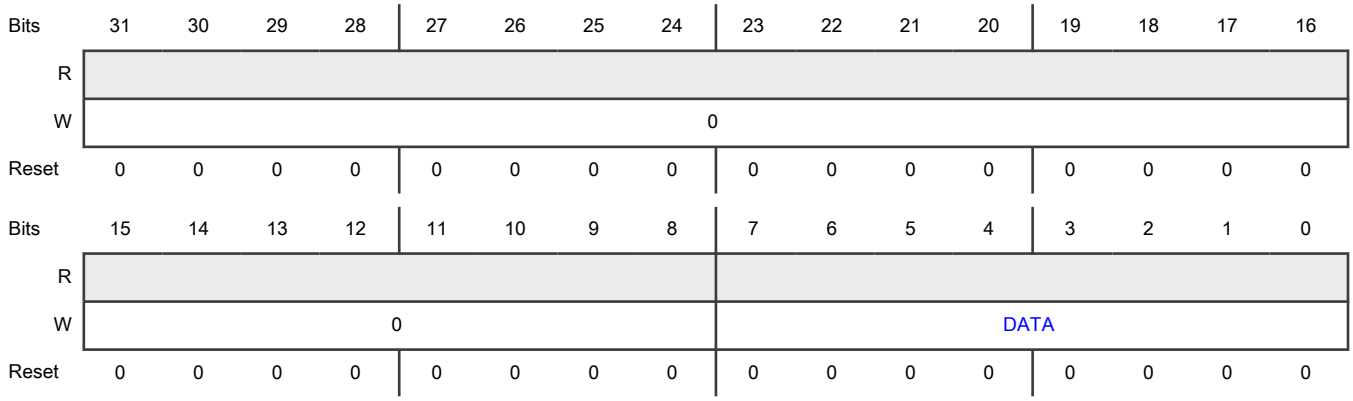
If clock stretching is disabled (SCFGR1[TXDSTALL] = 0), the transmit data must be written before the start of the target-transmit transfer, otherwise SSR[FEF] is set.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	STDR	—
LPI2C1	STDR	—
LPI2C2	STDR	—
LPI2C3	STDR	—
LPI2C4	STDR	—
LPI2C5	STDR	—
LPI2C6	STDR	—
LPI2C7	STDR	—
LPI2C8	STDR	—
LPI2C9	STDR	—
LPI2C10	STDR	—
LPI2C11	STDR	—
LPI2C12	STDR	—
LPI2C13	STDR	—
LPI2C15	—	STDR
LPI2C17	STDR	—
LPI2C18	STDR	—
LPI2C19	STDR	—
LPI2C20	STDR	—

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 DATA	Transmit Data Contains the I2C target data to transmit. Writing data to this register stores I2C target transmit data in this register.

68.3.6.1.31 Target Receive Data (SRDR)

Offset

Register	Offset
SRDR	170h

Function

Contains status of target receive data transfer.

NOTE

Each module instance supports a different number of registers.

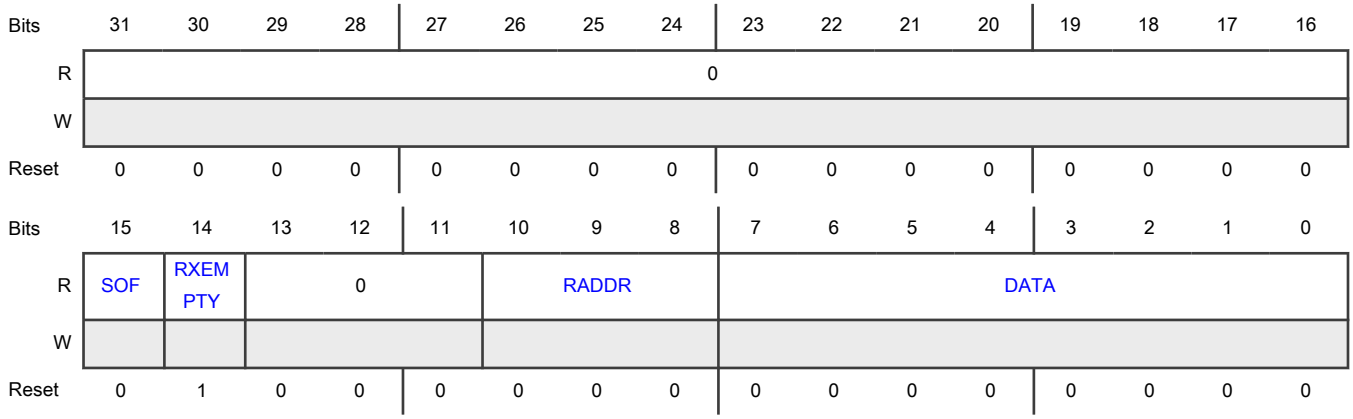
Instance	Register supported	Register not supported
LPI2C0	SRDR	—
LPI2C1	SRDR	—
LPI2C2	SRDR	—
LPI2C3	SRDR	—

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C4	SRDR	—
LPI2C5	SRDR	—
LPI2C6	SRDR	—
LPI2C7	SRDR	—
LPI2C8	SRDR	—
LPI2C9	SRDR	—
LPI2C10	SRDR	—
LPI2C11	SRDR	—
LPI2C12	SRDR	—
LPI2C13	SRDR	—
LPI2C15	—	SRDR
LPI2C17	SRDR	—
LPI2C18	SRDR	—
LPI2C19	SRDR	—
LPI2C20	SRDR	—

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition. 0b - Not first 1b - First
14 RXEMPTY	Receive Empty Indicates whether this register is empty. 0b - Not empty 1b - Empty
13-11 —	Reserved
10-8 RADDR	Received Address Contains the address received by the IC2 target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0 DATA	Received Data Contains the data received by the I2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

68.3.6.1.32 Target Receive Data Read Only (SRDROR)

Offset

Register	Offset
SRDROR	178h

Function

Contains the data received by the I2C target.

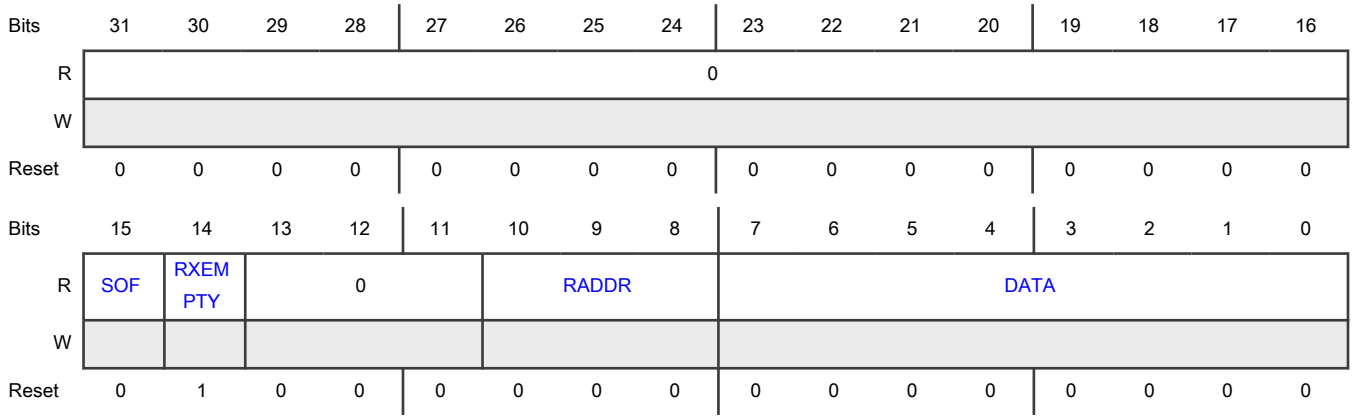
Reading this register returns the data received by the I2C target, but does not pull the data from the register.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
LPI2C0	SRDROR	—
LPI2C1	SRDROR	—
LPI2C2	SRDROR	—
LPI2C3	SRDROR	—
LPI2C4	SRDROR	—
LPI2C5	SRDROR	—
LPI2C6	SRDROR	—
LPI2C7	SRDROR	—
LPI2C8	SRDROR	—
LPI2C9	SRDROR	—
LPI2C10	SRDROR	—
LPI2C11	SRDROR	—
LPI2C12	SRDROR	—
LPI2C13	SRDROR	—
LPI2C15	—	SRDROR
LPI2C17	SRDROR	—
LPI2C18	SRDROR	—
LPI2C19	SRDROR	—
LPI2C20	SRDROR	—

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition. 0b - Not the first 1b - First
14 RXEMPTY	Receive Empty Indicates whether Target Receive Data (SRDR) is empty. 0b - Not empty 1b - Empty
13-11 —	Reserved
10-8 RADDR	Received Address Contains address received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0 DATA	Receive Data Contains data received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

68.3.6.1.33 Controller Transmit Command Burst (MTCBR0 - MTCBR127)

Offset

For n = 0 to 127:

Register	Offset
MTCBRn	200h + (n × 4h)

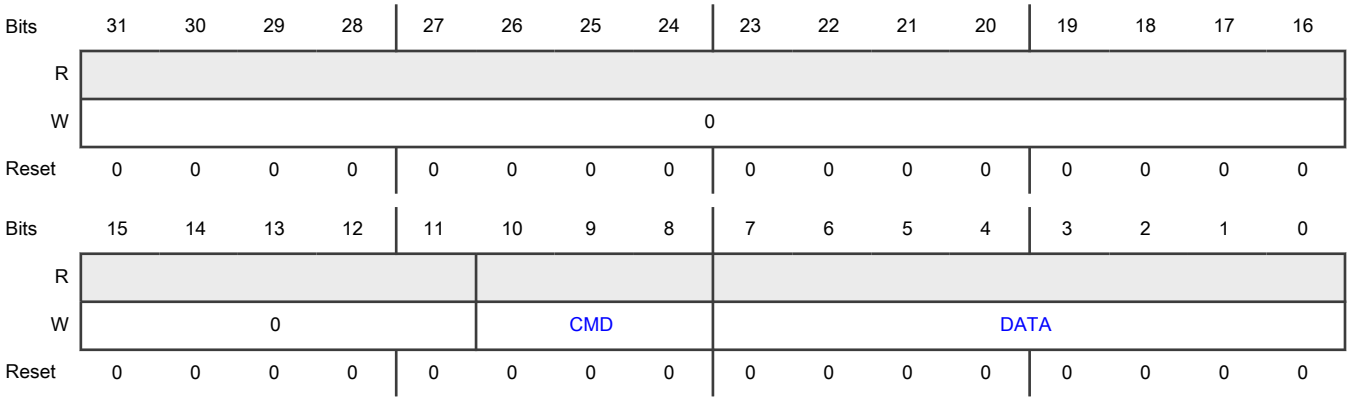
Function

Contains transmit Command and Data fields to support burst transfers.

Is an alias of the Controller Transmit Data Register designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using aligned 8-bit, 16-bit, or 32-bit writes. The size of the Controller Transmit Data Burst Register is 512-bytes.

- An aligned 32-bit write in this region pushes one entry into the transmit FIFO.
- An aligned 16-bit write in this region to MTCBRn[15:0] pushes one entry into the transmit FIFO.
- An 8-bit write in this region to MTCBRn[7:0] updates DATA[7:0], but does not push the data into the transmit FIFO.
- An 8-bit write in this region to MTCBRn[15:8] pushes the data written to CMD[2:0] plus the previously written DATA[7:0] into the transmit FIFO.
- An 8-bit or 16-bit write in this region to MTXBRn[31:16] is ignored.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-8 CMD	Command Command is written to the Controller Transmit Data Register.
7-0 DATA	Data Data is written to the Controller Transmit Data Register.

68.3.6.1.34 Transmit Data Burst (MTDBR0 - MTDBR252)

Offset

For n = 0 to 252:

Register	Offset
MTDBRn	400h + (n × 4h)

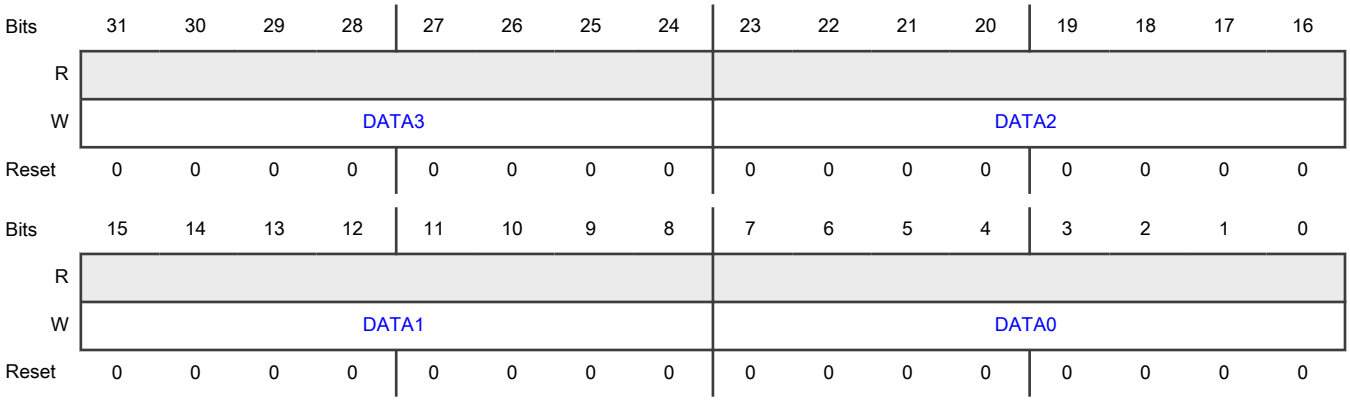
Function

Contains transmit Data fields to support burst transfers.

Is an alias of the Transmit Data Register designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using 8-bit, 16-bit, or 32-bit writes. The CMD field is always zero-extended for transmit data. The size of the Transmit Data Burst Register is 1012 bytes.

- An aligned 32-bit write in this region pushes four zero-extended DATA entries into the transmit FIFO (DATA0 byte first). Note that the register access is extended by 3 wait states.
- An aligned 16-bit write in this region to either half of a 32-bit word pushes two zero-extended DATA entries into the transmit FIFO (DATA0 or DATA2 first). Note that the register access is extended by 1 wait state.
- An 8-bit write in this region pushes one zero-extended DATA entry into the transmit FIFO.

Diagram



Fields

Field	Function
31-24 DATA3	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
23-16 DATA2	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
15-8 DATA1	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 DATA0	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.

68.3.6.1.35 Transmit Data Burst (MTDBR253 - MTDBR255)

Offset

Register	Offset
MTDBR253	7F4h
MTDBR254	7F8h
MTDBR255	7FCh

Function

Contains transmit Data fields to support burst transfers.

Is an alias of the Transmit Data Register designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using 8-bit, 16-bit, or 32-bit writes. The CMD field is always zero-extended for transmit data. The size of the Transmit Data Burst Register is 1024 bytes.

- An aligned 32-bit write in this region pushes four zero-extended DATA entries into the transmit FIFO (DATA0 byte first). Note that the register access is extended by 3 wait states.
- An aligned 16-bit write in this region to either half of a 32-bit word pushes two zero-extended DATA entries into the transmit FIFO (DATA0 or DATA2 first). Note that the register access is extended by 1 wait state.
- An 8-bit write in this region pushes one zero-extended DATA entry into the transmit FIFO.

NOTE

Each module instance supports a different number of registers.

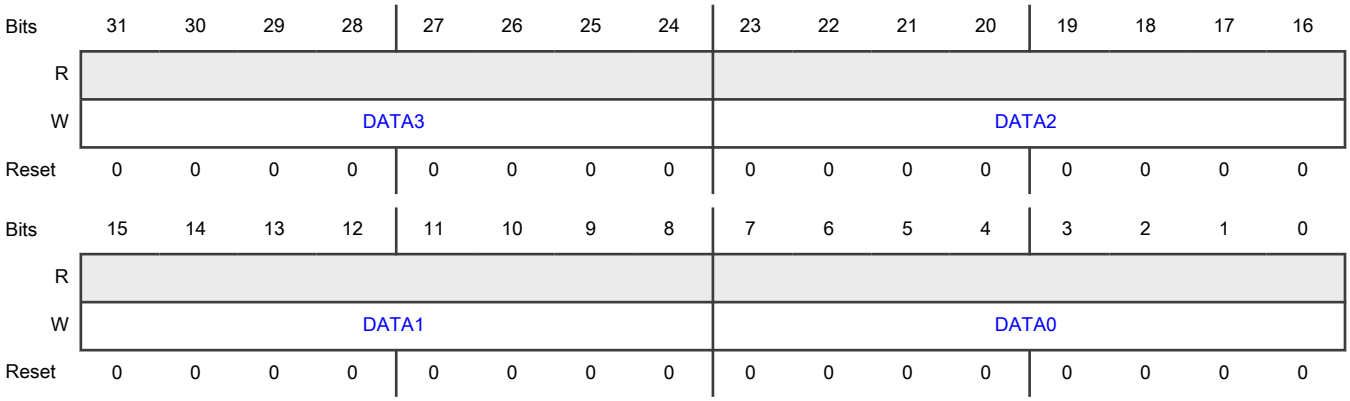
Instance	Register supported	Register not supported
LPI2C0	—	MTDBR253–MTDBR255
LPI2C1	—	MTDBR253–MTDBR255
LPI2C2	—	MTDBR253–MTDBR255
LPI2C3	—	MTDBR253–MTDBR255
LPI2C4	—	MTDBR253–MTDBR255
LPI2C5	—	MTDBR253–MTDBR255
LPI2C6	—	MTDBR253–MTDBR255

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
LPI2C7	—	MTDBR253–MTDBR255
LPI2C8	—	MTDBR253–MTDBR255
LPI2C9	—	MTDBR253–MTDBR255
LPI2C10	—	MTDBR253–MTDBR255
LPI2C11	—	MTDBR253–MTDBR255
LPI2C12	—	MTDBR253–MTDBR255
LPI2C13	—	MTDBR253–MTDBR255
LPI2C15	MTDBR253–MTDBR255	—
LPI2C17	—	MTDBR253–MTDBR255
LPI2C18	—	MTDBR253–MTDBR255
LPI2C19	—	MTDBR253–MTDBR255
LPI2C20	—	MTDBR253–MTDBR255

Diagram



Fields

Field	Function
31-24 DATA3	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
23-16	Data

Table continues on the next page...

Table continued from the previous page...

Field	Function
DATA2	Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
15-8 DATA1	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
7-0 DATA0	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.

68.4 Low Power Serial Peripheral Interface (LPSPI)

68.4.1 Overview

LPSPI provides an efficient interface (either as a controller or peripheral) to an SPI bus, which is a synchronous serial communication interface used in embedded systems. It is typically used to perform short distance communications between microcontrollers and peripheral devices, on printed circuit boards. Typical applications include interfacing with secure digital cards and LCD displays.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 652. Updated terms

Updated term	Deprecated term
Controller	Master
Peripheral	Slave

68.4.1.1 Block diagram

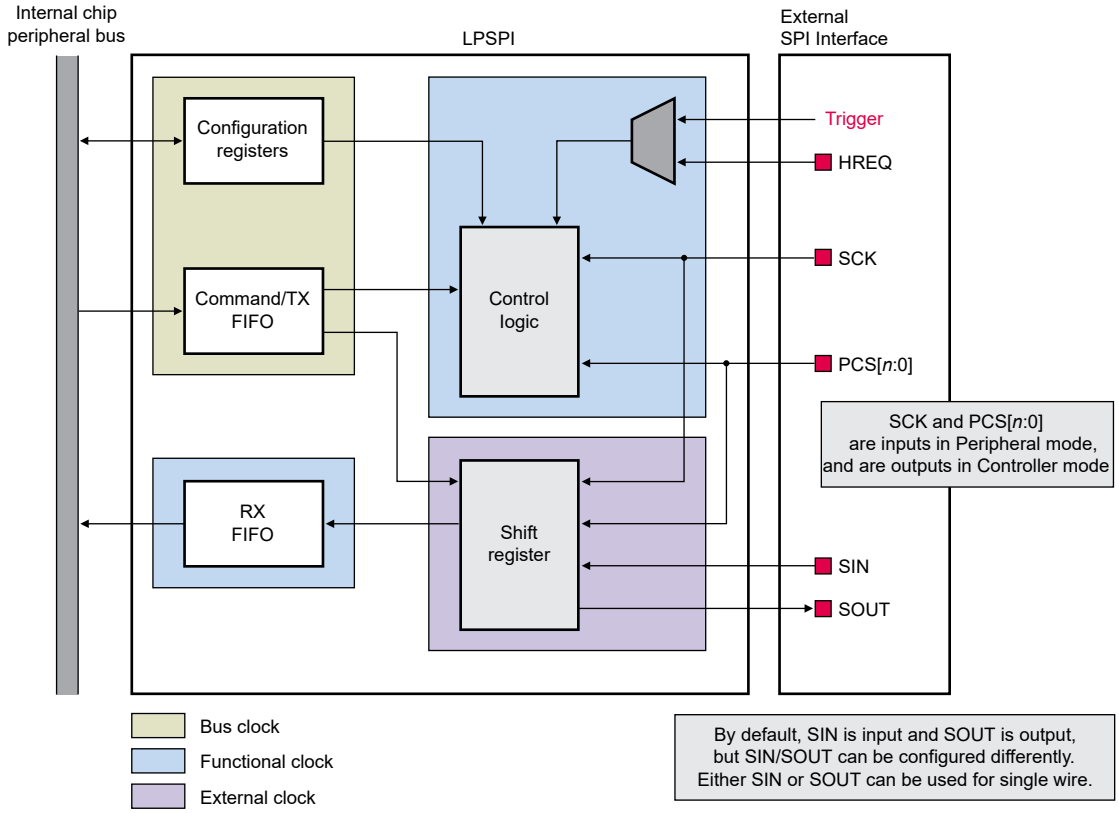


Figure 285. Block diagram

68.4.1.2 Features

- Minimal CPU overhead, with DMA transmit and receive requests supporting FIFO register accesses
- Operation continues in Stop mode, if configured to do so and an appropriate clock is available
- Support available for 32-bit word size
- Configurable clock polarity and phase
- Support available for 4 peripheral chip selects in Controller mode
- Support available for Peripheral mode
- 8-word transmit and command FIFO
- 8-word receive FIFO
- Flexible timing parameters in Controller mode, including SCK frequency and duty cycle, and delays between PCS and SCK edges
- Continuous transfer option to keep PCS asserted across multiple frames
- Full-duplex transfers that support 1-bit transmit and receive on each clock edge
- Half-duplex transfers that support:
  - 1-bit transmit or receive on each clock edge
  - 2-bit transmit or receive on each clock edge
  - 4-bit transmit or receive on each clock edge

- Option to use host request to control the start of an SPI bus transfer
- Receive data match logic that discards nonmatching data and interrupt on data match

68.4.2 Functional description

68.4.2.1 Controller mode

68.4.2.1.1 Transmit and command FIFO commands

The transmit and command FIFO is a combined FIFO that includes both transmit data words and command words. You store:

- Transmit data words in the transmit and command FIFO, by writing to Transmit Data (TDR).
- Command words in the transmit and command FIFO, by writing to Transmit Command (TCR).

When a command word is at the top of the transmit and command FIFO, the actions that can occur depend on whether LPSPI is busy or between frames (see TCR[CONT] and TCR[CONTC]). See Table 653 for conditions and possible corresponding actions when a command word is at the top of the transmit and command FIFO.

Table 653. Possible actions when a command word is at the top of the transmit and command FIFO

Condition	Action
LPSPI is enabled and idle.	The command word is pulled from the FIFO, and this command word controls all subsequent transfers.
LPSPI is busy and TCR[CONTC] is 0.	The SPI frame completes at the end of the existing word, ignoring TCR[FRAMESZ]. The command word is then pulled from the FIFO and that command word controls all subsequent transfers (or until the next update to the command word). Note that a command word with TCR[CONTC] = 0 always terminates the existing transfer regardless of the previous TCR[CONT] value.
LPSPI is busy; the existing TCR[CONT] value is 1 and the new TCR[CONTC] value is 1.	The command word must be updated at the frame boundary. The command word is pulled from the FIFO during the last SCK pulse of the existing frame (based on the value of FRAMESZ), and the frame continues using the new command value for the rest of the frame (or until the next update to the command word). When TCR[CONTC] = 1, only the lower 24 bits of the command word are updated. If the command word is updated at a word boundary, then the transfer halts (stops) after that word. TCR[CONTC] is ignored when not at a frame boundary, so the frame ends prematurely.

TCR[CONT] = 1 keeps PCS asserted at end of frame, allowing the transfer to continue.

TCR[CONTC] = 1 specifies that this command word must not terminate the existing frame, and the transfer can continue using the new command word.

TCR[CONTC] = 1 is restricted in the sense that the new command must load on a frame boundary, and the only way for a transfer to continue from a frame boundary is when the previous command has TCR[CONT] = 1.

You can read the current state of the existing command word from Transmit Command (TCR). It requires at least three LPSPI functional clock cycles for Transmit Command (TCR) to update after you write to it (assuming an empty FIFO), and LPSPI must be enabled (CR[MEN] = 1).

Writing to Transmit Command (TCR) does not initiate an SPI bus transfer, unless TCR[TXMSK] = 1. When TCR[TXMSK] = 1, a new command word is not loaded until the end of the existing frame (based on the value of TCR[FRAMESZ]); at the end of the transfer, TCR[TXMSK] transitions to 0.

In Controller mode, the LPSPI command word in Transmit Command (TCR) controls SPI attributes based on the selections in register fields. See Table 654 for TCR fields and associated functionality related to data transfer.

Table 654. Command word in Controller mode

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
CPOL	Clock polarity	Specifies the polarity of the SCK pin. Any change of CPOL value causes a transition on the SCK pin.	N
CPHA	Clock phase	Specifies the clock phase of the transfer.	N
PRESCALE	Prescaler value	Specifies a prescaler used to divide the LPSPI functional clock, to generate the timing parameters of the SPI bus transfer. Changing PRESCALE in conjunction with PCS enables LPSPI to connect to different peripheral devices at different frequencies.	N
PCS	Peripheral chip select	Specifies which PCS pin asserts for the transfer; the polarity of PCS is static and specified by CFGR1[PCSPOL]. If CFGR1[PCSCFG] = 1, do not select PCS[3:2].	N
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.	Y
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.	Y
CONT	Continuous transfer	Configures LPSPI for a continuous transfer that keeps PCS asserted between frames (as specified by FRAMESZ). You must write a new command word to cause PCS to negate. Also, this field supports changing the command word at frame size boundaries.	Y
CONTC	Continuing command	Indicates that this is a new command word for the existing continuous transfer. When CONTC = 1, the command word must only be written to the transmit and command FIFO on a frame boundary.	Y
RXMSK	Receive data mask	Masks the receive data and does not store the masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.	Y
TXMSK	Transmit data mask	Masks the transmit data; masked transmit data is not pulled from the transmit FIFO, and the output data pin is 3-stated (unless otherwise configured by CFGR1[OUTCFG]). This option is useful for half-duplex transfers.	Y
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse: • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. • 2-bit and 4-bit half-duplex transfers are useful for interfacing with QuadSPI memory devices, and either TXMSK or RXMSK must also be 1.	Y
FRAMESZ	Frame size	Configures the frame size in number of bits equal to (FRAMESZ + 1):	Y

Table continues on the next page...

Table 654. Command word in Controller mode (continued)

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
		• The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remaining bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.	

68.4.2.1.1.1 SPI bus transfers

LPSPI initiates an SPI bus transfer when all these conditions are true:

- Data is written to the transmit FIFO.
- The HREQ pin is asserted (or the HREQ function is disabled).
- LPSPI is enabled.

To perform the SPI bus transfer, LPSPI uses the attributes configured in Transmit Command (TCR) and the timing parameters defined in Clock Configuration (CCR).

The SPI bus transfer ends after the number of bits indicated by the value of FRAMESZ have been transferred (provided CONT = 0), or at the end of a word when a new transmit command word is at the top of the transmit and command FIFO. When LPSPI is disabled, the SPI bus transfers end after the transmit FIFO is empty and LPSPI is idle.

The HREQ input is only checked when PCS is negated.

68.4.2.1.1.2 Circular FIFO

The transmit and command FIFO supports a circular FIFO feature. This feature enables the LPSPI controller to (periodically) repeat a short data transfer that fits within the transmit and command FIFO, without requiring additional FIFO accesses. When the circular FIFO is enabled (CFGR0[CIRFIFO] = 1), the current state of the FIFO read pointer is saved and the status flags are not updated. After the FIFO is empty and LPSPI is idle, the FIFO read pointer is restored with the saved version, so the contents of the transmit and command FIFO are not permanently pulled from the FIFO when Circular FIFO mode is enabled.

68.4.2.1.2 Receive FIFO and data match

The receive FIFO stores received data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of being stored in the receive FIFO:

- Received data is written to the receive FIFO when the last bit of the word is sampled.
- If the transmit FIFO is empty during a multiple-word or continuous transfer, then the receive data is written to the receive FIFO before the transfer stalls (assuming CFGR1[NOSTALL] = 0) while waiting for new transmit data or for a command word to be written.

LPSPI provides a receive data match function that can match received data against one of the two words in DMR0 and DMR1, or against a masked data word. You can also configure the received data match function to compare only the first one or two received data words since the start of the frame:

- Received data that is already discarded because of TCR[RXMSK] cannot cause the data match flag to set, and delays the receive data match on the first received data word, until all discarded data is received.

- You can configure the receive data match function to discard all received data until a data match is detected, using CFGR0[RDMO].
- After a receive data match, to allow all subsequent data to be received, write 0 to CFGR0[RDMO], and then write 0 to SR[DMF].

68.4.2.1.3 Timing parameters

The timing parameters that are used for all SPI bus transfers are relative to the LPSPI functional clock divided by the selection specified in TCR[PRESCALE]. Although you can only change Clock Configuration (CCR) when LPSPI is disabled (CR[MEN] = 0), to support interfacing with different peripheral devices at different frequencies, you can change the TCR[PRESCALE] selection between SPI bus transfers by using Transmit Command (TCR).

NOTE

The minimum value shown in Table 655 is the minimum counter value, but the values of Clock Configuration (CCR) must also satisfy the data sheet specs based on the LPSPI functional clock frequency and prescaler value.

Table 655. Timing parameters

Clock Configuration (CCR) Clock Configuration 1 (CCR1)		Description	Minimum value	Maximum value
Field	Name			
SCKSET	SCK setup phase	Configures the SCK setup phase to (SCKSET + 1) cycles. The setup phase is the SCK high period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK low period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
SCKHLD	SCK hold phase	Configures the SCK hold phase to (SCKHLD + 1) cycles. The hold phase is the SCK low period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK high period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
PCSPCS	PCS-to-PCS delay	Configures the minimum delay between PCS negation and the next PCS assertion to (PCSPCS + PCSPCS + 2) cycles. When the command word is updated between transfers, there is a minimum of (PCSPCS + 1) cycles between the command word update and any change on PCS pins.	0 (2 cycles)	255 (512 cycles)
SCKSCK	SCK-to-SCK delay	Configures the delay during a continuous transfer between the last SCK edge of a frame and the first SCK edge of the continuing frame to (SCKSCK + 1) cycles. This is useful when the external peripheral requires a large delay between different words of an SPI bus transfer.	0 (1 cycle)	255 (256 cycles)

Table continues on the next page...

Table 655. Timing parameters (continued)

Clock Configuration (CCR) Clock Configuration 1 (CCR1)		Description	Minimum value	Maximum value
Field	Name			
PCSSCK	PCS-to-SCK delay	Configures the minimum delay between PCS assertion and the first SCK edge to (PCSSCK + 1) cycles.	0 (1 cycle)	255 (256 cycles)
SCKPCS	SCK-to-PCS delay	Configures the minimum delay between the last SCK edge and the PCS negation to (SCKPCS + 1) cycles.	0 (1 cycle)	255 (256 cycles)

Figure 286 shows the timing settings controlled by:

- TCR[CPHA]
- TCR[CPOL]
- CCR[SCKPCS]
- CCR[PCSSCK]
- CCR1[SCKSET]
- CCR1[SCKHLD]

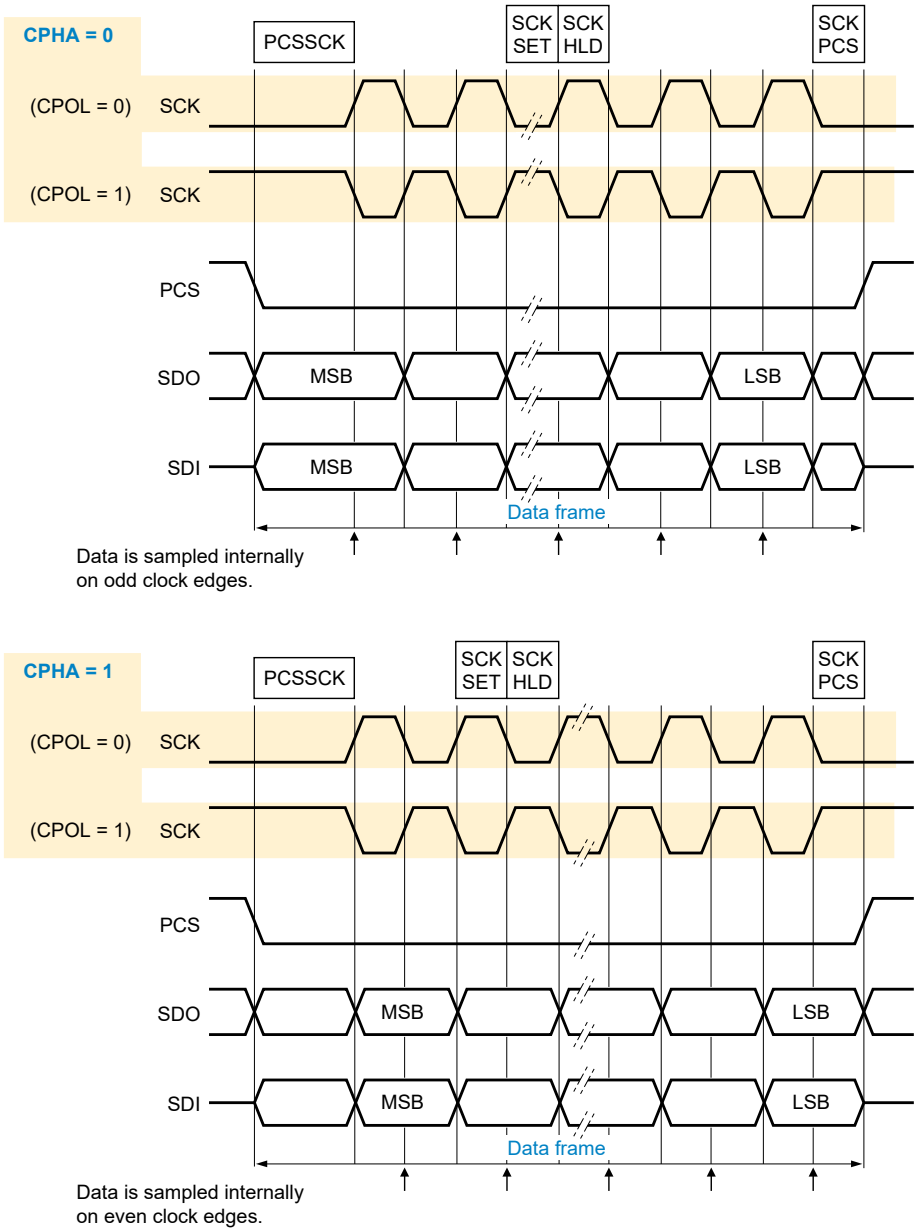


Figure 286. Clock phase (TCR[CPHA]) timing diagram example

To configure for a baud rate of 10 MHz with 50/50 duty cycle and with a functional clock frequency of 100 MHz, use the following settings:

- CCR1[SCKSET] = 0x4 (5 cycles)
- CCR1[SCKHLD] = 0x4 (5 cycles)
- CCR1[PCSPCS] = 0x8 (10 cycles)
- CCR1[SCKSCK] = 0x4 (5 cycles)
- CRR[PCSSCK] = 0x4 (5 cycles)
- CRR[SCKPCS] = 0x4 (5 cycles)
- TCR[PRESCALE] = 0x0 (divide by 1)

68.4.2.1.4 Pin configuration

Following are the pin configuration settings for half-duplex transfers:

- To swap directions or to support half-duplex transfers on the same pin, you can configure the SIN and SOUT pins using CFGR1[PINCFG].
- To specify whether an output data pin (SOUT, for example) 3-states when PCS is negated, or if the output data pin retains the last value, use CFGR1[OUTCFG].
- When configuring half-duplex transfers, you must configure the output data pins to 3-state when PCS is negated (CFGR1[OUTCFG] = 1).
- When performing half-duplex 2-bit transfers, you can write any value to CFGR1[PCSCFG].
- When performing half-duplex 4-bit transfers, you must write 1h to CFGR1[PCSCFG].

68.4.2.1.5 Clock loopback

Configure the LPSPI controller to use one of the following clocks to sample the input data:

- The SCK output clock
- A delayed version of the SCK output clock

The delayed version of the SCK is chosen by the SCK pin output delay, plus the SCK pin input delay, and is selected by writing 1 to CFGR1[SAMPLE]. Enabling the loopback version of the SCK pin can improve the setup time of the input data from the peripheral.

See the chip data sheet for the specific input setup time in Controller Loopback mode.

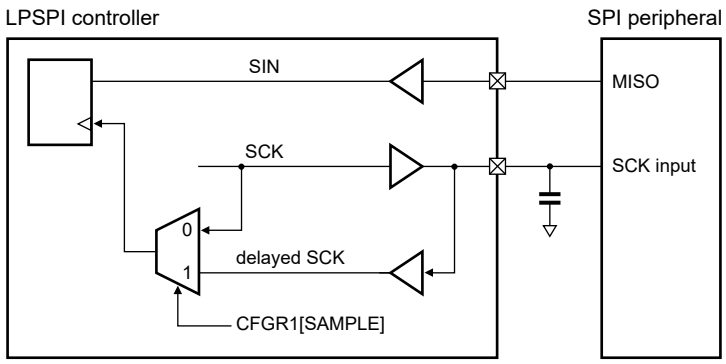


Figure 287. Clock loopback

68.4.2.2 Peripheral mode

LPSPI Peripheral mode:

- Uses the same shift register and logic that Controller mode uses.
- Does not use Clock Configuration (CCR).
- Requires Transmit Command (TCR) to remain static (unchanged) during SPI bus transfers.

68.4.2.2.1 Transmit and command FIFO commands

You must initialize Transmit Command (TCR) before enabling LPSPI in Peripheral mode, although this register is not updated until after LPSPI is enabled. After LPSPI is enabled, you must make changes to this register only when LPSPI is idle. In Peripheral mode, the LPSPI command word in this register controls SPI attributes. Before the PCS input asserts, the transmit FIFO must be filled with transmit data, or the transmit error flag sets.

Table 656. Command word in Peripheral mode

Transmit Command (TCR)		Description
Field	Name	
CPOL	Clock polarity	Specifies the polarity of the external SCK input.
CPHA	Clock phase	Specifies the clock phase of transfer.
PRESCALE	Prescaler value	Specifies the LPSPI functional clock prescaler.
PCS	Peripheral chip select	Specifies which PCS is used. The polarity of PCS is static and configured by CFGR1[PCSPOL]. If CFGR1[PCSCFG] is not equal to zero, then do not select the PCS[3:2] pins.
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.
CONT	Continuous transfer	When continuous transfer is selected in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, LPSPI passes through and transmits the received data until the next PCS negation. Whatever is shifted in on the receive data is shifted out as transmit data considering that there is a 32-bit shift register.
CONTC	Continuing command	When the continuing command is enabled in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, RXMSK is considered equal to 1 and TXMSK is considered equal to 0 until the next PCS negation. CONTC can be used to change the direction of a transfer after the number of bits indicated by FRAMESZ.
RXMSK	Receive data mask	Masks the receive data; LPSPI does not store masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.
TXMSK	Transmit data mask	Masks the transmit data so that the masked transmit data is not pulled from transmit FIFO, and the output data pin is 3-stated (unless otherwise specified in CFGR1[OUTCFG]). This option is useful for half-duplex transfers.
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse: • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. • 2-bit and 4-bit half-duplex transfers are useful for interfacing with QuadSPI memory devices, and at least either TCR[TXMSK] or TCR[RXMSK] must be 1.
FRAMESZ	Frame size	Specifies the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contain the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

68.4.2.2.2 Receive FIFO and data match

The receive FIFO stores receive data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of storing the received data in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of the two words in DMR0 and DMR1 or against a masked data word. You can also configure the data match function to compare only the first one or two received data words since the start of the frame.

- Received data that is already discarded because TCR[RXMSK] = 1 cannot cause the data match to set, and delays the match on the first received data word, until all discarded data is received.
- By using CFGR0[RDMO], you can also configure the receiver match function to discard all received data until a data match is detected.
- After a receive data match, to allow all subsequent data to be received, first write 0 to CFGR0[RDMO], then clear SR[DMF].

68.4.2.2.3 Partial received word

When the PCS pin deasserts and the receive shift register shifts in a partial word, you can configure the receive shift register to either discard the partial word or to store it in the receive FIFO. You must specify this using CFGR1[PARTIAL].

A partial word is defined as less than TCR[FRAMESZ] bits (when TCR[FRAMESZ] is equal or less than 32 bits, or it is the last word in a multi-word frame) or less than 32 bits (when TCR[FRAMESZ] is greater than 32 bits and not the last word in a multi-word frame).

A single-bit frame is not supported. A partial received word of 1 bit is supported, but a partial received frame of 1 bit is not supported.

68.4.2.2.4 Clocked interface

LPSPI supports interfacing with external controllers that provide only clock and data pins (PCS is not required). This interface requires:

- Writing 1 to TCR[CPHA] (data is changed on the leading edge of SCK and captured on the following edge).
- Configuring the PCS input to be always asserted (CFGR1[PCSPOL*n*] = 1). For example, to configure PCS[0] to be always asserted, write 1 to PCSPOL[0], and do not configure PCS[0] in the pin muxing. The chip-level drives PCS to a certain value (ideally 1); you could use CFGR1[PCSPOL*n*] to invert that value.
- Writing 1 to CFGR1[AUTOPCS] to enable automatic PCS generation. When CFGR1[AUTOPCS] = 1, a minimum of four LPSPI functional clock cycles (divided by the selection specified in TCR[PRESCALE]) is required between the last SCK edge of one word and the first SCK edge of the next word.

68.4.2.3 Low-power modes

LPSPI remains functional during low-power modes, if CR[DOZEN] = 0 and LPSPI uses an external or internal clock source that remains enabled. LPSPI can generate an interrupt or DMA request to cause a wake-up from low-power modes.

68.4.2.4 Debug mode

Table 657. Debug mode

Chip mode	LPSPI operation
Debug (the core is in Debug or Halted mode)	Can continue operating in Debug mode, if CR[DBGEN] = 1

68.4.2.5 Clocking

Table 658. LPSPI clocks

Type of clock	Description
Functional	• Asynchronous to the bus clock. • If the LPSPI functional clock remains enabled in low-power modes, then LPSPI can perform SPI bus transfers and low-power wakeups in both Controller and Peripheral modes. • LPSPI divides the functional clock by a prescaler; the resulting frequency must be at least two times faster than the SPI external clock frequency (SCK).
External	• The LPSPI shift register is clocked directly by the SCK clock. • How the SCK clock is generated or supplied depends on the mode (Controller or Peripheral): — In Controller mode, the SCK clock is generated internally. — In Peripheral mode, the SCK clock is supplied externally.
Bus	The bus clock is only used for bus accesses to the LPSPI control and configuration registers. The bus clock frequency must be high enough to support the data bandwidth requirements of the LPSPI registers, including the FIFOs.

See the chip-specific LPSPI information for more.

68.4.2.6 Reset

Table 659. LPSPI resets

Type of reset	Description
Chip	Resets the LPSPI logic and registers to their default states.
Software	• Resets the LPSPI logic and registers to their default states, except for the Control register. • The LPSPI software reset is controlled using CR[RST].
FIFO	• Resets the transmit and command FIFO and the receive FIFO. • CR[RTF] and CR[RRF] are write-only. • After being reset, FIFO is empty.

68.4.2.7 Interrupts and DMA requests

The following table lists sources (status flags) that can generate LPSPI interrupts and LPSPI transmit and receive DMA requests.

Table 660. Interrupts and DMA requests

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
TDF	Transmit data flag	Indicates that data can be written to transmit FIFO, as configured by the transmit FIFO watermark, FCR[TXWATER].	Y	TX	Y

Table continues on the next page...

Table 660. Interrupts and DMA requests (continued)

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
RDF	Receive data flag	Indicates that data can be read from the receive FIFO, as configured by the receive FIFO watermark, FCR[RXWATER].	Y	RX	Y
WCF	Word complete flag	Indicates that the word is complete and the last bit of the word has been sampled.	Y	N	Y
FCF	Frame complete flag	Indicates that the frame is complete and PCS is deasserted.	Y	RX	Y
TCF	Transfer complete flag	Indicates that transfer is complete, PCS is deasserted, and the transmit and command FIFO is empty.	Y	N	Y
TEF	Transmit error flag	Indicates a transmit and command FIFO underrun. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when transmit FIFO is empty), TEF cannot be set.	Y	N	Y
REF	Receive error flag	Indicates a receive FIFO overflow. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when receive FIFO is full), REF cannot be set.	Y	N	Y
DMF	Data match flag	Indicates that the received data matches the configured data match value.	Y	N	Y
MBF	Module busy flag	Indicates that LPSPI is busy performing an SPI bus transfer.	N	N	N

68.4.2.7.1 End-of-packet DMA transfer

The end-of-packet functionality is designed for serial interfaces where you may not be aware of the size of the transfer in advance and the data is pushed by an external device. Examples include UART receive, I2C Peripheral mode, and SPI Peripheral mode. The end-of-packet processing is intended to ensure that data is not stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and the DMA controller.

The condition that signals the end of packet is different for each serial interface but the serial peripheral and DMA process it in the same way. For example, UART end of packet is signaled by an idle line condition, I2C end of packet by a stop and/or repeated start condition, and SPI end of packet by PCS negation.

If you configure the serial peripheral to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, then data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop is complete and no additional data is pulled from the receive FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes following end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of the packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop is complete, including final offsets and optional interrupts, channel linking and scatter/gather. The final destination address can optionally be saved in the system memory or is available in the destination address register.

After the DMA terminates the major loop, no servicing of the receive FIFO occurs until either software or hardware reconfigures the DMA (for example, channel linking or scatter/gather). This delay must be minimized to avoid receiver FIFO overrun. NXP does not recommend automatic DMA end-of-packet processing when there are only a few words transferred between end-of-packet conditions. That is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

68.4.2.7.2 DMA support registers

To support efficient DMA transfers to the transmit and control FIFO, an alias register supports 32-bit write access to the Transmit Command (TCR) and an alias region supports incrementing 32-bit write accesses to the Transmit Data (TDR).

- Transmit Command Burst (TCBR) is a 32-bit alias register for writing to TCR.
- Transmit Data Burst (TDBR0 - TDBR127) is a 512-byte alias region that supports writing to the TDR.

The burst alias locations are contiguous. A DMA transfer can start by writing to the TCBR register to initialize the transfer and then increment into the TDBR region without changing the DMA transfer size. The alias registers can also be used by the DMA to perform burst transfers when accessing the transmit FIFO.

The transmit FIFO prevents writes that overflow the FIFO, but this prevention does not signal an error. Do not perform writes to the TDBR unless there is sufficient room in the transmit FIFO.

Receive Data Burst (RDBR0 - RDBR127) is a 512-byte alias region that supports reading the Receive Data. This can be used by the DMA to perform burst transfers when accessing the receive FIFO.

68.4.2.8 Peripheral triggers

The connection of the LPSPI peripheral triggers with other peripherals depends on the device that is used.

Table 661. Peripheral triggers

Type of trigger	Description	Additional information
Frame output	The frame output trigger: • Asserts at the end of each frame (when PCS deasserts). • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	LPSPI generates two output triggers that can be connected to other peripherals on the chip.
Word output	The word output trigger: • Asserts at the end of each received word. • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	
Input	To control the start of an LPSPI bus transfer, the LPSPI input trigger can be selected instead of the HREQ input:	

Table continues on the next page...

Table 661. Peripheral triggers (continued)

Type of trigger	Description	Additional information
	• The LPSPI input trigger is synchronized, and must assert for at least two cycles of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE] so that the input trigger can be detected. • When LPSPI is busy, the HREQ input (and therefore the LPSPI input trigger) is ignored. • When LPSPI is busy, both the HREQ and LPSPI input triggers are ignored. They are used to start a new transfer when LPSPI is idle.	

68.4.3 External signals

Table 662. External signals

Signal	Name	Description	I/O
SCK	Serial clock	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[0]	Peripheral chip select	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[1]/HREQ	Peripheral chip select or host request	Host request pin is selected when CFGR0[HREN] = 1 and CFGR0[HRSEL] = 0: • Input in either Peripheral mode or when used as controller host request • Output in either Controller mode or when used as peripheral host request	I/O
PCS[2]/DATA[2]	Peripheral chip select or data pin 2 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Peripheral mode • Output in Controller mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers • Output in half-duplex parallel data transmit transfers	I/O
PCS[3]/DATA[3]	Peripheral chip select or data pin 3 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Peripheral mode • Output in Controller mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers	I/O

Table continues on the next page...

Table 662. External signals (continued)

Signal	Name	Description	I/O
		• Output in half-duplex parallel data transmit transfers	
SOUT/DATA[0]	Serial data output	Can be configured as serial data input signal (used as data pin 0 in half-duplex parallel data transfers)	I/O
SIN/DATA[1]	Serial data input	Can be configured as serial data output signal (used as data pin 1 in half-duplex parallel data transfers)	I/O

68.4.4 Initialization

This module does not require initialization.

68.4.5 Memory map and registers

NOTE

- Writing to a read-only register or reading a write-only register can cause bus errors.
- LPSPI does not check values programmed in registers for validity, so you must take care to write valid values only.

68.4.5.1 LPSPI register descriptions

68.4.5.1.1 LPSPI memory map

LPSPI0 base address: 4011_0000h

LPSPI1 base address: 4011_1000h

LPSPI2 base address: 4011_2000h

LPSPI3 base address: 4011_3000h

LPSPI4 base address: 4017_1000h

LPSPI5 base address: 4017_2000h

LPSPI6 base address: 4017_3000h

LPSPI7 base address: 4017_4000h

LPSPI8 base address: 4019_9000h

LPSPI9 base address: 4019_A000h

LPSPI10 base address: 4019_B000h

LPSPI11 base address: 4019_C000h

LPSPI12 base address: 4019_D000h

LPSPI13 base address: 4019_E000h

LPSPI14 base address: 4048_4000h

LPSPI16 base address: 4040_5000h

LPSPI17 base address: 4032_6000h

LPSPI18 base address: 4032_7000h

LPSPI19 base address: 4032_8000h

LPSPI20 base address: 4032_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0200_0004h
4h	Parameter (PARAM)	32	R	0004_0303h
10h	Control (CR)	32	RW	0000_0000h
14h	Status (SR)	32	RW	0000_0001h
18h	Interrupt Enable (IER)	32	RW	0000_0000h
1Ch	DMA Enable (DER)	32	RW	0000_0000h
20h	Configuration 0 (CFGR0)	32	RW	0000_0000h
24h	Configuration 1 (CFGR1)	32	RW	0000_0000h
30h	Data Match 0 (DMR0)	32	RW	0000_0000h
34h	Data Match 1 (DMR1)	32	RW	0000_0000h
40h	Clock Configuration (CCR)	32	RW	0000_0000h
44h	Clock Configuration 1 (CCR1)	32	RW	0000_0000h
58h	FIFO Control (FCR)	32	RW	0000_0000h
5Ch	FIFO Status (FSR)	32	R	0000_0000h
60h	Transmit Command (TCR)	32	RW	0000_001Fh
64h	Transmit Data (TDR)	32	W	0000_0000h
70h	Receive Status (RSR)	32	R	0000_0002h
74h	Receive Data (RDR)	32	R	0000_0000h
78h	Receive Data Read Only (RDROR)	32	R	0000_0000h
3FCh	Transmit Command Burst (TCBR)	32	W	0000_0000h
400h - 5FCh	Transmit Data Burst (TDBR0 - TDBR127)	32	W	0000_0000h
600h - 7FCh	Receive Data Burst (RDBR0 - RDBR127)	32	R	0000_0000h

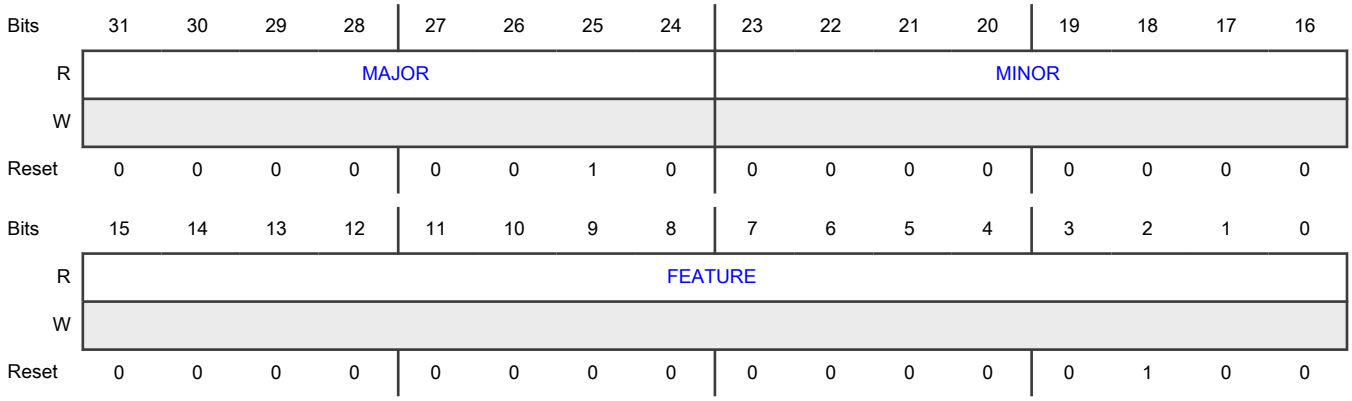
68.4.5.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function
Contains version numbers for the module design and feature set.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number of the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number of the module specification.
15-0 FEATURE	Module Identification Number Indicates the feature set number 0000_0000_0000_0100b - Standard feature set supporting a 32-bit shift register. All other values are reserved.

68.4.5.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains:

- Number of PCS pins.
- Receive FIFO size.
- Transmit FIFO size.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PCSNUM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Fields

Field	Function
31-24 —	Reserved
23-16 PCSNUM	PCS Number Indicates the number of PCS pins supported.
15-8 RXFIFO	Receive FIFO Size Indicates the maximum number of words in the receive FIFO. The maximum number of words is 2^{RXFIFO}.
7-0 TXFIFO	Transmit FIFO Size Indicates the maximum number of words in the transmit FIFO. The maximum number of words is 2^{TXFIFO}.

68.4.5.1.4 Control (CR)

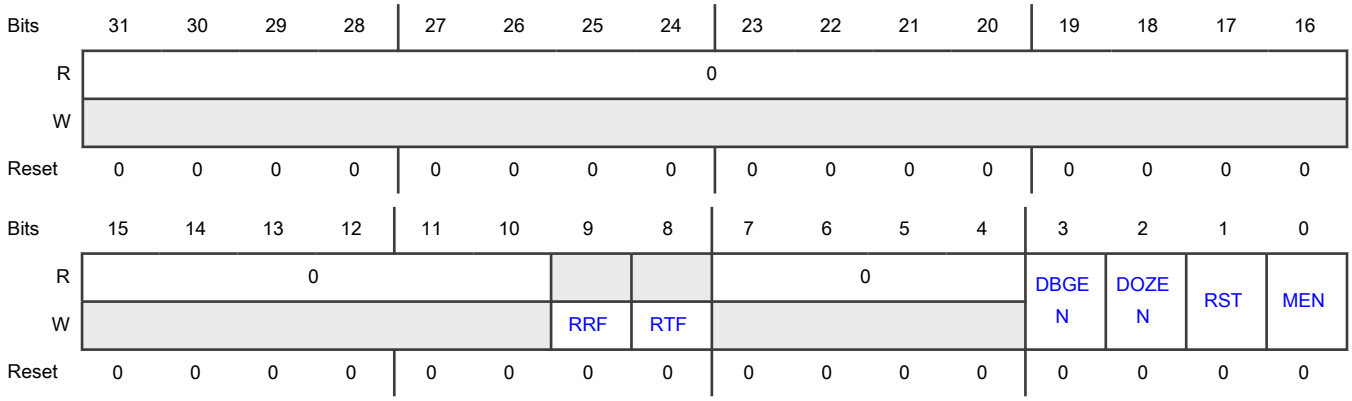
Offset

Register	Offset
CR	10h

Function

Contains fields that control the module operation.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Deletes all entries in the receive FIFO. This field always reads 0. 0b - No effect 1b - Reset
8 RTF	Reset Transmit FIFO Deletes all entries in the transmit FIFO. This field always reads 0. 0b - No effect 1b - Reset
7-4 —	Reserved
3 DBGEN	Debug Enable Enables LPSPI when the CPU is in Debug mode. If this field is 0, LPSPI is disabled when the CPU is halted; the PCS pin is deasserted after the transmit FIFO is empty regardless of the state of Transmit Command (TCR). You must update this field only when LPSPI is disabled (MEN = 0). 0b - Disable 1b - Enable
2 DOZEN	Doze Mode Enable Enables LPSPI when the chip is in Doze mode. You must update this field only when LPSPI is disabled (MEN = 0).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Enable 1b - Disable
1 RST	Software Reset Resets all internal logic and registers, except Control (CR). The reset takes effect immediately and remains asserted until you write 0 to it. There is no minimum delay required before clearing the software reset by writing 0. 0b - Not reset 1b - Reset
0 MEN	Module Enable Enables the module. After writing 0, MEN remains set until LPSPI has completed the current transfer and is idle. 0b - Disable 1b - Enable

68.4.5.1.5 Status (SR)

Offset

Register	Offset
SR	14h

Function
Contains data flow status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DMF	REF	TEF	TCF	FCF	WCF	0						RDF	TDF
W			W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-25 —	Reserved
24 MBF	Module Busy Flag Indicates, in Controller mode, whether there is data to transmit and LPSPI is able to transmit (for example, the HREQ pin is asserted). The HREQ pin deasserts after the PCS pin deasserts and the LPSPI controller has waited for half the time specified in CCR[DBT] with no new data to transmit. Peripheral mode sets this flag when LPSPI is enabled and PCS is asserted. 0b - LPSPI is idle 1b - LPSPI is busy
23-14 —	Reserved
13 DMF	Data Match Flag Indicates whether the received data matches DMR0[MATCH0] and/or DMR1[MATCH1] (as configured by CFGR1[MATCFG]). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No match 1b - Match When writing 0b - No effect 1b - Clear the flag
12 REF	Receive Error Flag Indicates a receive FIFO overflow error. When this flag is set: 1. End the transfer. 2. Empty the receive FIFO. 3. Clear this flag. 4. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overflow

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Overflow When writing 0b - No effect 1b - Clear the flag
11 TEF	Transmit Error Flag Indicates a transmit FIFO underrun error. When this flag is set: 1. End the transfer. 2. Clear this flag. 3. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underrun 1b - Underrun When writing 0b - No effect 1b - Clear the flag
10 TCF	Transfer Complete Flag Indicates, in Controller mode, whether all transfers are complete and LPSPI has returned to the Idle state and the transmit FIFO is empty. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
9 FCF	Frame Complete Flag Indicates whether a frame transfer is complete after PCS deasserts. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
8 WCF	Word Complete Flag Indicates whether the last bit of a received word is sampled. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the number of words in the receive FIFO is greater than the value in FCR[RXWATER]. 0b - Receive data not ready 1b - Receive data ready
0 TDF	Transmit Data Flag Indicates whether the number of words in the transmit FIFO is equal to or less than the value in FCR[TXWATER]. 0b - Transmit data not requested 1b - Transmit data requested

68.4.5.1.6 Interrupt Enable (IER)

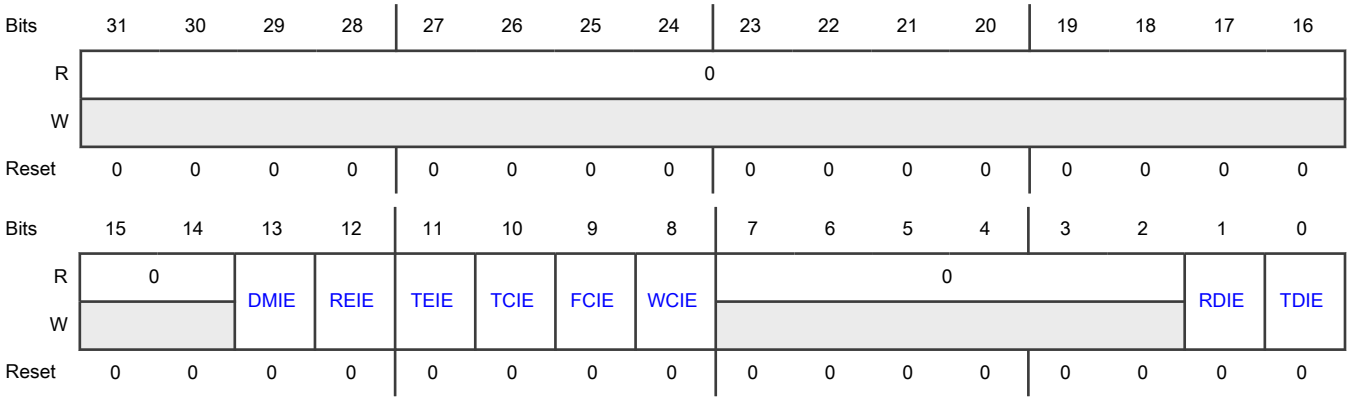
Offset

Register	Offset
IER	18h

Function

Enables interrupts based on data flow and errors.

Diagram



Fields

Field	Function
31-14 —	Reserved
13 DMIE	Data Match Interrupt Enable Enables the data match interrupt. 0b - Disable 1b - Enable
12 REIE	Receive Error Interrupt Enable Enables the receive error interrupt. 0b - Disable 1b - Enable
11 TEIE	Transmit Error Interrupt Enable Enables the transmit error interrupt. 0b - Disable 1b - Enable
10 TCIE	Transfer Complete Interrupt Enable Enables the transfer complete interrupt. 0b - Disable 1b - Enable
9	Frame Complete Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
FCIE	Enables the frame complete interrupt. 0b - Disable 1b - Enable
8 WCIE	Word Complete Interrupt Enable Enables the word complete interrupt. 0b - Disable 1b - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables the receive data interrupt. 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables the transmit data interrupt. 0b - Disable 1b - Enable

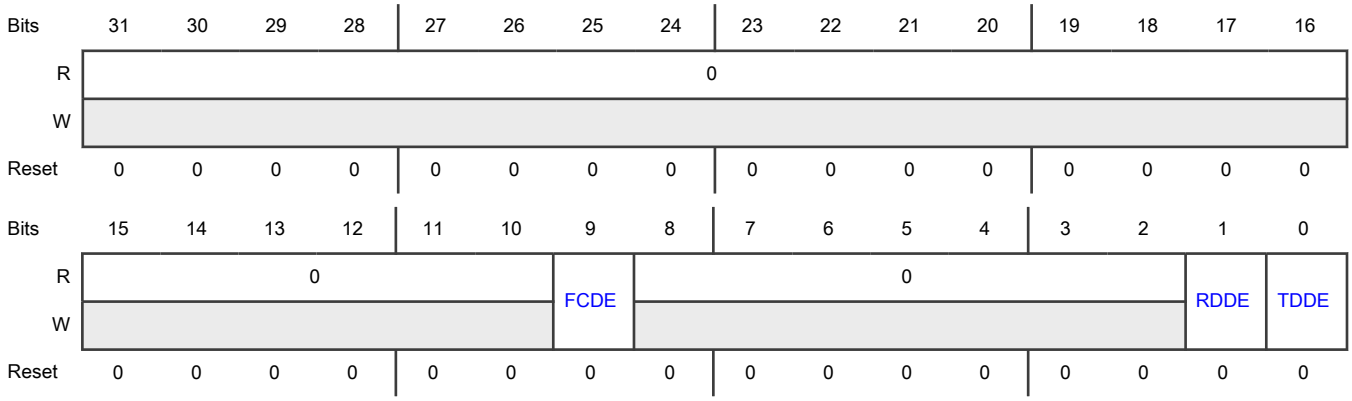
68.4.5.1.7 DMA Enable (DER)

Offset

Register	Offset
DER	1Ch

Function
Enables the DMA data flow.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 FCDE	Frame Complete DMA Enable Enables DMA end-of-packet processing. After the last word of a frame is read from the receive data FIFO, reading the receive data FIFO returns an end-of-packet signal with the receive data forced to FFFF_FFFFh. This continues until the DMA minor loop completes, and then SR[FCF] deasserts if the receive FIFO is empty or if LPSPI is busy (SR[MBF] = 1). **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
LPSPI0	DER	—
LPSPI1	DER	—
LPSPI2	DER	—
LPSPI3	DER	—
LPSPI4	DER	—
LPSPI5	DER	—
LPSPI6	DER	—
LPSPI7	DER	—
LPSPI8	DER	—

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>LPSPI9</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI10</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI11</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI12</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI13</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI14</td><td>—</td><td>DER</td></tr>
<tr><td>LPSPI16</td><td>—</td><td>DER</td></tr>
<tr><td>LPSPI17</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI18</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI19</td><td>DER</td><td>—</td></tr>
<tr><td>LPSPI20</td><td>DER</td><td>—</td></tr>
</table>
0b - Disable
1b - Enable
</td></tr>
<tr><td>8-2
—</td><td>Reserved</td></tr>
<tr><td>1
RDDE</td><td>Receive Data DMA Enable
Enables the receive data DMA.
0b - Disable
1b - Enable</td></tr>
<tr><td>0
TDDE</td><td>Transmit Data DMA Enable
Enables the transmit data DMA.
0b - Disable
1b - Enable</td></tr>
</table>

68.4.5.1.8 Configuration 0 (CFGR0)

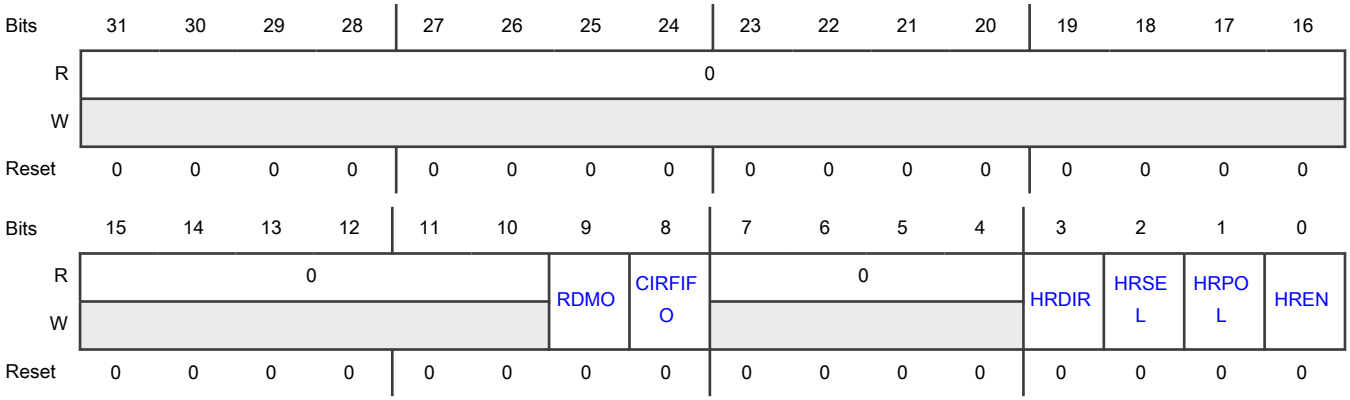
Offset

Register	Offset
CFGR0	20h

Function

Includes fields to configure LPSPI.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RDMO	Receive Data Match Only Enables receive data match. When enabled, all received data that does not cause SR[DMF] to assert is discarded: • Write 1 to this field when LPSPI is idle and SR[DMF] = 0. • After SR[DMF] = 1, this field is ignored. • To ensure that no receive data is lost when disabling RDMO, write 0 to this field before clearing SR[DMF]. See CFGR1[MATCFG] for the received data matching options. When disabled, all received data is stored in the receive FIFO. 0b - Disable 1b - Enable
8 CIRFIFO	Circular FIFO Enable Enables circular FIFO.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO is emptied as in normal operation, but when LPSPI is idle and the transmit FIFO is empty, the read pointer value is restored from the temporary register. This restoring of the read pointer causes the contents of the transmit FIFO to be cycled through repeatedly. **NOTE** The read pointer is restored for as long as this field is 1. Writing additional words to the FIFO when this field is 1 adds them to the end of the FIFO, up to the size of the transmit FIFO. 0b - Disable 1b - Enable
7-4 —	Reserved
3 HRDIR	Host Request Direction Specifies the direction of the HREQ pin. You must configure the HREQ pin only as an output when LPSPI is in Peripheral mode. The HREQ pin direction must be an input for Controller mode. 0b - Input 1b - Output
2 HRSEL	Host Request Select Specifies the source of the host request input. When the host request function is enabled with the HREQ pin, the PCS[1] function is disabled. 0b - HREQ pin 1b - Input trigger
1 HRPOL	Host Request Polarity Specifies the polarity of the HREQ pin or input trigger. 0b - Active high 1b - Active low
0 HREN	Host Request Enable Enables LPSPI, in Controller mode, to start a new SPI bus transfer only if the host request input is asserted. When LPSPI is busy, the host request input is ignored. In Peripheral mode, causes the HREQ output pin to assert when data is available to be transmitted. 0b - Disable 1b - Enable

68.4.5.1.9 Configuration 1 (CFGR1)

Offset

Register	Offset
CFGR1	24h

Function

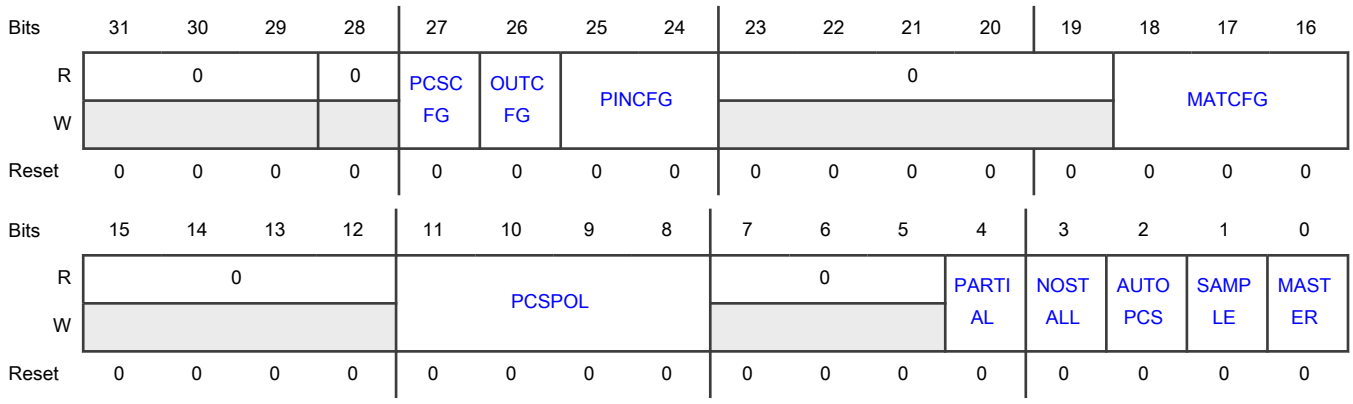
Includes fields to configure LPSPI. You must write to this register only when LPSPI is disabled.

In addition to pin and output configurations, this register contains match configuration details; the following table shows match conditions specified in MATCFG.

Table 663. Match conditions for CFGR1[MATCFG]

Condition	Description
Match first data word with compare word	Match if first data word equals MATCH0 logically ORed with MATCH1 first_data_word == (MATCH0 || MATCH1)
Match any data word with compare word	Match if any data word equals MATCH0 logically ORed with MATCH1 any_data_word == (MATCH0 || MATCH1)
Sequential match, first data word	Match if first data word equals MATCH0, and second data word equals MATCH1 (first_data_word == MATCH0) && (second_data_word == MATCH1)
Sequential match, any data word	Match if any data word equals MATCH0, and the next data word equals MATCH1 (any_data_word == MATCH0) && (next_data_word == MATCH1)
Match first data word (masked) with compare word (masked)	Match if first data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (first_data_word && MATCH1) == (MATCH0 && MATCH1)
Match any data word (masked) with compare word (masked)	Match if any data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (any_data_word && MATCH1) == (MATCH0 && MATCH1)

Diagram



Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27 PCSCFG	Peripheral Chip Select Configuration Specifies PCS pin configuration. When performing parallel transfers, you must configure this field to enable the desired transfer. 0b - PCS[3:2] configured for chip select function 1b - PCS[3:2] configured for half-duplex 4-bit transfers (PCS[3:2] = DATA[3:2])
26 OUTCFG	Output Configuration Specifies whether the output data is 3-stated between accesses (when PCS is deasserted). When performing half-duplex transfers, this field must be 1. 0b - Retain last value 1b - 3-stated
25-24 PINCFG	Pin Configuration Specifies the pins used for input and output data during serial transfers. **NOTE** When performing parallel transfers, this field must be 00b or 10b. 00b - SIN is used for input data; SOUT is used for output data 01b - SIN is used for both input and output data; only half-duplex serial transfers are supported 10b - SOUT is used for both input and output data; only half-duplex serial transfers are supported 11b - SOUT is used for input data; SIN is used for output data
23-19 —	Reserved
18-16 MATCFG	Match Configuration Specifies the condition that causes SR[DMF] to assert. See the match conditions listed in Table 1 for more information. **NOTE** When writing to this field, either the old value or new value must be in the disabled state (0). You cannot transition from a nonzero value to another nonzero value. 000b - Match is disabled 001b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010b - Match first data word with compare word 011b - Match any data word with compare word 100b - Sequential match, first data word 101b - Sequential match, any data word 110b - Match first data word (masked) with compare word (masked) 111b - Match any data word (masked) with compare word (masked)
15-12 —	Reserved
11-8 PCSPOL	Peripheral Chip Select Polarity Specifies the polarity of each PCS pin. Bit *n* in this field (the least-significant bit is bit 0) corresponds to PCS[*n*]. 0000b - Active low 0001b - Active high
7-5 —	Reserved
4 PARTIAL	Partial Enable Specifies whether LPSPI, when in Peripheral mode, stores a partial received word in the receive FIFO, or discards it, when PCS deasserts. See Partial received word for more information. 0b - Discard 1b - Store
3 NOSTALL	No Stall Disables a normal operating feature that causes LPSPI, when in Controller mode, to stall transfers when the transmit FIFO is empty or when the receive FIFO is full. This feature prevents transmit FIFO underruns and receive FIFO overruns. Writing 1 to this field disables this functionality. 0b - Disable 1b - Enable
2 AUTOPCS	Automatic PCS Enables automatic PCS generation. For correct operation in Peripheral mode, LPSPI requires the PCS signal to deassert between frames. Writing 1 to this field generates an internal PCS signal at the end of each transfer word when TCR[CPHA] = 1. When this field is 1, SCK must remain idle for at least four LPSPI functional clock cycles, divided by the prescaler (see TCR[PRESCALE]) selected between each word to ensure correct operation. This field is ignored in Controller mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
1 SAMPLE	Sample Point Specifies the SCK clock edge on which LPSPI, when in Controller mode, samples input data. Writing 1 to this field causes LPSPI to sample input data on a delayed loopback SCK clock edge, which improves the setup time when sampling data (see Clock loopback). In this configuration, the input data setup time in Controller mode is equal to the input data setup time in Peripheral mode. In Peripheral mode, this field is ignored. **NOTE** When SAMPLE = 1, both the input and output buffers must be enabled for the SCK pin. Buffers are configured at the chip level. 0b - SCK edge 1b - Delayed SCK edge
0 MASTER	Controller Mode Specifies the LPSPI operating mode, Controller or Peripheral. This field directly controls the direction of the SCK and PCS pins. 0b - Peripheral mode 1b - Controller mode

68.4.5.1.10 Data Match 0 (DMR0)

Offset

Register	Offset
DMR0	30h

Function

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register when CFGR1[MATCFG] > 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MATCH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MATCH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH0	Match 0 Value Specifies the MATCH0 value to be compared against received data.

68.4.5.1.11 Data Match 1 (DMR1)

Offset

Register	Offset
DMR1	34h

Function

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register while CFGR1[MATCFG] > 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MATCH1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MATCH1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH1	Match 1 Value Specifies the MATCH1 value to be compared against received data.

68.4.5.1.12 Clock Configuration (CCR)

Offset

Register	Offset
CCR	40h

Function

Contains clock configuration fields that are used only in Controller mode; you can only change them when LPSPI is disabled (CR[MEN] = 0).

Warning

Writing a 32-bit value to this register overwrites Clock Configuration 1 (CCR1); DBT and SCKDIV always read 0.

To avoid overwriting CCR1, do one of the following:

- Write to all four fields in Clock Configuration (CCR) simultaneously and only once in a 32-bit data.
- Modify the values of CCR[SCKPCS] and/or CCR[PCSSCK]; write only these two upper bytes in a 16-bit data or one of them in an 8-bit data.
- Modify CCR1[PCSPCS] and CCR1[SCKSCK] only or CCR1[SCKSET] and CCR1[SCKHLD] only, write respectively to CCR[DBT] or CCR[SCKDIV] in 8-bit data.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0							
W	DBT								SCKDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24	SCK-to-PCS Delay

Table continues on the next page...

Table continued from the previous page...

Field	Function
SCKPCS	Configures SCK-to-PCS delay. In Controller mode, this field helps you configure the delay from the last SCK edge to PCS negation: • The delay is equal to (SCKPCS + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 286 for more information.
23-16 PCSSCK	PCS-to-SCK Delay Configures PCS-to-SCK delay. In Controller mode, this field helps you configure the delay from PCS assertion to the first SCK edge: • The delay is equal to (PCSSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 286 for more information.
15-8 DBT	Delay Between Transfers Configures the delay between transfers. Writing to this field updates the contents of CCR1[PCSPCS] and CCR1[SCKSCK].
7-0 SCKDIV	SCK Divider Updates the contents of CCR1[SCKSET] and CCR1[SCKHLD]. Baud rate = function clock ÷ (2^PRESCALE × (SCKSET + SCKHLD + 2))

68.4.5.1.13 Clock Configuration 1 (CCR1)

Offset

Register	Offset
CCR1	44h

Function

Contains clock configuration fields, which are used only in Controller mode. You can change them only when LPSPI is disabled (CR[MEN] = 0).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKSCK								PCSPCS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCKHLD								SCKSET							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 SCKSCK	SCK Inter-Frame Delay Configures SCK inter-frame delay in Controller mode: • This field helps you configure the delay from the last SCK pulse of a frame and the first SCK pulse of the following frame, in a continuous transfer. • The delay is equal to (SCKSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. **NOTE** For backward compatibility, writing to CCR[DBT] updates this field with the value written.
23-16 PCSPCS	PCS to PCS Delay Configures PCS to PCS delay in Controller mode: • This field helps you configure the delay from the PCS negation to the next PCS assertion. • The delay is equal to (PCSPCS + PCSPCS + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is two cycles. • Half of the delay (PCSPCS + 1) occurs before PCS assertion and the other half of the delay (PCSPCS + 1) occurs after PCS negation. If the command word is updated between two transfers, then the command word is updated halfway between the PCS negation of the last transfer and PCS assertion of the next transfer. • The command word specifies which PCS signal is used, the polarity and phase of the SCK signal, and the selected prescaler. **NOTE** For backward compatibility, writing to CCR[DBT] updates this field with (DBT÷2) rounded up.
15-8	SCK Hold

Table continues on the next page...

Table continued from the previous page...

Field	Function
SCKHLD	Configures the hold phase of the SCK pin in Controller mode: • The hold phase is the delay between the SCK edge that samples the receive data and the SCK edge that drives the transmit data. • It is the SCK low period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK high period when CPHA = 0, CPOL = 0 and CPHA = 1, CPOL = 1. • The SCK hold phase delay is equal to (SCKHLD + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 286 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded down.
7-0 SCKSET	SCK Setup Configures the setup phase of the SCK pin in Controller mode: • The setup phase is the delay between the SCK edge that drives the transmit data and the SCK edge that samples the receive data. • It is the SCK high period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK low period when CPHA = 0 and CPOL = 0, or CPHA = 1 and CPOL = 1. • The SCK setup phase delay is equal to (SCKSET + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 286 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded up.

68.4.5.1.14 FIFO Control (FCR)

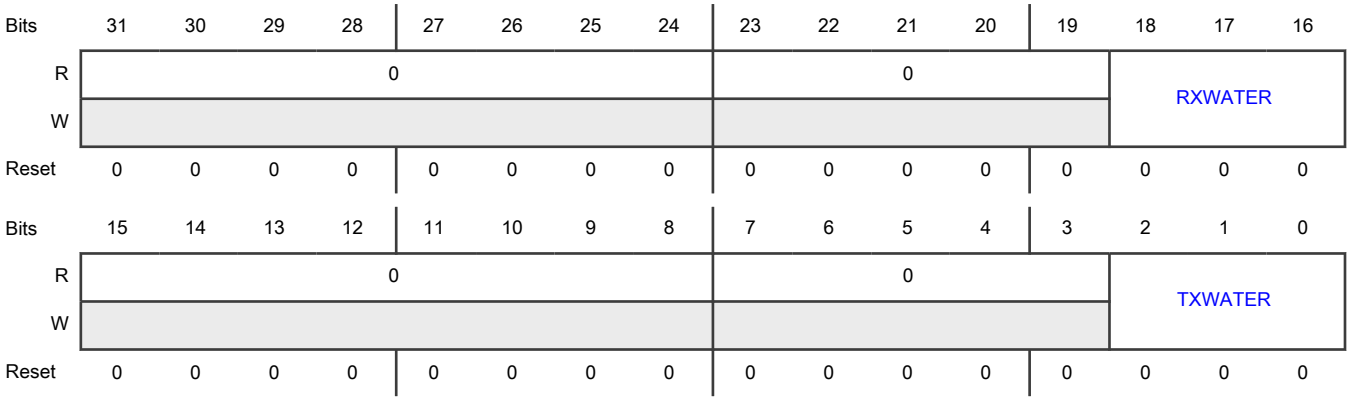
Offset

Register	Offset
FCR	58h

Function

Contains the receive FIFO and transmit FIFO watermark values.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-19 —	Reserved
18-16 RXWATER	Receive FIFO Watermark Causes LPSPI to set SR[RDF] when the number of words in the receive FIFO is greater than RXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.
15-8 —	Reserved
7-3 —	Reserved
2-0 TXWATER	Transmit FIFO Watermark Causes LPSPI to set SR[TDF] when the number of words in the transmit FIFO is equal to or less than TXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.

68.4.5.1.15 FIFO Status (FSR)

Offset

Register	Offset
FSR	5Ch

Function

Contains fields that indicate the number of words currently stored in the receive and transmit FIFOs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0				RXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0				TXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-20 —	Reserved
19-16 RXCOUNT	Receive FIFO Count Indicates the number of words currently stored in the receive FIFO.
15-8 —	Reserved
7-4 —	Reserved
3-0 TXCOUNT	Transmit FIFO Count Indicates the number of words currently stored in the transmit FIFO.

68.4.5.1.16 Transmit Command (TCR)

Offset

Register	Offset
TCR	60h

Function

Pushes the data into the transmit FIFO, in the same order as written.

When you write to either this register or to Transmit Data (TDR), each write pushes data into the transmit FIFO. You must write to this register only using 32-bit writes, which are tagged and cause the command register to update; after that the entry reaches the top of the FIFO and LPSPI is enabled. This allows changes to the command word and the transmit data itself to be interleaved. That is, writes to the two registers can be interleaved (write command word, then data word, then command word, and so on). Changing the command word causes all subsequent SPI bus transfers to be performed using the new command word:

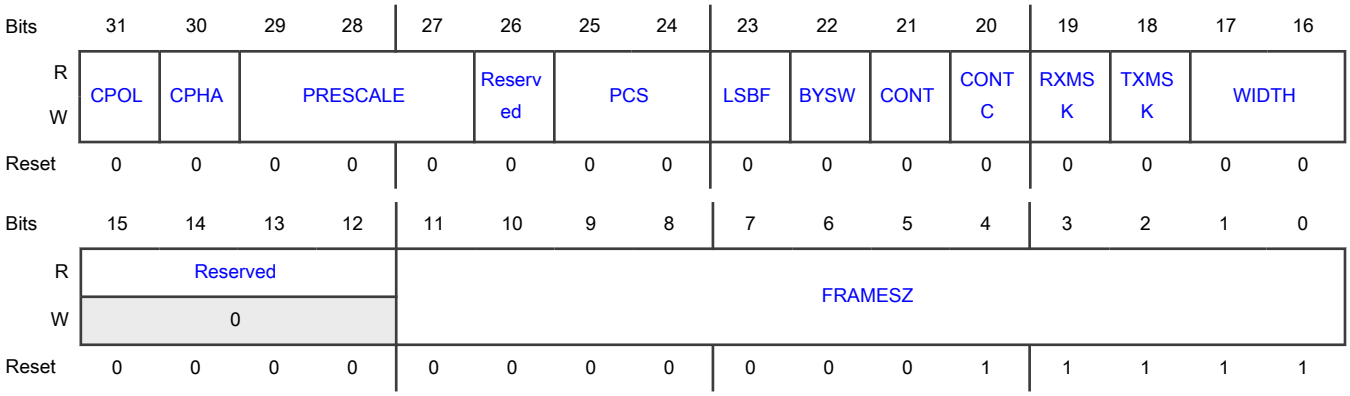
- In Controller mode, writing a new command word does not initiate a new transfer, unless TXMSK is 1. Transfers are initiated by transmit data in the transmit FIFO, or by a new command word (with TXMSK = 1). Hardware writes 0 to TXMSK when PCS deasserts.
- In Controller mode, if the command word is changed before an existing frame has completed, then the existing frame terminates and the command word updates. The command word can be changed during a continuous transfer, if CONTC of the new command word is 1 and the command word is written on a frame size boundary.
- In Peripheral mode, the command word must be changed only when LPSPI is idle and there is no SPI bus transfer.

Avoid resetting the transmit FIFO after writing to this register; wait for the command register to update from the FIFO first.

Avoid register reading problems: Reading this register returns the current state of the register. Reading this register at the same time that it is loaded from the transmit FIFO can return an incorrect register value. It is recommended to:

- Read this register when the transmit FIFO is empty.
- Read this register more than once and then compare the returned values.

Diagram



Fields

Field	Function
31 CPOL	Clock Polarity Specifies the value of SCK when it is idle. You can update this field only when PCS is deasserted. See Figure 286 for more information. 0b - Inactive low 1b - Inactive high
30 CPHA	Clock Phase Indicates whether data is captured or changed on the leading edge of SCK and captured or changed on the following edge of SCK. You can update this field only when PCS is deasserted. See Figure 286 for more information. 0b - Captured 1b - Changed
29-27 PRESCALE	Prescaler Value Specifies the division of the LPSPI functional clock. For all SPI bus transfers, this value is applied to Clock Configuration (CCR). You can update this field only when PCS is deasserted. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32 110b - Divide by 64 111b - Divide by 128
26 —	Reserved
25-24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. This field is updated only when PCS is deasserted. **NOTE** This entire field is not fully supported in every LPSPI module instance. See the chip-specific LPSPI information. 00b - Transfer using PCS[0] 01b - Transfer using PCS[1] 10b - Transfer using PCS[2]

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Transfer using PCS[3]
23 LSBF	LSB First Indicates whether data is transferred with MSB first or LSB first. 0b - MSB first 1b - LSB first
22 BYSW	Byte Swap Swaps the contents of [31:24] with [7:0] and [23:16] with [15:8] for each transmit data word read from the FIFO and for each received data word stored to the FIFO (or compared with match registers). 0b - Disable byte swap 1b - Enable byte swap
21 CONT	Continuous Transfer Enables continuous transfer: • In Controller mode, this field keeps PCS asserted at the end of the frame size until a command word is received that starts a new frame. • In Peripheral mode, when this field is enabled, LPSPI only transmits the first FRAMESZ bits, after which LPSPI transmits received data (assuming a 32-bit shift register) until the next PCS negation. 0b - Disable 1b - Enable
20 CONTC	Continuing Command Enables the command word to be changed within a continuous transfer in Controller mode: • The initial command word must enable continuous transfer (CONT = 1). • The continuing command must have CONTC = 1. • The continuing command word must be loaded on a frame size boundary. For example, if the continuous transfer has a frame size of 64 bits, then a continuing command word must be loaded on a 64-bit boundary. In Peripheral mode, this field modifies the internal RXMSK and TXMSK configuration after the first FRAMESZ bits and until PCS negation: • Receive data is discarded after the first FRAMESZ bits. If CONT is also 1, this does not block the transmission of received data. • Transmit data is not masked after the first FRAMESZ bits. This allows the first FRAMESZ bits to be received and a response transmitted. 0b - Command word for start of new transfer 1b - Command word for continuing transfer
19	Receive Data Mask

Table continues on the next page...

Table continued from the previous page...

Field	Function
RXMSK	Masks receive data (receive data is not stored in the receive FIFO). 0b - Normal transfer 1b - Mask receive data
18 TXMSK	Transmit Data Mask Masks transmit data (no data is loaded from the transmit FIFO and the output pin is 3-stated). In Controller mode, TXMSK initiates a new transfer that cannot be aborted by another command word. TXMSK automatically transitions to 0 at the end of the transfer. 0b - Normal transfer 1b - Mask transmit data
17-16 WIDTH	Transfer Width Configures serial (1-bit) or parallel transfers. For half-duplex parallel transfers, either RXMSK or TXMSK must be 1. 00b - 1-bit transfer 01b - 2-bit transfer 10b - 4-bit transfer 11b - Reserved
15-12 —	Reserved
11-0 FRAMESZ	Frame Size Configures the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

68.4.5.1.17 Transmit Data (TDR)

Offset

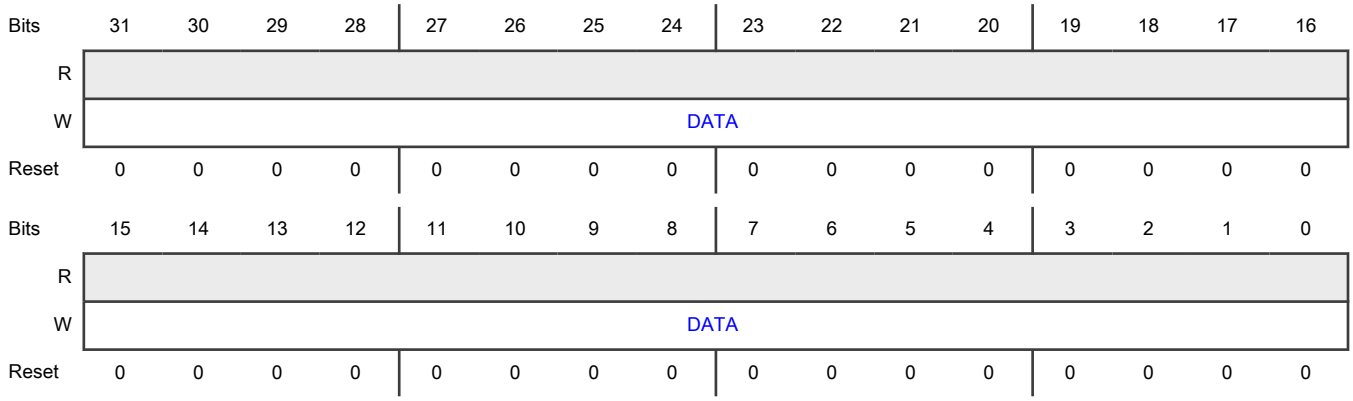
Register	Offset
TDR	64h

Function

Pushes the data into the transmit FIFO, in the same order that the data is written. You can write to this register using 32-, 16-, or 8-bit writes.

When you write to this register or to Transmit Command (TCR), each write pushes data into the FIFO with zero pushed in unwritten bytes.

Diagram



Fields

Field	Function
31-0 DATA	Transmit Data Indicates transmit data. Both 8-bit and 16-bit writes of transmit data zero-extend the data written and push the data into the transmit FIFO. To zero-extend 8-bit and 16-bit writes (to 32 bits) means that the higher order (most significant) empty parts of the 8-bit and 16-bit writes are filled with zeroes.

68.4.5.1.18 Receive Status (RSR)

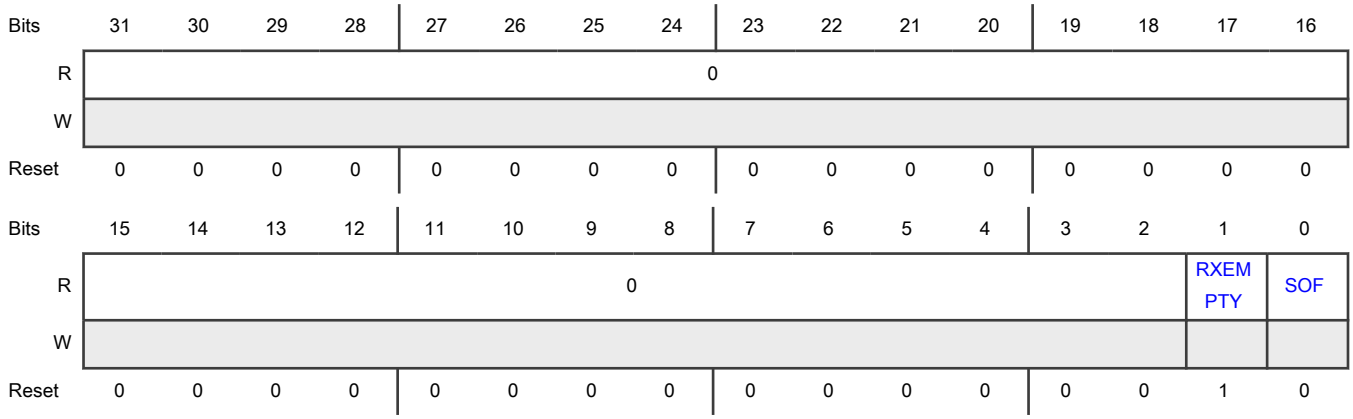
Offset

Register	Offset
RSR	70h

Function

Contains data flow status fields for receive FIFO.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RXEMPTY	RX FIFO Empty Indicates whether the receive FIFO is empty. 0b - Not empty 1b - Empty
0 SOF	Start of Frame Indicates whether this is the first data word received after PCS assertion. 0b - Subsequent data word or RX FIFO is empty (RXEMPTY=1). 1b - First data word

68.4.5.1.19 Receive Data (RDR)

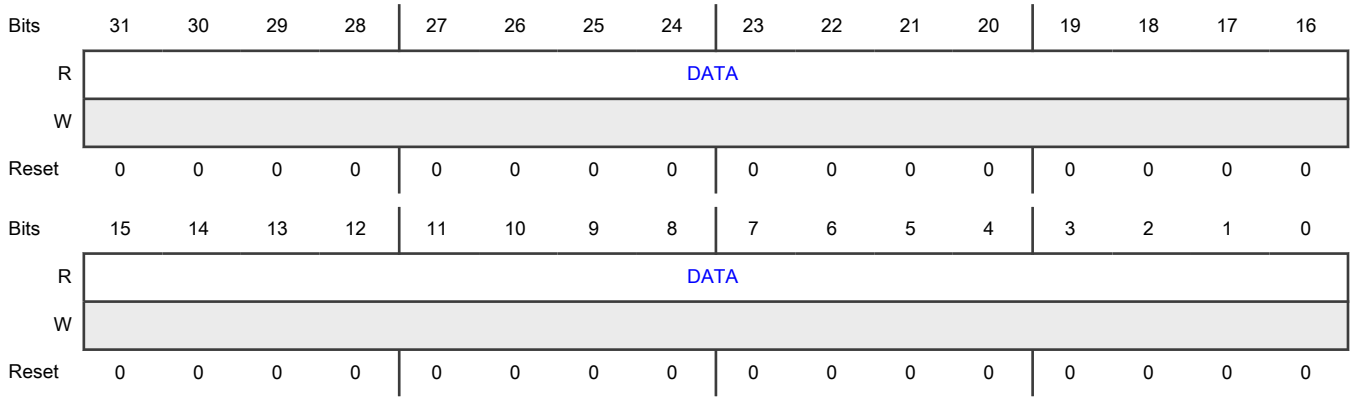
Offset

Register	Offset
RDR	74h

Function

Pulls the first entry from the receive FIFO.

Diagram



Fields

Field	Function
31-0 DATA	Receive Data

68.4.5.1.20 Receive Data Read Only (RDROR)

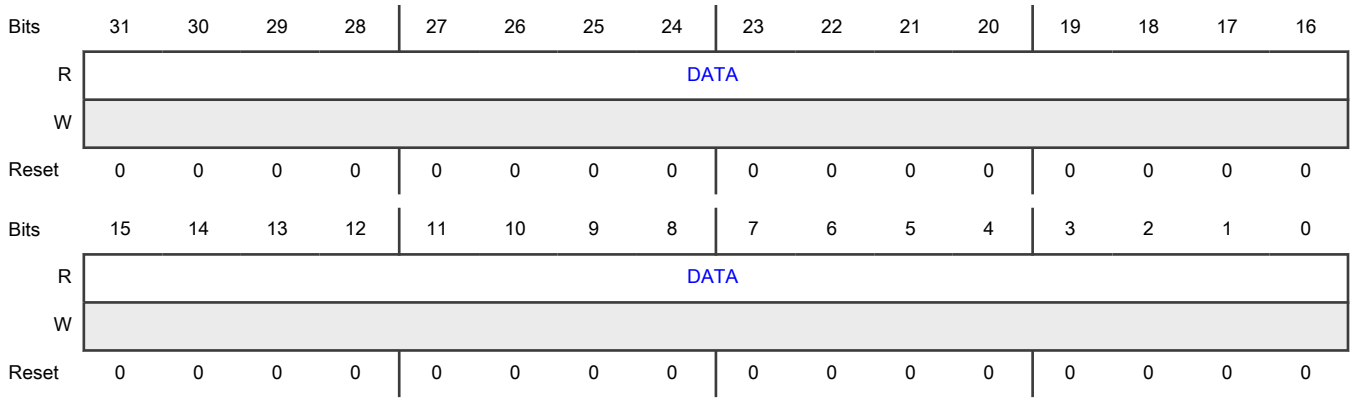
Offset

Register	Offset
RDROR	78h

Function

Returns the first entry in the receive FIFO but does not remove the data from the FIFO.

Diagram



Fields

Field	Function
31-0 DATA	Receive Data

68.4.5.1.21 Transmit Command Burst (TCBR)

Offset

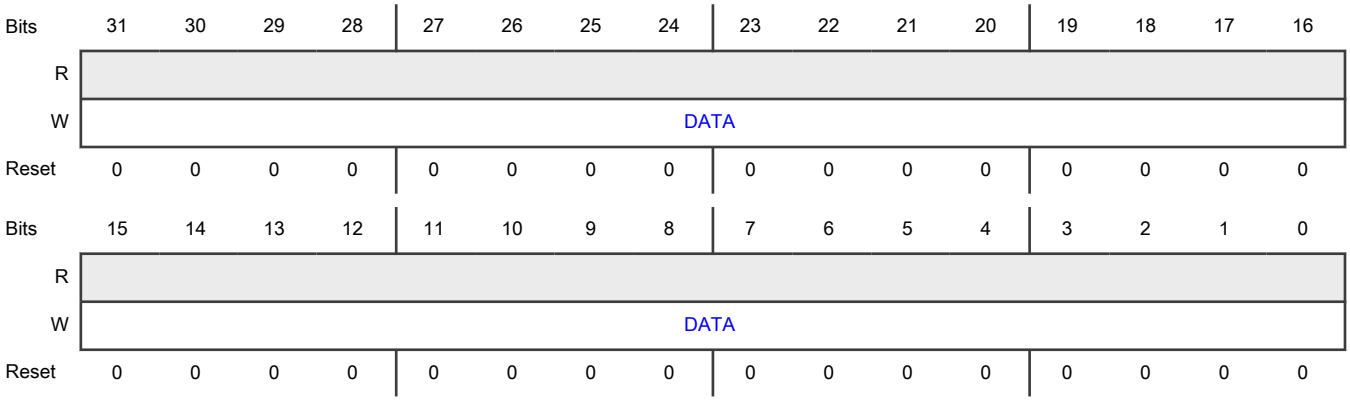
Register	Offset
TCBR	3FCh

Function

Supports burst transfers of command data to the transmit FIFO for use with the DMA controller.

See DMA support registers.

Diagram



Fields

Field	Function
31-0 DATA	Command Data Writes data to Transmit Command (TCR).

68.4.5.1.22 Transmit Data Burst (TDBR0 - TDBR127)

Offset

For n = 0 to 127:

Register	Offset
TDBRn	400h + (n × 4h)

Function

Supports burst transfers of data to the transmit FIFO for use with the DMA controller. The size of this register is 512 bytes.

See DMA support registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Data Writes data to Transmit Data (TDR).

68.4.5.1.23 Receive Data Burst (RDBR0 - RDBR127)

Offset

For n = 0 to 127:

Register	Offset
RDBRn	600h + (n × 4h)

Function

Supports burst transfers of data from the receive FIFO. The size of this register is 512 bytes.

See DMA support registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Data Reads data from Receive Data (RDR).

68.5 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)

68.5.1 Overview

LPUART provides asynchronous, serial communication capabilities with external devices. It supports the non-return-to-zero (NRZ) encoding format and infrared data association (IrDA)-compatible, low-speed serial infrared (SIR) protocol. LPUART can continue operating when the processor is in Low-Power mode, if an appropriate peripheral clock is available.

68.5.1.1 Block diagram

Figure 288 shows the transmitter portion of LPUART.

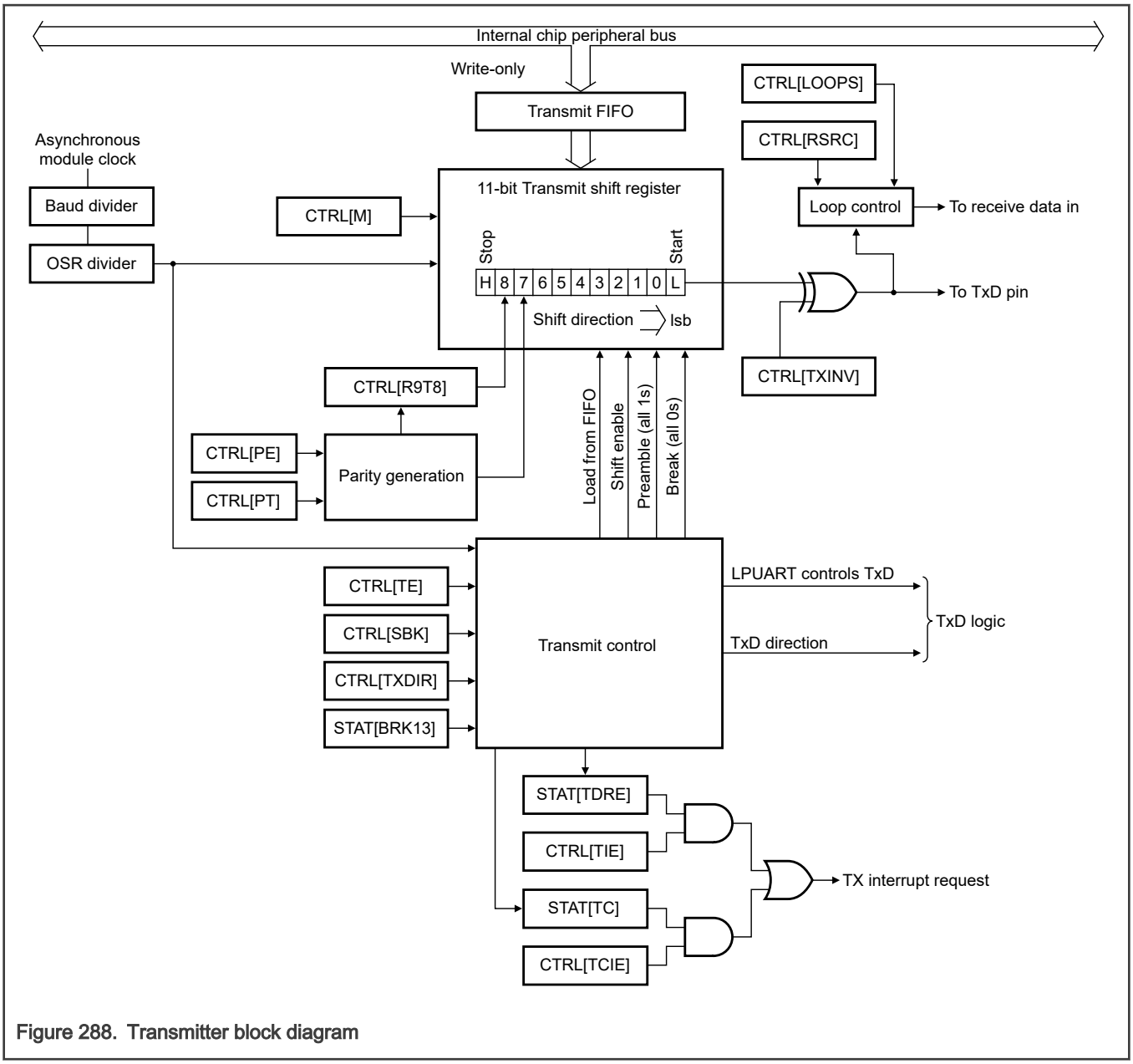


Figure 288. Transmitter block diagram

Figure 289 shows the receiver portion of LPUART.

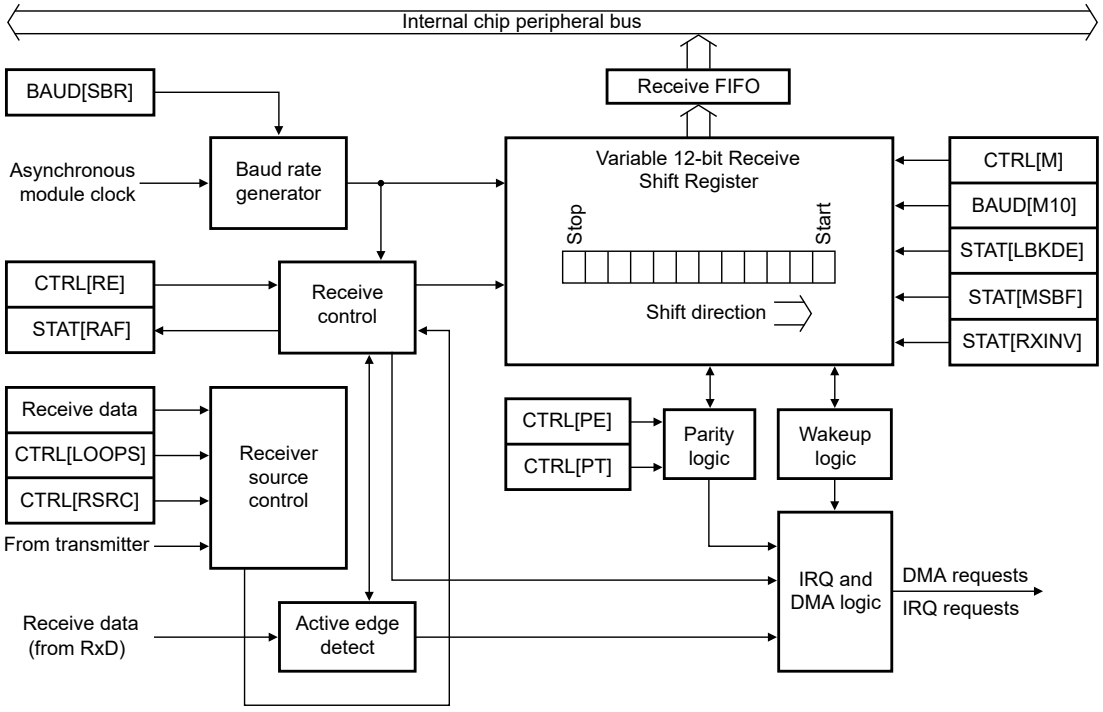


Figure 289. Receiver block diagram

68.5.1.2 Features

- Full-duplex, standard NRZ format
- Programmable baud rates (13-bit modulo divider) with a configurable oversampling ratio (OSR) from 4× to 32×
- Asynchronous operation of transmit and receive baud rates with respect to the bus clock:
  - Baud rate can be configured independently of the bus clock frequency.
  - Operation in Low-Power modes is supported.
- Interrupt, DMA, or polled operations:
  - Transmit data empty and transmission complete
  - Receive data full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
  - Receive data match
- Hardware parity generation and checking
- Programmable 7-bit, 8-bit, 9-bit, or 10-bit character length
- Programmable 1-bit or 2-bit stop bits
- Support for three receiver wake-up methods:
  - Idle line wake-up

  - Address mark wake-up
  - Receive data match
- Automatic address matching to reduce ISR overhead:
  - Address mark matching
  - Idle line address matching
  - Address match start, address match end
- Optional 13-bit and 11-bit break character generation
- Configurable idle length detection supporting 1, 2, 4, 8, 16, 32, 64, or 128 idle characters
- Selectable transmitter output and receiver input polarity
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with a programmable pulse width
- Independent FIFO structure for transmit and receive functions:
  - Separate configurable watermarks for receive and transmit requests
  - Option for receiver to assert request after a configurable number of idle characters, if receive FIFO is not empty

68.5.2 Functional description

LPUART supports full-duplex, asynchronous, NRZ serial communication and comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. The following sections describe all LPUART blocks.

68.5.2.1 Baud rate generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and transmitter. The value, ranging from 1 to 8191, written to BAUD[SBR] determines the baud clock divisor for the asynchronous LPUART baud clock. The baud rate clock drives the receiver, while a bit clock, generated from the baud rate clock divided by the OSR, drives the transmitter. Depending on the OSR, the receiver has an acquisition rate of 4 to 32 samples per bit time. LPUART requires BAUD[SBR] and BAUD[OSR] to accurately match the baud rate requirement. There is a relationship between the required baud rate and the input clock frequency, SBR, and OSR. See the below figure for details.

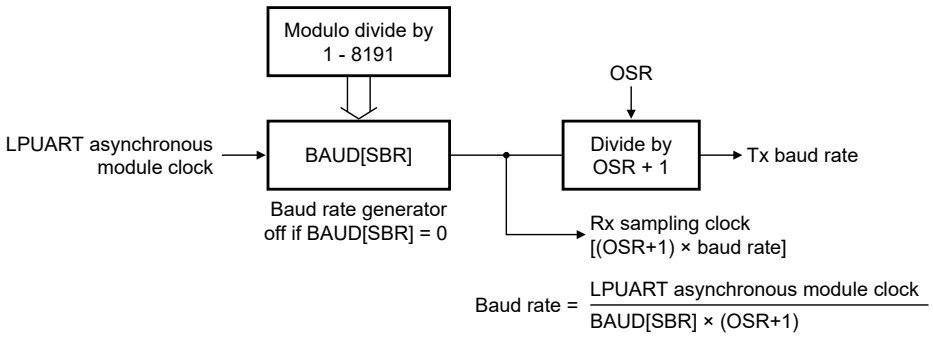


Figure 290. Baud rate generation

Baud rate generation is subject to these sources of error:

- Integer division of the asynchronous LPUART baud clock may not give the exact target frequency.
- Synchronization with the asynchronous LPUART baud clock can lead to a phase shift.

Baud rate generation is a free-running counter that continues whenever the transmitter or receiver is enabled. The transmitter bit clock continues whenever the transmitter is enabled; each transmitted character aligns to the next edge of the transmit bit clock.

In general, configuring OSR for a higher ratio and/or sampling on both edges of the clock slightly improves LPUART's tolerance to baud rate mismatch between the received data and LPUART configured baud rate. However, the three data samples in each bit (see Data sampling technique) are also closer together, which may impact noise sensitivity.

68.5.2.2 Baud rate tolerance

A transmitting device may operate at a baud rate below or above that of the receiver.

Accumulated bit time misalignment can cause one of the three stop bit data samples to fall outside the actual stop bit. A noise error will occur if the three samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the three stop bit samples are a logic zero.

As the receiver samples an incoming frame, it may resynchronize the oversampling clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

In general, increasing the number of samples per bit will increase the baud rate tolerance and decreasing the number of samples per bit will reduce the baud rate tolerance. Note that since LPUART implements triple voting on consecutive receive data samples, increasing the number of samples per bit will move those samples closer together which would reduce the width of noise that can be filtered by the triple voting logic.

68.5.2.3 Calculating baud rate tolerance

Using the following definitions:

- SAM is the number of sample points per bit (valid range from 8 to 32; equal to (OSR + 1) × (BOTHEDGE + 1)).
- BIT is the number of bits in a character including start, data and stop bits (valid range from 9 to 13).

The ideal baud rate tolerance can be calculated as follows:

- Slow data rate tolerance = ((SAM ÷ 2) - 1) ÷ ((SAM × BIT) - (SAM ÷ 2) + 2)
- Fast data rate tolerance = ((SAM ÷ 2) - 2) ÷ (SAM × BIT)

As an example, if configured for 8-bit data, 1 stop bit (BIT = 10) and with OSR=0x7 and BOTHEDGE = 1 (SAM = 16):

- Slow data rate tolerance = (8 - 1) ÷ (160 - 8 + 2) = 7 ÷ 154 = 4.54%
- Fast data rate tolerance = (8 - 2) ÷ 160 = 6 ÷ 160 = 3.75%

If configured for 9-bit data with 1 stop bit (BIT = 11) with same oversampling configuration, then:

- Slow data rate tolerance = (8 - 1) ÷ (176 - 8 + 2) = 7 ÷ 170 = 4.12%
- Fast data rate tolerance = (8 - 2) ÷ 176 = 6 ÷ 176 = 3.41%

NOTE

Additional factors can contribute to a lower baud rate tolerance than the ideal. These include clock uncertainty or jitter on the LPUART functional clock source, differences in rise and fall times on the transmitter output and synchronization of the external receive pin to the local LPUART functional clock.

68.5.2.4 Transmitter functional description

This section describes the functioning of the LPUART transmitter, as shown in the transmitter portion of Block diagram, as well as specialized functions for sending break and idle characters.

The transmitter output (TXD) idle state defaults to logic high; the transmitter output is inverted when you write 1 to CTRL[TXINV], which becomes 0 following reset. You can enable the transmitter by writing 1 to CTRL[TE]. This queues a preamble character that is one full character frame of the Idle state. The transmitter then remains idle until data is available in the transmit FIFO and programs store data in the transmit FIFO by writing to Data (DATA).

The central element of the LPUART transmitter is the transmit shift register that is 9-bit to 13-bit long depending on the settings of CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS]. Going forward in this discussion, assume that CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] are 0, selecting the normal 8-bit Data mode, in which the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new character, the value waiting in transmit FIFO is transferred to the transmit shift register, synchronized with the baud rate clock, and STAT[TDRE] becomes 1 to indicate that another character may be written to the transmit FIFO at Data (DATA).

If no new character is waiting in the transmit FIFO after a stop bit is shifted out of the TXD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TXD high, waiting for more characters to transmit.

Writing 0 to CTRL[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed (that could include a data character, idle character, or break character), although the transmitter does not start transmitting another character.

68.5.2.4.1 Break character length

CTRL[SBK] sends break characters, originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 9-bit to 12-bit times, including the start and stop bits. You can enable a longer break of 13-bit times by writing 1 to STAT[BRK13]. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, the program writes 1 and then writes 0 to CTRL[SBK]. This action queues a break character to be sent as soon as the shifter is available. If CTRL[SBK] remains 1 when the queued break moves into the shifter, synchronized with the baud rate clock, an additional break character is queued. When LPUART is the receiving module, it receives a break character as 0s in all data bits and a framing error (STAT[FE] = 1) is detected.

You can also transmit a break character by writing to Data (DATA) with DATA[FRETSC] = 1 and the data bits clear. This supports transmitting the break character as part of the normal data stream and also allows DMA to transmit a break character.

When idle line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, write 0 and then write 1 to CTRL[TE]. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while CTRL[TE] becomes 0, the LPUART transmitter does not release control of the TXD pin.

You can also write to Data (DATA) to transmit an idle character, with DATA[FRETSC] and DATA[R9T9] = 1 and the values of all the other fields = 0. This supports transmitting the idle character as part of the normal data stream and also allows DMA to transmit an idle character.

As shown in the following table, STAT[BRK13], CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] affect the length of the break character.

Table 664. Break character length

STAT[BRK13]	CTRL[M]	BAUD[M10]	CTRL[M7]	BAUD[SBNS]	Break character length (in bit times)
0	0	0	0	0	10
0	0	0	0	1	11
0	0	0	1	0	9
0	0	0	1	1	10
0	1	0	—	0	11
0	1	0	—	1	12
0	—	1	—	0	12
0	—	1	—	1	13
1	0	0	0	0	13

Table continues on the next page...

Table 664. Break character length (continued)

STAT[BRK13]	CTRL[M]	BAUD[M10]	CTRL[M7]	BAUD[SBNS]	Break character length (in bit times)
1	0	0	0	1	13
1	0	0	1	0	12
1	0	0	1	1	12
1	1	0	—	0	14
1	1	0	—	1	14
1	—	1	—	0	15
1	—	1	—	1	15

68.5.2.4.2 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS_B. If the CTS operation is enabled, the character is transmitted when CTS_B is asserted. If CTS_B is deasserted in the middle of a transmission with characters remaining in the transmitter FIFO, the character in the transmit shift register is complete. Any characters in the FIFO wait for CTS_B to assert again, and TXD remains in the mark state (idle state) until CTS_B is reasserted. The CTS_B pin must assert for longer than one bit period to guarantee that a new transmission is started when the transmitter is idle with CTS.

If the CTS operation is disabled, the transmitter ignores the state of CTS_B.

The transmitter's CTS_B signal can be enabled even if the same LPUART receiver's RTS_B signal is disabled.

68.5.2.4.3 Transceiver driver enable

The transmitter can use RTS_B as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS_B for details. If the RTS operation is enabled, when a character is placed into an empty transmit shift register, RTS_B asserts 1-bit time before the start bit is transmitted. RTS_B remains asserted for the whole time that the transmit shift register has any characters. RTS_B deasserts 1-bit time after all characters in the transmit FIFO and shift register are completely sent, including the last stop bit. In other words, when RTS_B is used as a transceiver enable, RTS_B asserts 1-bit time before the transmitter starts transmitting and negates 1-bit time after the transmitter goes idle.

Transmitting a break character also asserts RTS_B, with the same assertion and deassertion timing as having a character in the transmit shift register.

The transmitter's RTS_B signal asserts only when the transmitter is enabled. However, the transmitter's RTS_B signal is unaffected by its CTS_B signal. RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled mid-way through a data transfer.

68.5.2.4.4 Transceiver driver enable using RTS_B

RS-485 is a multiple drop communication protocol in which the LPUART transceiver's driver is three-stated unless LPUART is driving. The transmitter can use the RTS_B signal to enable the driver of a transceiver. The polarity of RTS_B can be matched to the polarity of the transceiver's driver enable signal.

The following figure shows the receiver enable signal asserted. This connection can also connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled when driving. A pullup can pull RXD to a nonfloating value during this time. You can refine this option further by operating LPUART in Single-Wire mode, freeing the RXD pin for other uses.

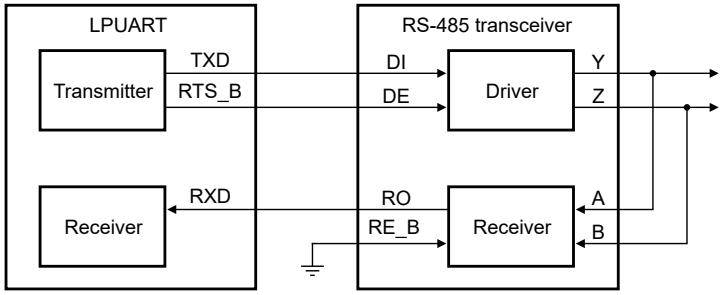


Figure 291. Transceiver driver enable using RTS_B

68.5.2.5 Receiver functional description

This section discusses the functioning of the LPUART receiver, as shown in the receiver portion of Block diagram. The section also discusses:

- The data sampling technique used to reconstruct receiver data.
- Different variations of the receiver wake-up function.

You can invert the receiver input by writing 1 to STAT[RXINV] and enable the receiver by writing 1 to CTRL[RE]. Character frames consist of a start bit of logic 0, along with N (7, 8, 9, 10) bits (MSB or LSB first), and one or two stop bits of logic 1. For information about 7-bit, 9-bit, or 10-bit Data mode, see Data modes. Going forward in this discussion, assume that LPUART is configured for a normal 8-bit Data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full (STAT[RDRF] = 0), the data character is transferred to the receive FIFO, resulting in STAT[RDRF] becoming 1. However, if STAT[RDRF] is already 1, indicating that the receive data buffer is already full, STAT[OR] becomes 1 and the new data is lost.

Because the LPUART receiver is separate from the receive FIFO, the receive shift register can receive the next word when the receive FIFO is full, and it is only at the end of the character that the next data is written into the receive FIFO, potentially triggering the overrun flag if the FIFO is full.

When a program detects that the receive data register is full (STAT[RDRF] = 1), it gets the data from the FIFO by reading Data (DATA). See Interrupts for details about flag clearing.

68.5.2.5.1 Data sampling technique

The LPUART receiver supports a configurable oversampling rate of between 4× and 32× of the baud rate clock for sampling. The receiver starts by considering logic level samples at the oversampling rate times the baud rate to search for a falling edge on the RXD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is located, three more samples are taken at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2 to ensure that this is a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes they are synchronized to a received character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2, to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, noise flag (STAT[NF]) becomes 1 when the received character is transferred to the receive FIFO.

When the LPUART receiver is configured to sample on both edges of the baud rate clock (that is, when BAUD[BOTHEDGE] = 1), the number of segments in each received bit is effectively doubled (from 1 to OSR× 2). The start and data bits are then sampled at OSR, OSR + 1, and OSR + 2. You must enable sampling on both edges of the clock for oversampling rates of 4× to 7×. This sampling is optional for higher oversampling rates.

The synchronization feature of LPUART synchronizes the internal oversampling counter with a detected falling edge on the receive signal, and to adjust the data sampling window. The falling edge detection needs three consecutive 1s prior to the "1->0" (one to zero) transition. After the initial falling edge detection for the start bit, the circuit continuously monitors the next falling edge, and resets the counter after another falling edge is detected. This synchronization to the start bit is termed as resynchronization.

When BAUD[RESYNCDIS] is 0, you perform this falling edge detection and resynchronization not only for the start bit but also for the rest of the character reception after the start bit.

When BAUD[RESYNCDIS] is 1, you perform the falling edge detection and resynchronization only for the start bit. The use case for disabling the resynchronization is protocols that require this (for example, LIN 2.1 prohibits resynchronization within a byte).

The following table and figure explain LPUART resynchronization.

Table 665. LPUART resynchronization settings

Resynchronization	BAUD[RESYNCDIS] = 0	BAUD[RESYNCDIS] = 1
For the starting bit falling edge	Yes	Yes
For all falling edges after the start bit	Yes	No

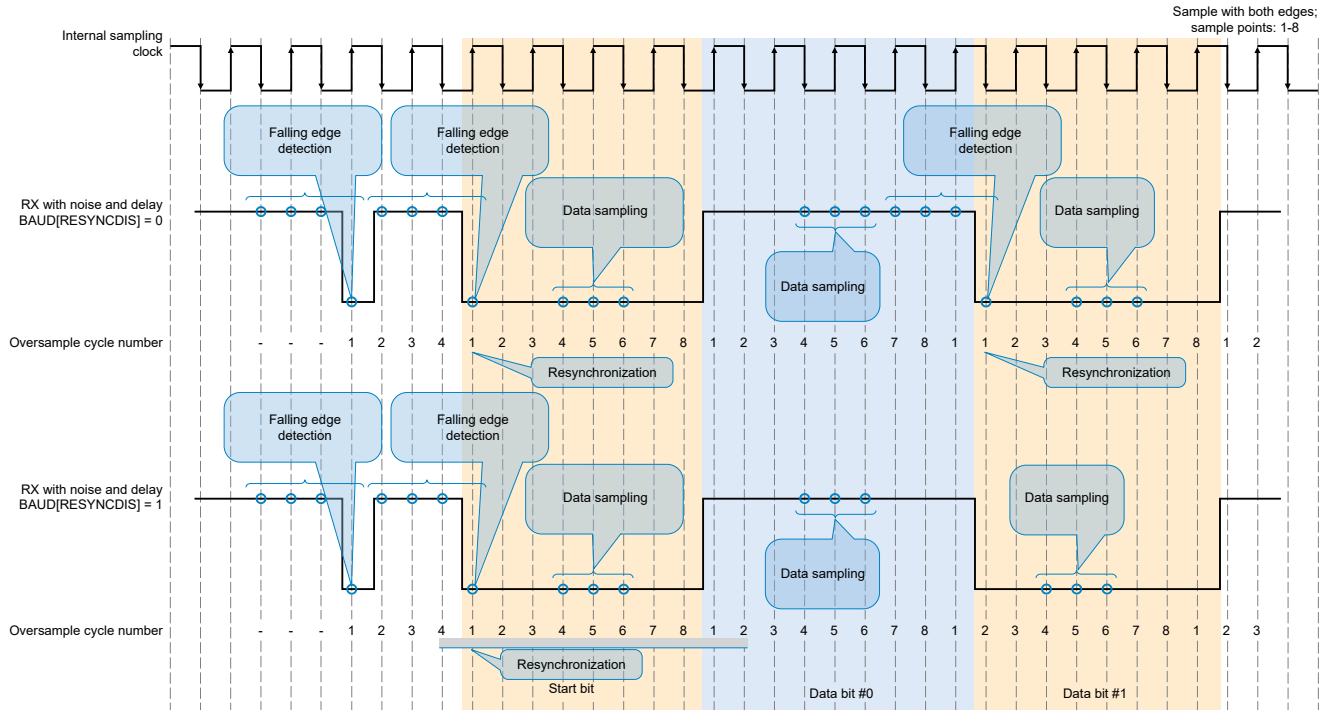


Figure 292. LPUART resynchronization diagram

68.5.2.5.2 Receiver wake-up operation

Receiver wake-up and receiver address matching are hardware mechanisms that allow an LPUART receiver to ignore the characters in a message intended for a different receiver.

During receiver wake-up, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write 1 to CTRL[RWU].

When CTRL[RWU] and STAT[RWUID] are 1, the status fields associated with the receiver, with the exception of STAT[IDLE], are inhibited from becoming 1, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, all receivers automatically force CTRL[RWU] to become 0. This results in all receivers waking up in time to look at the first character(s) of the next message.

During receiver address matching, the address matching is performed in hardware and the LPUART receiver ignores all characters that do not meet the address match requirements.

Table 666. Receiver wake-up options

CTRL[RWU]	BAUD[MAEN1] | BAUD[MAEN2]	BAUD[MATCFG]	CTRL[WAKE]: STAT[RWUID]	Receiver wake-up
0	0	X	X	Normal operation
1	0	00	00	Receiver wake-up on idle line; STAT[IDLE] = 0
1	0	00	01	Receiver wake-up on idle line; STAT[IDLE] = 1
1	0	00	10	Receiver wake-up on address mark
1	1	11	10	Receiver wake-up on data match
0	1	00	X0	Address mark address match; STAT[IDLE] = 0 for discarded characters
0	1	00	X1	Address mark address match; STAT[IDLE] = 1 for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Match on and match off; STAT[IDLE] = 0 for discarded characters
0	1	10	X1	Match on and match off; STAT[IDLE] = 1 for discarded characters

68.5.2.5.2.1 Idle line wake-up

When CTRL[WAKE] is 0, you can configure the receiver for an idle line wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a full character time of the idle-line level.

When CTRL[RWU] is set the receiver ignores everything until an idle condition is detected. After the idle condition is detected the CTRL[RWU] is cleared by hardware and the receiver starts receiving data.

The STAT[RWUID] indicates that whether the STAT[IDLE] flag is set or not as a result of the idle condition detection which clears CTRL[RWU].

The STAT[RWUID] says that if the STAT[IDLE] flag is to be set as a result of the idle condition which clears CTRL[RWU].

CTRL[M], CTRL[M7], and BAUD[M10] select 7-bit to 10-bit Data mode and BAUD[SBNS] selects a 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 9 to 13 bit times because of the start and stop bits.

When CTRL[RWU] is 1 and STAT[RWUID] is 0, the idle condition that wakes up the receiver does not lead to STAT[IDLE] becoming 1. The receiver wakes up and waits for the first data character of the next message that leads to STAT[RDRF] becoming 1 and generates an interrupt if enabled. When STAT[RWUID] is 1, any idle condition leads to STAT[IDLE] becoming 1 and generates an interrupt if enabled, regardless of whether CTRL[RWU] is 0 or 1.

These are the ways to detect an idle line:

- When CTRL[ILT] is 0, the idle bit counter starts after the start bit so that the stop bit and any logic 1s at the end of a character count to calculate the full character time of idle.
- When CTRL[ILT] is 1, the idle bit counter does not start until after the stop bit time so that the data in the last character of the previous message does not impact the idle detection.

68.5.2.5.2.2 Address mark wake-up

When CTRL[WAKE] is 1, you can configure the receiver for an address mark wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a logic 1 in the most significant bit of the received character. When parity is enabled, the second most significant bit is used for address mark wake-up.

Address mark wake-up allows messages to contain idle characters, but requires one bit to be reserved for use in address frames. The logic 1 in the most significant bit (or second most significant bit when parity is enabled) of an address frame writes 0 to CTRL[RWU] and writes 1 to STAT[RDRF]. In this case, the character with the address mark bit is received even if the receiver is sleeping during most of this character time.

68.5.2.5.2.3 Data match wake-up

When CTRL[RWU] and CTRL[WAKE] are 1, and BAUD[MATCFG] equals 11, the receiver is configured for a data match wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a character that matches MATCH[MA1] when BAUD[MAEN1] is 1, or that matches MATCH[MA2] when BAUD[MAEN2] is 1.

68.5.2.5.2.4 Address match operation

You can enable the address match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 0. In this function, a character that the RXD pin receives with a logic 1 in the most significant bit (or the second most significant bit when parity is enabled) is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is only transferred to the receive buffer, and STAT[RDRF] becomes 1 if the comparison matches. All subsequent characters received with a logic 0 in the most significant bit (or the second most significant bit when parity is enabled) are considered to be data associated with the address and are transferred to the receive FIFO. If no marked address match occurs, no transfer is made to the receive FIFO, and all the characters that follow, with logic 0 in the most significant bit (or second most significant bit when parity is enabled), are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

The address match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, a marked address is compared only to the associated Match Address (MATCH) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, a marked address is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

68.5.2.5.2.5 Idle match operation

You can enable the idle match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 1. In this function, the first character that the RXD pin receives after an idle line condition is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is transferred only to the receive buffer, and STAT[RDRF] becomes 1, if the comparison matches. All subsequent characters are considered to be data associated with the address and are transferred

to the receive FIFO until the next idle line condition is detected. If no address match occurs, no transfer is made to the receive FIFO, and all the frames that follow, until the next idle condition, are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

An idle match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, the first character after an idle line is compared only to the associated Data (DATA) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, the first character after an idle line is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

68.5.2.5.2.6 Match on, match off operation

The match on, match off operation is enabled when both BAUD[MAEN1] and BAUD[MAEN2] are 1 and BAUD[MATCFG] = 10. In this function, a character that the RXD pin receives matches MATCH[MA1] and is transferred to the receive buffer, and STAT[RDRF] becomes 1. All subsequent characters are considered to be data and are also transferred to the receive FIFO, until a character that matches MATCH[MA2] is received. The character that matches MATCH[MA2], along with all subsequent characters, is discarded; and this continues until another character that matches MATCH[MA1] is received. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

NOTE

The match on, match off operation requires both BAUD[MAEN1] and BAUD[MAEN2] to be 1.

68.5.2.5.3 Hardware flow control

To support hardware flow control, you can program the receiver to automatically assert and deassert RTS_B:

- RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS_B for more information.
- If the receiver RTS functionality is enabled, the receiver automatically deasserts RTS_B if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1.
- The receiver asserts RTS_B when STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. There is no impact if STAT[RDRF] is 1 already.
- Even if RTS_B is deasserted, the receiver continues to receive characters until the receive FIFO is overrun.
- If the receiver RTS functionality is disabled, the receiver's RTS_B remains deasserted.
- When RTS is driven by receiver, it will be negated when the receiver is disabled irrespective of the FIFO level.

68.5.2.6 Additional LPUART functions

68.5.2.6.1 Data modes

You can configure the LPUART transmitter and receiver to operate in 7-bit Data mode by writing 1 to CTRL[M7], 9-bit Data mode by writing 1 to CTRL[M], or 10-bit Data mode by writing 1 to BAUD[M10]. In 9-bit Data mode, there exists a ninth data bit and in 10-bit mode, there exists a tenth data bit.

When performing 8-bit writes to the transmit FIFO, the ninth and tenth bits are pushed into the FIFO from CTRL[T8] and CTRL[T9]. For coherent 8-bit writes, you must write to CTRL[T8] and CTRL[T9] before writing to Data (DATA)[7:0]. However, if the values in CTRL[T8] or CTRL[T9] do not need to change, it is not necessary to update CTRL[T8] and CTRL[T9] before every 8-bit write to Data (DATA).

When performing 16-bit or 32-bit writes to the transmit FIFO, all 10 bits are pushed into the transmit FIFO from the write data.

When performing 8-bit reads of the receive FIFO, the ninth and tenth bits are held in CTRL[R8] and CTRL[R9] but you must read them before reading Data (DATA). A 16-bit or 32-bit read of the receive FIFO returns all 10 bits in Data (DATA).

The 9-bit Data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with the address mark wake-up so that the ninth data bit can serve as the wake-up bit. The 10-bit Data mode is typically used with parity and address mark wake-up so that the ninth data bit can serve as the wake-up bit and the tenth bit can serve as the parity bit. In custom protocols, the ninth and/or tenth bits can also serve as software-controlled markers.

68.5.2.6.2 Idle length

An idle character is one where the start bit, all data bits, and stop bits are in the mark position (idle state, generally logic 1). You can configure CTRL[ILT] to start detecting an idle character from the previous start bit (any data bits and stop bits count for idle character detection) or from the previous stop bit.

You can also use CTRL[IDLECFG] to configure the number of idle characters that must be received before an idle line condition is detected. This field configures the number of idle characters that must be received before STAT[IDLE] becomes 1, STAT[RAF] becomes 0, and DATA[IDLINE] becomes 1 with the next received character.

CTRL[IDLECFG] also affects the idle line wake-up and idle match operations. When either the address match or match on/off operation is enabled, writing 1 to STAT[RWUID] causes any discarded characters to be treated as idle characters.

68.5.2.6.3 Loop mode

Enable Loop mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 0. You, sometimes, use Loop mode to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and LPUART does not use the RXD pin.

Loop mode also internally connects the RTS_B output to the CTS_B input.

68.5.2.6.4 Single-Wire mode

Enable Single-Wire mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 1. Single-Wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and TXD pin (the RXD pin is not used).

In Single-Wire mode, CTRL[TXDIR] controls the direction of serial data on the TXD pin. When CTRL[TXDIR] becomes 0, the TXD pin is an input to the receiver and the transmitter is temporarily disconnected from the TXD pin so that an external device can send serial data to the receiver. When CTRL[TXDIR] = 1, the TXD pin is an output that the transmitter drives. The internal loop back connection is disabled, and as a result, the receiver is unable to receive characters that the transmitter sends out.

68.5.2.7 Peripheral triggers

The connection of the LPUART peripheral triggers with other peripherals is chip-specific.

68.5.2.7.1 Output triggers

LPUART generates the following output triggers that can be connected to other peripherals on the chip:

- The transmit word trigger asserts at the end of each transmitted word and negates after 1-bit period.
- The transmit data trigger is identical to the TXD pin output, but without support for input trigger modulation.
- The receive word trigger asserts at the end of each received word that is written to the receive FIFO, for one oversampling clock period.
- The receive idle trigger asserts when STAT[IDLE] becomes 1, and negates when the next valid start bit is detected.

68.5.2.7.2 Input trigger

LPUART supports a peripheral input trigger that you can configure in one of the following ways:

- By enabling the CTS function: You can connect the input trigger (active high) instead of the CTS_B (active low) pin input. The input trigger must assert for longer than 1-bit clock period when the transmitter is idle, with data to send, to guarantee a new transmission.

- By making the input trigger modulate the transmit data output (trigger is logically ANDed with the TXD output): The input trigger is expected to be a free-running clock (carrier signal) that generates from a timer or PWM source with a frequency that is greater than the bit-clock frequency. The carrier signal must not toggle faster than the maximum supported bit time.
- By connecting the input trigger instead of the RXD pin input: The input trigger is expected to be generated from a receive data source, such as an analog comparator or external pin.

68.5.2.8 Infrared (IR) interface

LPUART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses, transforming them to serial bits, which are then sent to LPUART. The IrDA physical layer specification defines a half-duplex IR communication link for exchanging data. The full standard includes data rates up to 16 Mbit/s. The LPUART IrDA support is limited to SIR mode that supports data rates only between 2.4 kbit/s and 115.2 kbit/s.

LPUART has an infrared transmit encoder and a receive decoder. The infrared decoder converts the received character from the IrDA format to the NRZ format, which the receiver uses. It also has an OSR oversampling baud rate clock counter that filters noise and indicates when a 1 is received. LPUART transmits serial bits of data, which the infrared submodule encodes, to transmit a narrow pulse for every zero bit. No pulse is transmitted for every single bit. When receiving data, an IR photo diode (external to LPUART) detects the IR pulses. The IR receive decoder transforms them to CMOS levels. The infrared receive decoder then stretches the narrow pulses to get back to a serial bit stream that LPUART receives. You can invert the polarity of transmitted pulses and expected receive pulses so that a direct connection can be made to external IrDA transceiver modules that use active-high pulses.

The IR submodule receives its clock sources from LPUART. The submodule selects one of these clocks to generate either 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulses during transmission.

68.5.2.8.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from the transmit shift register to the TXD signal. A narrow pulse is transmitted for a 0 bit and no pulse is transmitted for a 1 bit. The narrow pulse is sent at the start of the bit with a duration of 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR of a bit time. A narrow low pulse is transmitted for a 0 bit when CTRL[TXINV] is 0, while a narrow high pulse is transmitted for a 0 bit when CTRL[TXINV] is 1.

68.5.2.8.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each 0 received and no pulse is expected for each 1 received. A narrow low pulse is expected for a 0 bit when STAT[RXINV] is 0, while a narrow high pulse is expected for a 0 bit when STAT[RXINV] is 1. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

68.5.2.8.3 Start-bit detection

When STAT[RXINV] is 0, the first falling edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently of each other.

68.5.2.8.4 Noise filtering

The decoder ignores any rising edges detected during the first half of the infrared decoder counter, and can leave any pulses less than one oversampling baud clock as undetected. This is regardless of whether the pulse is seen in the first or second half of the count.

68.5.2.8.5 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as 0, which is sent to the receiver. The decoder counter is also reset.

68.5.2.8.6 High-bit detection

At OSR oversampling baud rate clocks after the previous rising edge, if a rising edge is not seen, the decoder sends a 1 to the receiver.

If the next bit is 0, which arrives late, a low bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, the delay of a 0 is not recorded as noise.

68.5.2.9 Modes of operation

68.5.2.9.1 Low-Power modes

LPUART remains functional during low-power modes, provided CTRL[DOZEEN] is 0 and the LPUART functional clock is enabled. LPUART can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPUART to be disabled in low-power modes, when CTRL[DOZEEN] is 1. In this case, the transmitter and receiver finish transmitting and receiving the current word.

If LPUART is disabled in low-power modes, it can generate a wake-up via STAT[RXEDGIF] if the receiver detects an active edge.

NOTE

See the chip-specific information for specific low-power modes available on your chip.

68.5.2.9.2 Debug mode

LPUART remains functional in Debug mode.

68.5.2.10 Clocking

Table 667. Types of clocks

Clock	Description
Functional	Is asynchronous to the bus clock and can remain enabled in Low-Power modes to support transmit and/or receive functions, including low-power wake-up.
Bus	Is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPUART transmit and receive registers, including the FIFOs.

68.5.2.11 Reset

Table 668. Types of resets

Reset	Description
Chip	Enables the logic and registers for the LPUART transmitter and receiver to reset to their default states.
Software	Resets the LPUART logic and registers to their default states, except for Global (GLOBAL). GLOBAL[RST] controls the LPUART software reset.
FIFO	Implements write-only control fields that reset the transmit FIFO (FIFO[TXFLUSH]) and receive FIFO (FIFO[RXFLUSH]). After a FIFO is reset, that FIFO becomes empty.

68.5.2.12 Interrupts

The LPUART transmitter has two status fields that can optionally generate hardware interrupt requests. If STAT[TDRE] is 1, it indicates that there is room in the transmit FIFO to write another transmit character to Data (DATA). If CTRL[TIE] is 1, a hardware interrupt is requested when STAT[TDRE] is 1.

STAT[TC] indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TXD at the inactive level. This field is often used in systems with modems to determine when it is safe to turn off the modem. If CTRL[TCIE] is 1, a hardware interrupt is requested when STAT[TC] is 1. Instead of hardware interrupts, software polling may be used to monitor STAT[TDRE] and STAT[TC] if the corresponding CTRL[TIE] or CTRL[TCIE] field is 0.

When a program detects that STAT[RDRF] is 1, it gets the data from this field by reading Data (DATA). The field becomes 0 by reading Data (DATA).

STAT[IDLE] includes logic that prevents it from becoming 1 repeatedly when the RXD line remains idle for an extended period of time. STAT[IDLE] becomes 0 when you write 1 to it, and cannot become 1 again until the receiver has received at least one new character and has 1 as the value of STAT[RDRF].

If the associated error is detected in the received character that caused STAT[RDRF] to become 1, STAT[NF], STAT[FE], and STAT[PF] become 1 at the same time STAT[RDRF] becomes 1. These flags do not become 1 in overrun cases.

If STAT[RDRF] is already 1 when a new character is ready to be transferred from the receive shifter to the receive FIFO, STAT[OR] becomes 1, instead of the data along with any associated STAT[NF], STAT[FE], or STAT[PF] condition getting lost.

If the received character matches the contents of MATCH[MA1] and/or MATCH[MA2], then STAT[MA1F] and/or STAT[MA2F] become 1 at the same time that STAT[RDRF] becomes 1.

At any time, an active edge on the RXD serial data input pin causes STAT[RXEDGIF] to become 1. STAT[RXEDGIF] becomes 0 when you write 1 to it. This function depends on the receiver being enabled (the value of CTRL[RE] being 1).

68.5.2.13 DMA

68.5.2.13.1 DMA burst support

To support efficient DMA transfers to the transmit FIFO, two alias regions are implemented to support incrementing 8-bit, 16-bit, or 32-bit write accesses to the transmit FIFO:

- Transmit Command Burst (TCBR0 - TCBR127) is a 512-byte region that supports pushing 16-bit data into the transmit FIFO.
- Transmit Data Burst (TDBR0 - TDBR255) is a 1024-byte region that supports pushing zero extended 8-bit data into the transmit FIFO.

The aforementioned regions are contiguous, so a DMA transfer can start in Transmit Command Burst (TCBR0 - TCBR127) to initialize the transfer including address mark, idle word, or break character, and then complete the transfer in Transmit Data Burst (TDBR0 - TDBR255) with the data to transmit, without changing the transfer size.

The transmit FIFO block writes overflow the FIFO, but that does not signal an error. Do not perform 32-bit writes to Transmit Data Burst (TDBR0 - TDBR255) unless there are four empty slots in the transmit FIFO, and do not perform 16-bit writes to this register unless there are two empty slots in the transmit FIFO.

68.5.2.13.2 End-of-packet DMA transfers

The end-of-packet functionality is designed for serial interfaces where you may not know the size of the transfer in advance and the data is being pushed by an external device. The end-of-packet processing ensures that data does not become stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet is different for each serial interface, but the serial peripheral and DMA process it in the same way. For example, an idle line condition signals the UART end of packet, the Stop and/or Repeated Start condition signals the I2C end of packet, and PCS negation signals the SPI end of packet.

When the serial peripheral is configured to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end of packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop has completed and no additional data is pulled from the receive FIFO. The field that triggered the end-of-packet condition becomes 0 when the minor loop completes following the end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. You can, optionally, save the final destination address to the system memory. The final destination address is also available in the destination address register.

Because the DMA terminates the major loop, no servicing of the receive FIFO occurs until you or the hardware reconfigures the DMA (for example, channel linking or scatter-gather). You must minimize this delay to avoid receiver FIFO overrun. The automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. This is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

68.5.3 External signals

Table 669. External signals

Signal	Description	I/O
TXD	Transmit data: This pin is normally an output, but is an input (tristated) in Single-Wire mode whenever the transmitter is disabled or the transmit direction is configured for receive data.	I/O
RXD	Receive data	I
CTS_B	Clear-to-send	I
RTS_B	Request-to-send	O

68.5.4 Initialization

This module does not require initialization.

68.5.5 LPUART register descriptions

LPUART includes registers to control baud rate, select options, report status, and store transmit and receive data. Access to an address outside the valid memory map generates a bus error.

NOTE

Writing to a read-only (RO) register or reading a write-only (WO) register can cause bus errors. LPUART does not verify whether programmed values in the registers are correct; you must write valid values to them.

68.5.5.1 LPUART memory map

LPUART0 base address: 4011_0000h

LPUART1 base address: 4011_1000h

LPUART2 base address: 4011_2000h

LPUART3 base address: 4011_3000h

LPUART4 base address: 4017_1000h

LPUART5 base address: 4017_2000h

LPUART6 base address: 4017_3000h

LPUART7 base address: 4017_4000h

LPUART8 base address: 4019_9000h

LPUART9 base address: 4019_A000h

LPUART10 base address: 4019_B000h

LPUART11 base address: 4019_C000h

LPUART12 base address: 4019_D000h

LPUART13 base address: 4019_E000h

LPUART17 base address: 4032_6000h

LPUART18 base address: 4032_7000h

LPUART19 base address: 4032_8000h

LPUART20 base address: 4032_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0404_0003h
4h	Parameter (PARAM)	32	R	0000_0303h
8h	Global (GLOBAL)	32	RW	0000_0000h
Ch	Pin Configuration (PINCFG)	32	RW	0000_0000h
10h	Baud Rate (BAUD)	32	RW	0F00_0004h
14h	Status (STAT)	32	RW	00C0_0000h
18h	Control (CTRL)	32	RW	0000_0000h
1Ch	Data (DATA)	32	RW	0000_1000h
20h	Match Address (MATCH)	32	RW	0000_0000h
24h	MODEM IrDA (MODIR)	32	RW	0000_0000h
28h	FIFO (FIFO)	32	RW	00C0_0022h
2Ch	Watermark (WATER)	32	RW	0000_0000h
30h	Data Read-Only (DATARO)	32	R	0000_1000h
200h - 3FCh	Transmit Command Burst (TCBR0 - TCBR127)	32	W	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
400h - 7FCh	Transmit Data Burst (TDBR0 - TDBR255)	32	W	0000_0000h

68.5.5.2 Version ID (VERID)

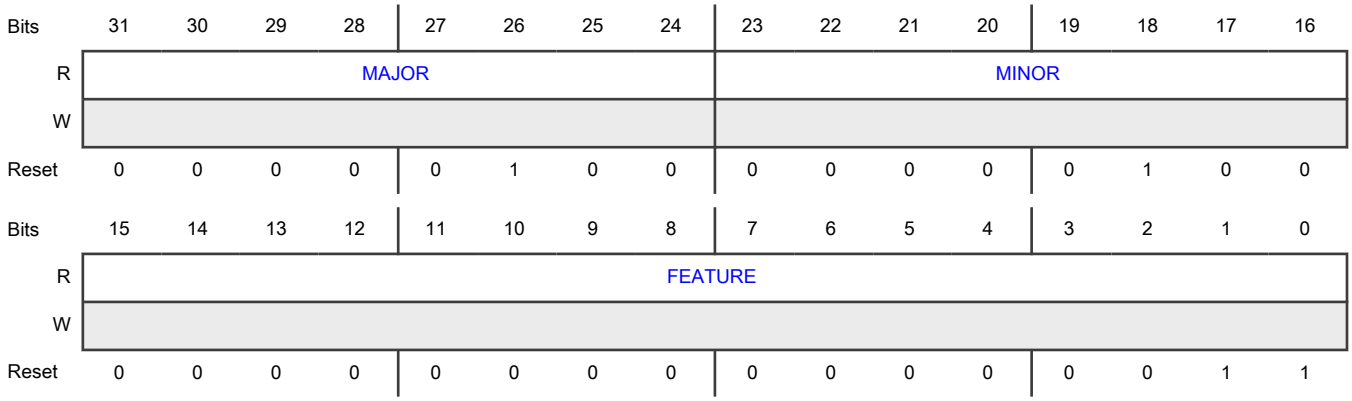
Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the chip and also specifies the inclusion and exclusion of several optional features.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Indicates the feature set number. 0000_0000_0000_0001b - Standard feature set 0000_0000_0000_0011b - Standard feature set with MODEM and IrDA support

68.5.5.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function
Indicates the parameter configuration for this instance on the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Fields

Field	Function
31-16 —	Reserved
15-8 RXFIFO	Receive FIFO Size Indicates the number of characters in the receive FIFO, which is 2^RXFIFO.
7-0 TXFIFO	Transmit FIFO Size Indicates the number of characters in the transmit FIFO, which is 2^TXFIFO.

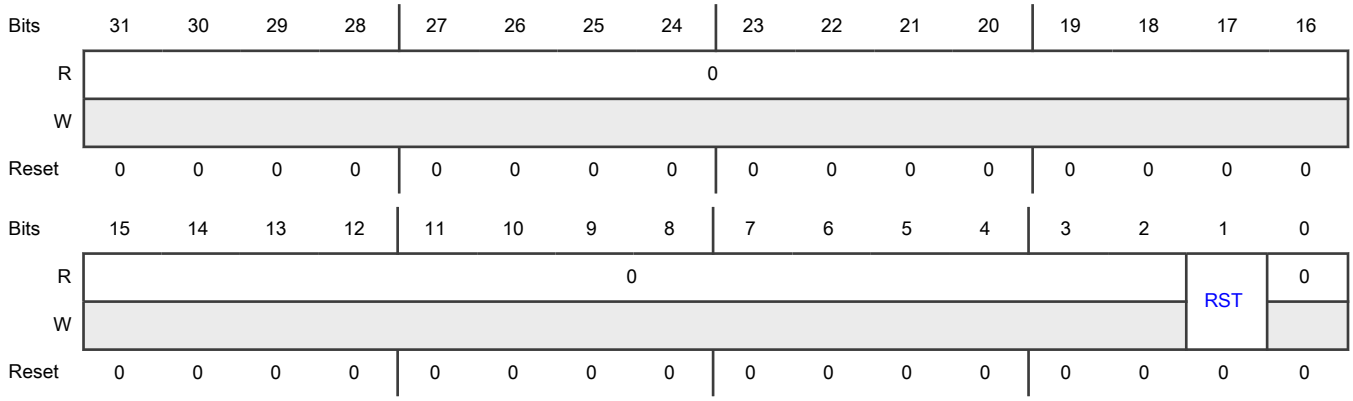
68.5.5.4 Global (GLOBAL)

Offset

Register	Offset
GLOBAL	8h

Function
Performs global functions.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RST	Software Reset Specifies whether the module is reset. This field resets all internal logic and registers, except Global (GLOBAL). The reset takes effect immediately and remains asserted until you negate it. There is no minimum delay required before clearing the software reset. 0b - Not reset 1b - Reset
0 —	Reserved

68.5.5.5 Pin Configuration (PINCFG)

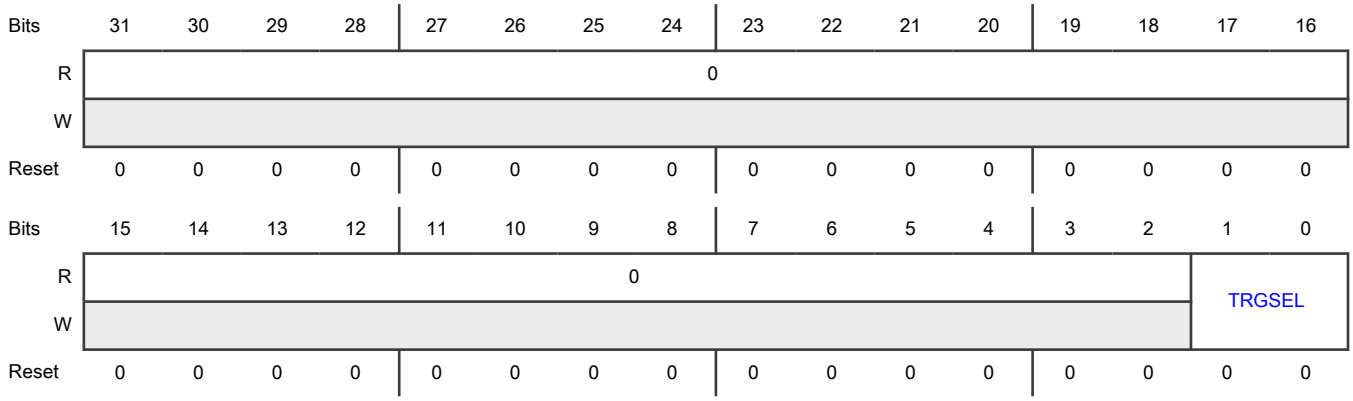
Offset

Register	Offset
PINCFG	Ch

Function

Enables the selection of input pins.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 TRGSEL	Trigger Select Configures the input trigger usage. You must change the value of this field only when both the transmitter and receiver are disabled. 00b - Input trigger disabled 01b - Input trigger used instead of the RXD pin input 10b - Input trigger used instead of the CTS_B pin input 11b - Input trigger used to modulate the TXD pin output, which (after TXINV configuration) is internally ANDed with the input trigger

68.5.5.6 Baud Rate (BAUD)

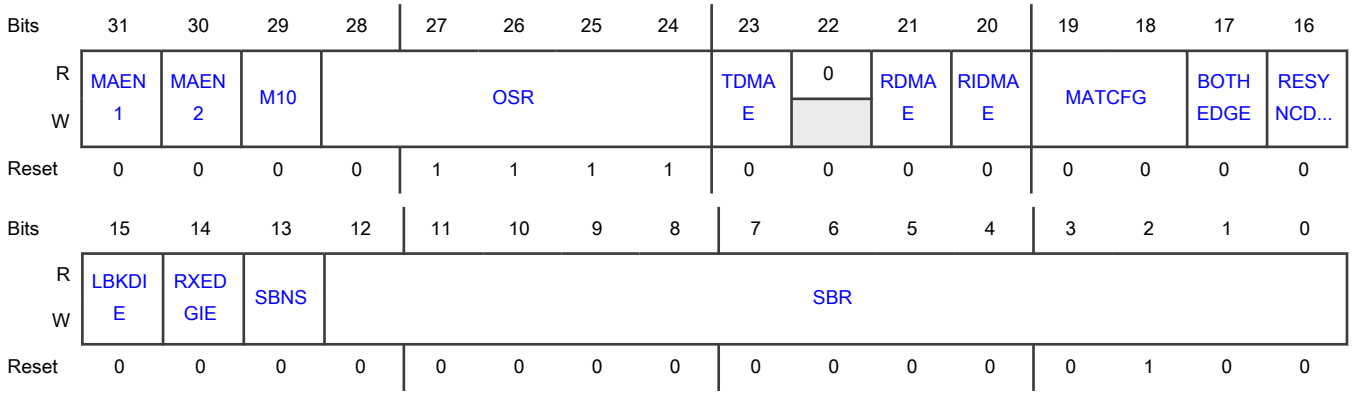
Offset

Register	Offset
BAUD	10h

Function

Configures the baud rate.

Diagram



Fields

Field	Function
31 MAEN1	Match Address Mode Enable 1 Enables automatic address matching or data matching mode for MATCH[MA1]. If this field is 0, normal operation takes place. 0b - Disable 1b - Enable
30 MAEN2	Match Address Mode Enable 2 Enables automatic address matching or data matching mode for MATCH[MA2]. If this field is 0, normal operation takes place. 0b - Disable 1b - Enable
29 M10	10-Bit Mode Select Causes the tenth bit to be a part of the serial transmission. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - Receiver and transmitter use 7-bit to 9-bit data characters 1b - Receiver and transmitter use 10-bit data characters
28-24 OSR	Oversampling Ratio Configures the OSR of the receiver. You must change the value of this field only when both the transmitter and receiver are disabled. **NOTE** BAUD[OSR] results in an OSR of BAUD[OSR] + 1, for example, BAUD[OSR] = 0_0101b results in a final division by 6. 0_0000b - Results in an OSR of 16 0_0001b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0_0010b - Reserved 0_0011b - Results in an OSR of 4 (requires BAUD[BOTHEDGE] to be 1) 0_0100b - Results in an OSR of 5 (requires BAUD[BOTHEDGE] to be 1) 0_0101b - Results in an OSR of 6 (requires BAUD[BOTHEDGE] to be 1) 0_0110b - Results in an OSR of 7 (requires BAUD[BOTHEDGE] to be 1) 0_0111b - Results in an OSR of 8 0_1000b - Results in an OSR of 9 0_1001b - Results in an OSR of 10 0_1010b - Results in an OSR of 11 0_1011b - Results in an OSR of 12 0_1100b - Results in an OSR of 13 0_1101b - Results in an OSR of 14 0_1110b - Results in an OSR of 15 0_1111b - Results in an OSR of 16 1_0000b - Results in an OSR of 17 1_0001b - Results in an OSR of 18 1_0010b - Results in an OSR of 19 1_0011b - Results in an OSR of 20 1_0100b - Results in an OSR of 21 1_0101b - Results in an OSR of 22 1_0110b - Results in an OSR of 23 1_0111b - Results in an OSR of 24 1_1000b - Results in an OSR of 25 1_1001b - Results in an OSR of 26 1_1010b - Results in an OSR of 27 1_1011b - Results in an OSR of 28 1_1100b - Results in an OSR of 29 1_1101b - Results in an OSR of 30 1_1110b - Results in an OSR of 31 1_1111b - Results in an OSR of 32
23 TDMAE	Transmitter DMA Enable Enables STAT[TDRE] to generate a DMA request. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
22 —	Reserved
21 RDMAE	Receiver Full DMA Enable Enables STAT[RDRF] to generate a DMA request. 0b - Disable 1b - Enable
20 RIDMAE	Receiver Idle DMA Enable Enables STAT[IDLE] to generate a DMA request. If this field is 1, reading Data (DATA) when either DATA[RXEMPT] or DATA[IDLINE] is 1 generates an end-of-packet response until the completion of the DMA minor loop. During an end-of-packet response, reading Data (DATA) returns 0000_33FFh and does not pull data from the receive FIFO. STAT[IDLE] becomes 0 on completion of the minor loop, provided an end-of-packet response is generated and either the receive FIFO is empty or the receiver is active. 0b - Disable 1b - Enable
19-18 MATCFG	Match Configuration Configures the match addressing mode used. You must change the value of this field only when both the transmitter and receiver are disabled. 00b - Address match wake-up 01b - Idle match wake-up 10b - Match on and match off 11b - Enables RWU on data match and match on or off for the transmitter CTS input
17 BOTHEDGE	Both Edge Sampling Enables sampling of the received data on both edges of the baud rate clock, effectively doubling the number of times the receiver samples the input data for a given OSR. This field must be 1 for OSRs between x4 and x7 and is optional for higher OSRs. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver samples input data using the rising edge of the baud rate clock. If this field is 1, the receiver samples input data using the rising and falling edges of the baud rate clock. 0b - Rising edge 1b - Both rising and falling edges
16 RESYNCDIS	Resynchronization Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Disables resynchronization of the received data word when a data one followed by data zero transition is detected. You must change the value of this field only when the receiver is disabled. 0b - Enable 1b - Disable
15 LBKDIE	LIN Break Detect Interrupt Enable Enables STAT[LBKDIF] to generate hardware interrupt requests. If this field is 0, hardware interrupts from STAT[LBKDIF] (uses polling) are disabled. If this field is 1, hardware interrupts are requested when STAT[LBKDIF] is 1. 0b - Disable 1b - Enable
14 RXEDGIE	RX Input Active Edge Interrupt Enable Enables STAT[RXEDGIF] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[RXEDGIF] are disabled. If this field is 1, hardware interrupts are requested when STAT[RXEDGIF] is 1. Changing the value of CTRL[LOOPS] or CTRL[RSRC] when this field (RXEDGIE) is 1 can cause STAT[RXEDGIF] to become 1. 0b - Disable 1b - Enable
13 SBNS	Stop Bit Number Select Determines whether data characters include one or two stop bits. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - One stop bit 1b - Two stop bits
12-0 SBR	Baud Rate Modulo Divisor Sets the modulo divide rate for the baud rate generator. • If SBR is 0, baud rate generator is disabled. • If SBR is 1–8191, baud rate = baud clock ÷ ((OSR + 1) × SBR). You must update the 13-bit baud rate setting [SBR12:SBR0] only when both the transmitter and receiver are disabled (both CTRL[RE] and CTRL[TE] are 0).

68.5.5.7 Status (STAT)

Offset

Register	Offset
STAT	14h

Function

Provides the module status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	W1C	W1C										W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0				0		0						AME	LBKFE
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LBKDIF	LIN Break Detect Interrupt Flag Indicates whether a LIN break character is detected. This field becomes 1 when the LIN break detect circuitry is enabled and a LIN break character is detected. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
30 RXEDGIF	RXD Pin Active Edge Interrupt Flag Indicates whether an active edge on the receive pin has occurred.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field becomes 1 whenever the receiver is enabled and an active edge (falling if STAT[RXINV] is 0; rising if STAT[RXINV] is 1) on the RXD pin occurs. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not occurred 1b - Occurred When writing 0b - No effect 1b - Clear the flag
29 MSBF	MSB First Specifies the first bit that is transmitted after the start bit. If this field is 0, LSB (bit 0) is the first bit transmitted after the start bit (which means, the first bit received after the start bit is identified as bit 0). If this field is 1, MSB (identified as bit 9, bit 8, bit 7, or bit 6) is the first bit that is transmitted, after the start bit, depending on the settings of CTRL[M], CTRL[PE], and BAUD[M10]. Writing 1 to this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - LSB 1b - MSB
28 RXINV	Receive Data Inversion Specifies whether receive data is inverted. Writing 1 to this field reverses the polarity of the received data input. You must change the value of this field only when the receiver is disabled. **NOTE** Writing 1 to this field inverts the RXD input for all cases: data bits, start and stop bits, break, and idle. 0b - Inverted 1b - Not inverted
27 RWUID	Receive Wake Up Idle Detect Controls, for CTRL[RWU] on idle character detection, whether the idle character that wakes up the receiver writes 1 to STAT[IDLE]. For address match wake-up, this field controls whether STAT[IDLE] = 1 when the address does not match. You must change the value of this field only when the receiver is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If this field is 0, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] does not become 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] does not become 1 when an address does not match. If this field is 1, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] becomes 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] becomes 1 when an address does not match. 0b - STAT[IDLE] does not become 1 1b - STAT[IDLE] becomes 1
26 BRK13	Break Character Generation Length Selects the longer transmitted break character length. The state of this field does not affect the detection of a framing error. You must change the value of this field only when the transmitter is disabled. You can send a break character by writing 1 to CTRL[SBK], or by writing the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0. 0b - 9 to 13 bit times 1b - 12 to 15 bit times
25 LBKDE	LIN Break Detection Enable Enables LIN break detection. If this field is 0, LIN break detect is disabled, and only a normal break character can be detected. If this field is 1, LIN break detect is enabled and the LIN break character is detected at a length of 11 bit times (if CTRL[M] is 0), 12 bit times (if CTRL[M] is 1), or 13 bit times (if BAUD[M10] is 1). This field selects a longer break character detection length. When the field is 1, receive data is not stored in the receive FIFO. **NOTE** This field enables the LIN break detect circuit and disables writing receive data to FIFO. Therefore, it ignores all characters except a LIN break. 0b - Disable 1b - Enable
24 RAF	Receiver Active Flag Indicates whether the LPUART receiver is idle or active. This field becomes 1 when the receiver detects the beginning of a valid start bit, and the field becomes 0 automatically when the receiver detects an idle line. 0b - Idle, waiting for a start bit 1b - Receiver active (RXD pin input not idle)
23 TDRE	Transmit Data Register Empty Flag Indicates whether the transmit FIFO level is greater than, equal to, or less than the watermark.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	After the transmit FIFO is enabled, this field becomes 1 when the number of datawords in the transmit FIFO is equal to, or less than the number that WATER[TXWATER] indicates. To make the value of this field 0, write to it until the number of words in the transmit FIFO is greater than the number that WATER[TXWATER] indicates. After the transmit FIFO is disabled, this field becomes 1 to indicate that the FIFO level is less than the watermark. To make the value of this field 0 again, write to Data (DATA). This register is not affected by a character that is in the process of being transmitted; it is updated at the start of each transmitted character. 0b - Greater than watermark 1b - Equal to or less than watermark
22 TC	Transmission Complete Flag Indicates whether the transmitter is active. This field becomes 0 when a transmission is in progress or a preamble or break character is loaded; in other words, when the transmitter is active (sending data, a preamble, or a break). The field becomes 1 when the transmit buffer is empty and no data, preamble, or break character is being transmitted; in other words, when the transmission activity is complete. When this happens, the transmit data output signal becomes idle (logic 1). This field becomes 0 after you write to Data (DATA) to transmit new data, queuing a preamble by first writing 0 and then writing 1 to CTRL[TE], queuing a break character by writing 1 to CTRL[SBK]. 0b - Transmitter active 1b - Transmitter idle
21 RDRF	Receive Data Register Full Flag Indicates whether the receive FIFO level is less than, equal to, or greater than the watermark. This field becomes 1 when the number of datawords in the receive buffer is greater than the number that WATER[RXWATER] indicates and the receive FIFO is enabled. To write 0 to this field, read Data (DATA) until the number of datawords in the receive FIFO is equal to, or less than the number that WATER[RXWATER] indicates. When the receive FIFO is disabled, this field (RDRF) becomes 1 if the receive buffer (Data (DATA)) is full. To make this field 0, read Data (DATA). A character that is in the process of being received does not cause a change in this field until the entire character is received. Even if this field is 1, the character continues to be received until an overrun condition occurs after the entire character is received. 0b - Equal to or less than watermark 1b - Greater than watermark
20 IDLE	Idle Line Flag Indicates whether an idle line is detected. This field becomes 1 when the LPUART receive line becomes idle for a full character time after a period of activity. When CTRL[ILT] is 0, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bit time count towards the full character time of logic high, 10 to 13 bit times, needed for the receiver to detect an idle line. After CTRL[ILT] becomes 1, the receiver does not start counting idle bit times until after the stop bits. The stop bits and any logic high bit times at the end

Table continues on the next page...

Table continued from the previous page...

Field	Function
	of the previous character do not count towards the full character time of logic high needed for the receiver to detect an idle line. For this field to become 0, write 1 to it. After the field becomes 0, you cannot write 1 to it again until after a new character is stored in the receive buffer or a LIN break character writes 1 to STAT[LBKDIF]. This field becomes 1 only once, even if the receive line remains idle for an extended period. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Idle line detected 1b - Idle line not detected When writing 0b - No effect 1b - Clear the flag
19 OR	Receiver Overrun Flag Indicates whether there is receive overrun. This field becomes 1 when you cannot prevent STAT[RDRF] from overflowing with data. The field becomes 1 immediately after the stop bit is completely received for the dataword that overflows the buffer and all the other error fields (STAT[FE], STAT[NF], and STAT[PF]) are prevented from becoming 1. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If STAT[LBKDE] is enabled and a LIN break is detected, this field becomes 1 if STAT[LBKDIF] is not 0 before the next data character is received. When this field is 1, no additional data is stored in the receive FIFO even if sufficient room exists. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overrun 1b - Receive overrun (new LPUART data is lost) When writing 0b - No effect 1b - Clear the flag
18 NF	Noise Flag Indicates whether noise is detected in the received character of Data (DATA). The advanced sampling technique used in the receiver takes three samples in each of the received bits. If some of these samples disagree with the rest of the samples within any bit time in the frame, then noise is detected for that character. This field becomes 1 whenever the next character to be read from Data (DATA) is received with noise detected within the character.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No noise detected 1b - Noise detected When writing 0b - No effect 1b - Clear the flag
17 FE	Framing Error Flag Indicates whether a framing error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received with logic 0 detected where a stop bit was expected. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No framing error detected (this does not guarantee that the framing is correct) 1b - Framing error detected When writing 0b - No effect 1b - Clear the flag
16 PF	Parity Error Flag Indicates whether a parity error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received when parity is enabled (CTRL[PE] is 1) and the parity bit in the received character does not agree with the expected parity value. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No parity error detected 1b - Parity error detected When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
15 MA1F	Match 1 Flag Indicates whether the received data is equal to MATCH[MA1]. This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA1]. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not equal to MA1 1b - Equal to MA1 When writing 0b - No effect 1b - Clear the flag
14 MA2F	Match 2 Flag Indicates whether the received data is equal to MATCH[MA2]. This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA2]. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not equal to MA2 1b - Equal to MA2 When writing 0b - No effect 1b - Clear the flag
13-10 —	Reserved
9-8 —	Reserved
7-2 —	Reserved
1 AME	Address Mark Enable Configures the location of the address mark when configured for MSB first transfers.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This field has no effect when configured for LSB first and you must change the value of this field only when both the transmitter and receiver are disabled. If this field is 0, address mark in character is MSB. If this field is 1, the address mark is stored in Data (DATA) at MSB (or MSB-1 when the parity bit is enabled). In other words, the address mark in character is the last bit before the stop bit (or parity bit when enabled). 0b - Disable 1b - Enable
0 LBKFE	LIN Break Flag Enable Enables the LIN break flag to assert whenever a LIN break character is detected. Unlike STAT[LBKDE], this does not impact data being stored in the receive data buffer, but does cause STAT[LBKDIF] to become 1 whenever a LIN break is detected. Because a LIN break is longer than a normal character, the LIN break triggers a write to STAT[RDRF] with the data fields as 0 and STAT[FE] as 1. The character following the LIN break has DATA[LINBRK] as 1 to indicate that the previous character was a LIN break. You must change the value of this field only when both the transmitter and receiver are disabled. If this field is 1, the LIN break character is detected at a length of 11-bit times (if CTRL[M] is 0), 12 (if CTRL[M] is 1), or 13 (if BAUD[M10] is 1). 0b - Disable 1b - Enable

68.5.5.8 Control (CTRL)

Offset

Register	Offset
CTRL	18h

Function

Controls various optional features of the LPUART system.

You must write to the fields of this register only when both the transmitter and receiver are disabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1IE	MA2IE	0	0	M7	IDLECFG			LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 R8T9	Receive Bit 8 Transmit Bit 9 Contains R8 and T9 that correspond to different functions. R8 is the ninth data bit received after you configure LPUART for 9-bit or 10-bit data formats. When reading 9-bit or 10-bit data, read R8 before reading Data (DATA). T9 is the tenth data bit transmitted after you configure LPUART for 10-bit data formats. When writing 10-bit data, write T9 before writing to Data (DATA). If T9 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA). **NOTE** R8 is a read-only bit and T9 is a write-only bit; the value read is different from the value written.
30 R9T8	Receive Bit 9 Transmit Bit 8 Contains R9 and T8 that correspond to different functions. R9 is the tenth data bit received after you configure LPUART for 10-bit data formats. When reading 10-bit data, read R9 before reading Data (DATA). T8 is the ninth data bit transmitted after you configure LPUART for 9-bit or 10-bit data formats. When writing 9-bit or 10-bit data, write T8 before writing to Data (DATA). If T8 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA). **NOTE** R9 is a read-only field and T8 is a write-only field; the value read is different from the value written.
29 TXDIR	TXD Pin Direction in Single-Wire Mode Determines the direction of data at the TXD pin, in Single-Wire mode, when LPUART is configured for a single-wire half-duplex operation (CTRL[LOOPS] and CTRL[RSRC] are 1). When writing 0 to this field, the transmitter finishes transmitting the current character (if any) before the receiver starts receiving data from the TXD pin.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Input 1b - Output
28 TXINV	Transmit Data Inversion Specifies whether transmit data is inverted. Writing 1 to this field reverses the polarity of the transmitted data output. This action inverts the TXD output for all cases: data bits, start and stop bits, break, and idle. 0b - Not inverted 1b - Inverted
27 ORIE	Overrun Interrupt Enable Enables STAT[OR] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when OR interrupts are disabled. 0b - Disable 1b - Enable
26 NEIE	Noise Error Interrupt Enable Enables STAT[NF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when NF interrupts are disabled. 0b - Disable 1b - Enable
25 FEIE	Framing Error Interrupt Enable Enables STAT[FE] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when FE interrupts are disabled. 0b - Disable 1b - Enable
24 PEIE	Parity Error Interrupt Enable Enables STAT[PF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when PF interrupts are disabled. 0b - Disable 1b - Enable
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests if STAT[TDRE] is 1. 0b - Disable 1b - Enable
22	Transmission Complete Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
TCIE	Enables STAT[TC] to generate interrupt requests if STAT[TC] is 1. 0b - Disable 1b - Enable
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate hardware interrupt requests if STAT[RDRF] is 1. 0b - Disable 1b - Enable
20 ILIE	Idle Line Interrupt Enable Enables hardware interrupts. This field enables STAT[IDLE] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[IDLE] are disabled and polling is used, and if this field is 1, hardware interrupts are enabled when STAT[IDLE] is 1. 0b - Disable 1b - Enable
19 TE	Transmitter Enable Enables the LPUART transmitter. Using this field, you can also queue an idle preamble by first writing 0 and then writing 1 to this field. After this field becomes 0, the field reads 1 until the transmitter has completed the current character and the TXD pin is tristated. You can also queue a single idle character by writing to the transmit FIFO with DATA[FRETSC] and DATA[R9T9] = 1. 0b - Disable 1b - Enable
18 RE	Receiver Enable Enables the LPUART receiver. After you write 0 to this field, this field remains 1 until the receiver finishes receiving the current character (if any). 0b - Disable 1b - Enable
17 RWU	Receiver Wake-Up Control Specifies whether the LPUART receiver in standby is waiting for a wake-up condition. You can write 1 to this field to place the LPUART receiver in a Standby state. The field becomes 0 automatically when an RWU event occurs, that is, in case of an idle event when CTRL[WAKE] is 0 or an address match when CTRL[WAKE] is 1 and STAT[RWUID] is 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE You must write 1 to this field only when CTRL[WAKE] is 0 (wake-up on idle), if the channel is currently not idle. You can determine this by the value of STAT[RAF]. If the field is 1 to wake up an idle event and the channel is already idle, LPUART, possibly, discards the data. This is because the data must be received or a LIN break is detected after an Idle condition is detected before the IDLE flag is allowed to be reasserted. 0b - Normal receiver operation 1b - LPUART receiver in standby, waiting for a wake-up condition
16 SBK	Send Break Specifies whether queue break character(s) are to be sent. Writing 1 and then 0 to this field queues a break character in the transmit data stream. Additional break characters of 9 to 13 bits, or 12 to 15 bits if STAT[BRK13] is 1, and bit times of logic 0 are queued as long as this field is 1. Depending on the timing when this field is 1 and 0, relative to the character currently being transmitted, a second break character may be queued before you write 0 to this field. If the time taken to write 0 to this field is too long, for example, if the field does not become 0 by the end of the first break character, a second break character is sent. This is compared to queuing a break character through the transmit FIFO that guarantees only one break character is sent. You can also queue a single break character by writing to the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0. 0b - Normal transmitter operation 1b - Queue break character(s) to be sent
15 MA1IE	Match 1 (MA1F) Interrupt Enable Enables the MA1F interrupt. 0b - Disable 1b - Enable
14 MA2IE	Match 2 (MA2F) Interrupt Enable Enables the MA2F interrupt. 0b - Disable 1b - Enable
13 —	Reserved
12 —	Reserved
11 M7	7-Bit Mode Select Specifies the data characters that the receiver and transmitter use.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	You must change the value of this field only after both the transmitter and receiver are disabled. 0b - 8-bit to 10-bit 1b - 7-bit
10-8 IDLECFG	Idle Configuration Configures the number of idle characters that must be received before you write 1 to STAT[IDLE]. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 128
7 LOOPS	Loop Mode Select Selects Loop mode. After this field becomes 1, the RXD pin is disconnected from LPUART and the transmitter output is internally connected to the receiver input. The transmitter and receiver must be enabled to use the loop function. In Loop mode or Single-Wire mode, the transmitter outputs are internally connected to the receiver input (see CTRL[RSRC]). 0b - Normal operation: RXD and TXD use separate pins 1b - Loop mode or Single-Wire mode
6 DOZEEN	Doze Mode Enables LPUART in Doze mode. If this field is 1, LPUART remains active when not in Doze mode. 0b - Enable 1b - Disable
5 RSRC	Receiver Source Select Determines the source of the receiver shift register input if CTRL[LOOPS] is 1. This field has no effect unless CTRL[LOOPS] is 1. If this field is 0, internal Loopback mode is selected. LPUART does not use the RXD pin. Additionally, the CTS_B pin is not used and internally driven by the RTS_B output. If this field is 1, single-wire LPUART mode is selected where the TXD pin is connected to the transmitter output and receiver input.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Internal Loopback mode 1b - Single-wire mode
4 M	9-Bit Or 8-Bit Mode Select Specifies the data characters that the receiver and transmitter use. 0b - 8-bit 1b - 9-bit
3 WAKE	Receiver Wake-Up Method Select Determines which condition wakes up LPUART when CTRL[RWU] = 1 and BAUD[MATCFG] = 0 (this field must be 1 when BAUD[MATCFG] = 11): • Address mark in the bit preceding the stop bit (or bit preceding the parity bit when parity is enabled) of the received data character • An idle condition on the receive pin input signal If this field is 0, CTRL[RWU] is configured for idle line wake-up, and if this field is 1, CTRL[RWU] is configured with address mark wake-up. 0b - Idle 1b - Mark
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or the stop bit. If the count begins after the start bit, a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE** In case you write 1 to this field, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0b - After the start bit 1b - After the stop bit
1 PE	Parity Enable Enables hardware parity generation and checking. If parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0b - Disable 1b - Enable
0 PT	Parity Type Selects the type of parity, even or odd, if parity is enabled (CTRL[PE] = 1):

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• Odd parity means that the total number of logic 1 bits in the data character, including the parity bit, is odd. • Even parity means that the total number of 1s in the data character, including the parity bit, is even. 0b - Even parity 1b - Odd parity

68.5.5.9 Data (DATA)

Offset

Register	Offset
DATA	1Ch

Function

Supports 8-bit, 16-bit, or 32-bit writes, each type of write performing a separate function. An 8-bit write to DATA[7:0] pushes {CTRL[R8T9], CTRL[R9T8], DATA[7:0]} the transmit FIFO with TSC clear. A 16-bit or 32-bit write pushes the data written into the FIFO and does not update the value of CTRL[R8T9] or CTRL[R9T8].

Reads and writes of this register are also involved in the automatic flag clearing mechanisms for some of the LPUART status fields.

NOTE

Reads return the contents of the read-only receive FIFO and writes go to the write-only transmit FIFO, making this register work as a set of two separate registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOISY	PARITYE	FRETSC	RXEMPT	IDLINE	LINBRK	R9T9	R8T8	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15 NOISY	Noisy Data Received Indicates whether the current received dataword contained in DATA[R9:R0] is received with noise. 0b - Received without noise 1b - Received with noise
14 PARITYE	Parity Error Indicates whether the current received dataword contained in DATA[R9:R0] is received with a parity error. 0b - Received without a parity error 1b - Received with a parity error
13 FRETSC	Frame Error Transmit Special Character Specifies the way the dataword is received. For reads, this field indicates that the current received dataword contained in DATA[R9:R0] is received with a frame error. For writes, the field indicates that a break or idle character is to be transmitted instead of the contents in DATA[T9:T0]. T9 indicates a break character when it is 0 and indicates an idle character when it is 1. The contents of DATA[T8:T0] must be 0. 0b - Received without a frame error on reads or transmits a normal character on writes 1b - Received with a frame error on reads or transmits an idle or break character on writes
12 RXEMPT	Receive Buffer Empty Indicates whether the receive buffer contains valid data. This field becomes 1 when there is no data in the receive buffer. The field does not consider data in the receive shift register. 0b - Valid data 1b - Invalid data and empty
11 IDLINE	Idle Line Indicates whether the receiver line was idle before receiving the character in DATA[9:0]. It can be read as “1” on the first character when the receiver is first enabled. The difference between this field and STAT[IDLE] is that, STAT[IDLE] flag does not set on an idle line after the receiver is first enabled, it needs to receive a character before it can become set, whereas this field does not have this limitation and can be set on the first character received if an idle line is detected beforehand. 0b - Not idle 1b - Idle
10	LIN Break

Table continues on the next page...

Table continued from the previous page...

Field	Function
LINBRK	Indicates whether the receiver line detected a LIN break before receiving the character in DATA[9:0]. This field requires the value of STAT[LBKDIF] to be 1. If this field is 0, the LIN break detect circuitry is disabled. 0b - Not detected 1b - Detected
9 R9T9	Read receive FIFO bit 9 or write transmit FIFO bit 9
8 R8T8	Read receive FIFO bit 8 or write transmit FIFO bit 8
7 R7T7	Read receive FIFO bit 7 or write transmit FIFO bit 7
6 R6T6	Read receive FIFO bit 6 or write transmit FIFO bit 6
5 R5T5	Read receive FIFO bit 5 or write transmit FIFO bit 5
4 R4T4	Read receive FIFO bit 4 or write transmit FIFO bit 4
3 R3T3	Read receive FIFO bit 3 or write transmit FIFO bit 3
2 R2T2	Read receive FIFO bit 2 or write transmit FIFO bit 2
1 R1T1	Read receive FIFO bit 1 or write transmit FIFO bit 1
0 R0T0	Read receive FIFO bit 0 or write transmit FIFO bit 0

68.5.5.10 Match Address (MATCH)

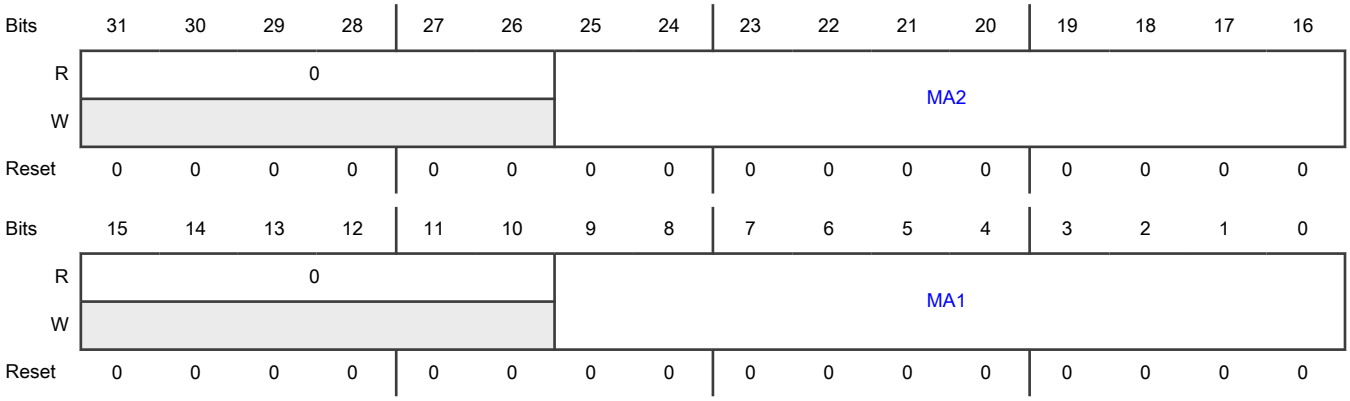
Offset

Register	Offset
MATCH	20h

Function

Provides addresses for address matching during the receiver operation.

Diagram



Fields

Field	Function
31-26 —	Reserved
25-16 MA2	Match Address 2 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] only when the associated Baud Rate (BAUD) field is 0.
15-10 —	Reserved
9-0 MA1	Match Address 1 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] fields only when the associated field in Baud Rate (BAUD) is 0.

68.5.5.11 MODEM IrDA (MODIR)

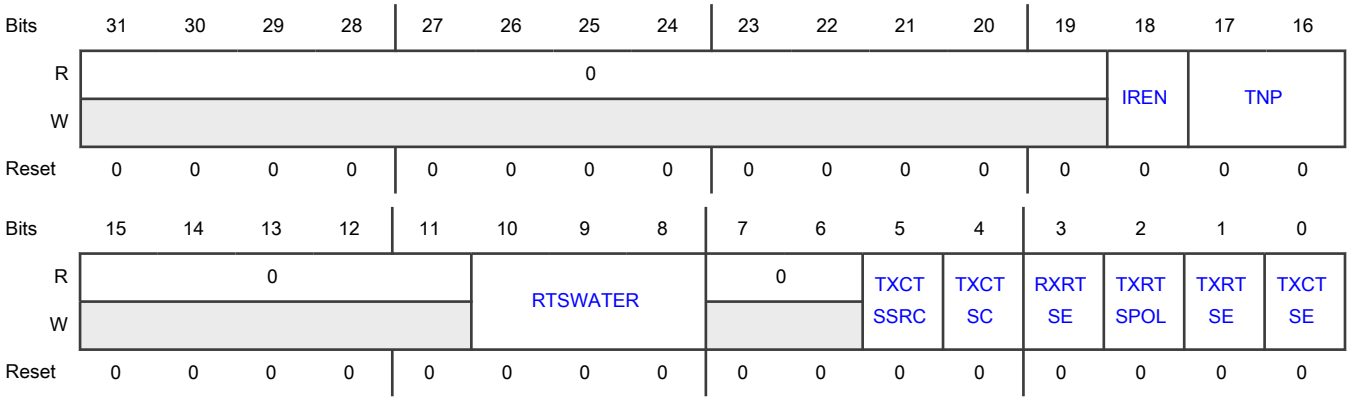
Offset

Register	Offset
MODIR	24h

Function

Controls options for setting the MODEM configuration.

Diagram



Fields

Field	Function
31-19 —	Reserved
18 IREN	IR Enable Enables IR modulation and demodulation. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - Disable 1b - Enable
17-16 TNP	Transmitter Narrow Pulse Specifies whether LPUART transmits a 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulse when the IR pulse is enabled. You must change the value of this field only when both the transmitter and receiver are disabled. The IR pulse width must be configured to less than half of the OSR. Common pulse widths are 3 ÷ 16, 1 ÷ 16, 1 ÷ 32, or 1 ÷ 4 of the bit length. You can configure these by selecting the appropriate OSR and pulse width. 00b - 1 ÷ OSR 01b - 2 ÷ OSR 10b - 3 ÷ OSR 11b - 4 ÷ OSR
15-11 —	Reserved
10-8	Receive RTS Configuration

Table continues on the next page...

Table continued from the previous page...

Field	Function
RTSWATER	Configures the assertion and negation of the receiver's RTS_B output. The receiver's RTS_B output negates when the number of empty words in the receive FIFO is greater or equal to the value of this field. If this field is 0, the RTS_B pin negates when the receive FIFO is full. For the purpose of receive RTS_B generation, the number of words in the receive FIFO updates when a start bit is detected. This supports additional latency between RTS_B negation and the external transmitter ceasing transmission. If both receive RTS_B and address or data matching is enabled, RTS_B could assert at the end of a character if there exists no match. You must change the value of this field only when the receiver is disabled.
7-6 —	Reserved
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0b - The CTS_B pin 1b - An internal connection to the receiver address match result
4 TXCTSC	Transmit CTS Configuration Configures whether the CTS state or input is checked or sampled at the start of each character or only when the transmitter is idle. 0b - Sampled at the start of each character 1b - Sampled when the transmitter is idle
3 RXRTSE	Receiver RTS Enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver has no effect on RTS. If this field is 1, RTS is deasserted if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1. RTS is asserted if STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. **NOTE** Do not write 1 to both MODIR[RXRTSE] and MODIR[TXRTSE]. 0b - Disable 1b - Enable
2 TXRTSPOL	Transmitter RTS Polarity Controls the polarity of the transmitter RTS. This field does not affect the polarity of the receiver RTS that remains negated in the active-low state unless MODIR[TXRTSE] is 1. You must change the value of this field only when the transmitter is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Active low 1b - Active high
1 TXRTSE	Transmitter RTS Enable Controls the operation of RTS before and after a transmission. You must change the value of this field only when the transmitter is disabled. If this field is 0, the transmitter has no effect on RTS, and if this field is 1, a character is placed into an empty transmit shift register. RTS asserts 1-bit time before the start bit is transmitted and deasserts 1-bit time after all characters in the transmitter FIFO and shift register are completely sent, including the last stop bit. 0b - Disable 1b - Enable
0 TXCTSE	Transmitter CTS Enable Enables the operation of the transmitter. You can write 1 to this field irrespective of the states of MODIR[TXRTSE] and MODIR[RXRTSE]. If this field is 1, the transmitter checks the state of the CTS signal each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the TXD signal remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS, when a character is being sent, do not affect its transmission. 0b - Disable 1b - Enable

68.5.5.12 FIFO (FIFO)

Offset

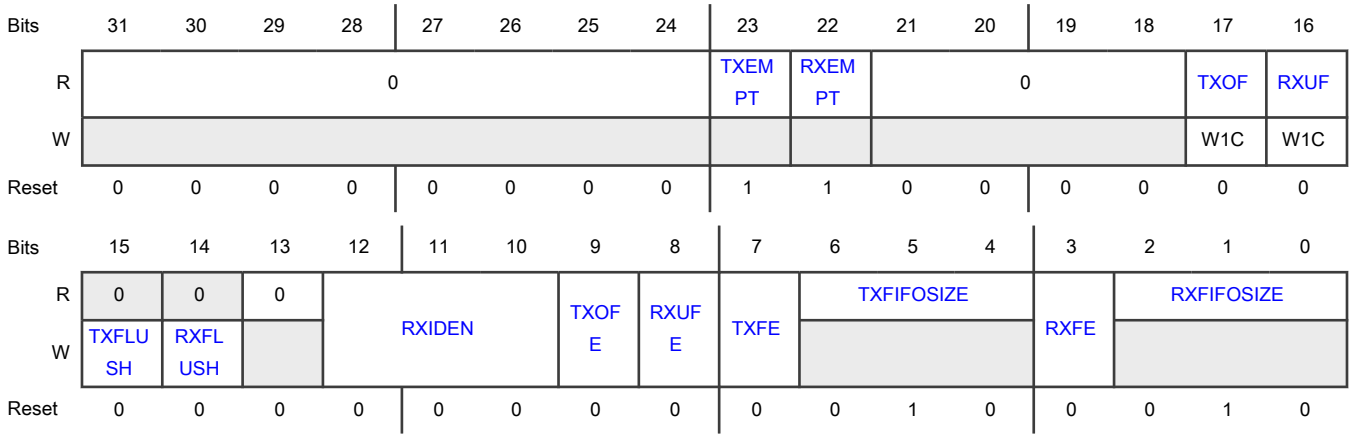
Register	Offset
FIFO	28h

Function

Provides you the ability to turn on and turn off the FIFO functionality.

This register also provides you the size of the FIFO that has been implemented. You can read this register at any time and must write to it only when CTRL[RE] and CTRL[TE] are 0 and the FIFO is empty.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 TXEMPT	Transmit FIFO Or Buffer Empty Indicates whether the transmit buffer is empty. This field becomes 1 when there is no data in the transmit FIFO or buffer. The field does not consider data in the transmit shift register. 0b - Not empty 1b - Empty
22 RXEMPT	Receive FIFO Or Buffer Empty Indicates whether the receive buffer is empty. This field becomes 1 when there is no data in the receive FIFO or buffer. The field does not consider data in the receive shift register. 0b - Not empty 1b - Empty
21-18 —	Reserved
17 TXOF	Transmitter FIFO Overflow Flag Indicates whether more data has been written to the transmit FIFO than it can hold. If this field is 0, no transmit FIFO overflow has occurred since the last time the field was cleared, and if this field is 1, at least one transmit FIFO overflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[TXOFE]. However, an interrupt is issued to the host only if FIFO[TXOFE] is 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No effect 1b - Clear the flag
16 RXUF	Receiver FIFO Underflow Flag Indicates whether more data has been read from the receive FIFO than was present. If this field is 0, no receive FIFO underflow has occurred since the last time the field was cleared, and if this field is 1, at least one receive FIFO underflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[RXUFE]. However, an interrupt is issued to the host only if FIFO[RXUFE] is 1. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underflow 1b - Underflow When writing 0b - No effect 1b - Clear the flag
15 TXFLUSH	Transmit FIFO Flush Causes all data that is stored in the transmit FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the transmit FIFO or buffer clears out. This does not affect data in the transmit shift register. 0b - No effect 1b - All data flushed out
14 RXFLUSH	Receive FIFO Flush Causes all data that is stored in the receive FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the receive FIFO or buffer clears out.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	This does not affect data in the receive shift register. 0b - No effect 1b - All data flushed out
13 —	Reserved
12-10 RXIDEN	Receiver Idle Empty Enable Enables STAT[RDRF] to become 1 when the receiver is idle for a number of idle characters and the FIFO is not empty. 000b - Disable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle 001b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for one character 010b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for two characters 011b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for four characters 100b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for eight characters 101b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 16 characters 110b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 32 characters 111b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 64 characters
9 TXOFE	Transmit FIFO Overflow Interrupt Enable Enables FIFO[TXOF] to generate an interrupt to the host. 0b - Disable 1b - Enable
8 RXUFE	Receive FIFO Underflow Interrupt Enable Enables FIFO[RXUF] to generate an interrupt to the host. 0b - Disable 1b - Enable
7 TXFE	Transmit FIFO Enable Enables the transmit FIFO. If this field is 0, the transmit buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[TXFIFOSIZE]. Both CTRL[TE] and CTRL[RE] must be 0 before you change the value of this field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If this field is 1, the built-in FIFO structure for the transmit buffer is enabled. FIFO[TXFIFOSIZE] indicates the size of the FIFO structure. 0b - Disable 1b - Enable
6-4 TXFIFOSIZE	Transmit FIFO Buffer Depth Indicates the maximum number of transmit datawords (transmit FIFO buffer depth) that can be stored in the transmit buffer. 000b - 1 001b - 4 010b - 8 011b - 16 100b - 32 101b - 64 110b - 128 111b - 256
3 RXFE	Receive FIFO Enable Enables the receive FIFO. If this field is 0, the receive buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[RXFIFOSIZE]. Both CTRL[RE] and CTRL[TE] must be 0 before you change the value of this field. If this field is 1, the built-in FIFO structure for the receive buffer is enabled. FIFO[RXFIFOSIZE] indicates the size of the FIFO structure. 0b - Disable 1b - Enable
2-0 RXFIFOSIZE	Receive FIFO Buffer Depth Indicates the maximum number of receive datawords (receive FIFO buffer depth) that can be stored in the receive buffer before an overrun occurs. 000b - 1 001b - 4 010b - 8 011b - 16 100b - 32 101b - 64 110b - 128 111b - 256

68.5.5.13 Watermark (WATER)

Offset

Register	Offset
WATER	2Ch

Function

Provides the ability to set a programmable threshold for notification, or sets the programmable thresholds to indicate that transmit data can be written or receive data can be read.

You may read this register at any time but must write to it only when CTRL[TE] is 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				RXCOUNT				0				RXWATER			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TXCOUNT				0				TXWATER			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 RXCOUNT	Receive Counter Indicates the number of datawords in the receive FIFO or buffer. If a dataword is being received in the receive shift register, it is not included in the count. This value may be used in conjunction with FIFO[RXFIFOSIZE] to calculate the room left in the receive FIFO or buffer.
23-19 —	Reserved
18-16 RXWATER	Receive Watermark Generates an interrupt or a DMA request if the number of datawords in the receive FIFO or buffer is greater than the value of this field. For proper operation, the value of this field must be less than the size of the receive FIFO or buffer, as indicated by FIFO[RXFIFOSIZE] and FIFO[RXFE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-12 —	Reserved
11-8 TXCOUNT	Transmit Counter Indicates the number of datawords in the transmit FIFO or buffer. If a dataword is being transmitted to the transmit shift register, it is not included in the count. This value may be used in conjunction with the value of FIFO[TXFIFOSIZE] to calculate the room left in the transmit FIFO or buffer.
7-3 —	Reserved
2-0 TXWATER	Transmit Watermark Generates an interrupt or a DMA request when the number of datawords in the transmit FIFO or buffer is equal to or less than the value of this field. For proper operation, the value of this field must be less than the size of the transmit buffer or FIFO, as indicated by FIFO[TXFIFOSIZE] and FIFO[TXFE].

68.5.5.14 Data Read-Only (DATARO)

Offset

Register	Offset
DATARO	30h

Function

Indicates the first entry in the receive FIFO, but does not pull data from the FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Receive Data Indicates the first entry from the receive FIFO. This register has the same functionality as that of Data (DATA).

68.5.5.15 Transmit Command Burst (TCBR0 - TCBR127)

Offset

For a = 0 to 127:

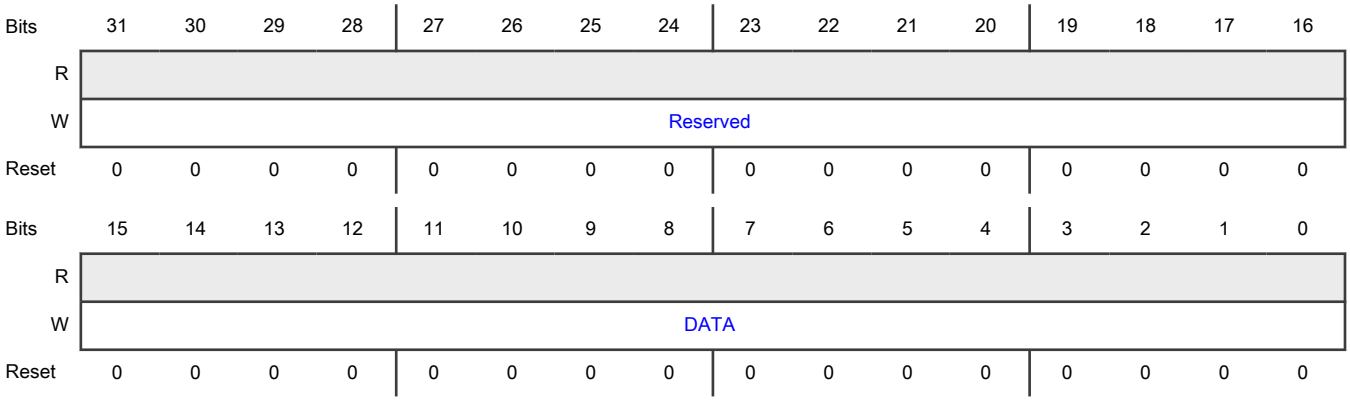
Register	Offset
TCBRa	200h + (a × 4h)

Function

Acts as an alias of Data (DATA), designed to support incrementing burst transfers to the transmit FIFO by a DMA controller, using aligned 8-bit, 16-bit, or 32-bit writes. The size of this register is 512 bytes:

- An aligned 32-bit write in this region pushes one entry into the transmit FIFO.
- An aligned 16-bit write in this region to TCBR*x*[15:0] pushes one entry into the transmit FIFO.
- An 8-bit write in this region to TCBR*x*[7:0] updates DATA[7:0], but does not push the data into the transmit FIFO.
- An 8-bit write in this region to TCBR*x*[15:8] pushes the data written to DATA[15:8] plus the previously written DATA[7:0] into the transmit FIFO.
- An 8-bit or 16-bit write in this region to TXBR*x*[31:16] is ignored.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Data Enables writing data to Data (DATA).

68.5.5.16 Transmit Data Burst (TDBR0 - TDBR255)

Offset

For a = 0 to 255:

Register	Offset
TDBRa	400h + (a × 4h)

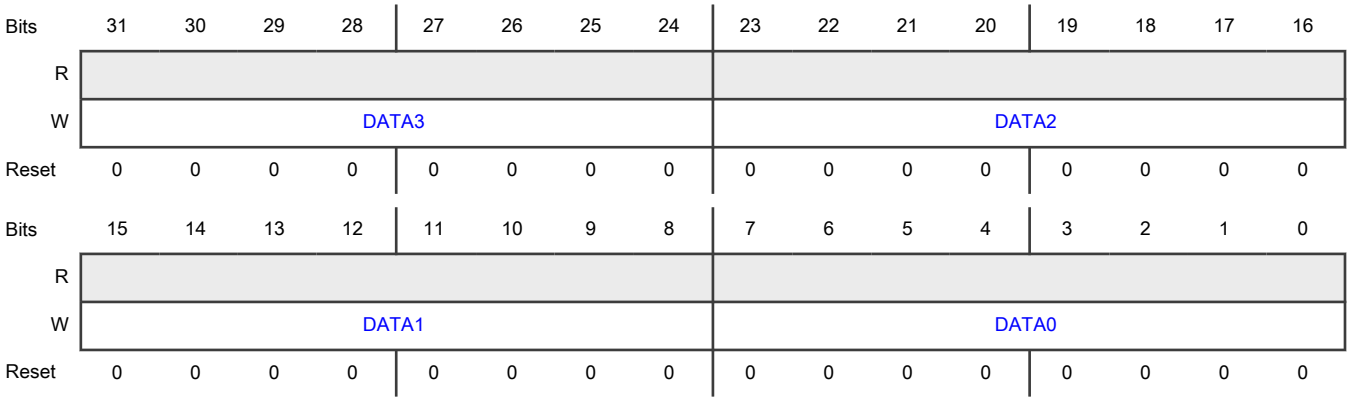
Function

Acts as an alias of Data (DATA), designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using 8-bit, 16-bit, or 32-bit writes. The size of this register is 1024 bytes:

- An aligned 32-bit write in this region pushes four DATA byte entries into the transmit FIFO (DATA0 byte first). The register access is extended by three wait states.
- An aligned 16-bit write in this region to either half of a 32-bit word pushes two DATA byte entries into the transmit FIFO (DATA0 or DATA2 first). The register access is extended by one wait state.
- An 8-bit write in this region pushes one DATA byte entry into the transmit FIFO.

Byte writes to Data (DATA) use the contents of CTRL[R9T8] for the ninth data bit, CTRL[R8T9] for the tenth data bit, and zero extend DATA[FRETSC].

Diagram



Fields

Field	Function
31-24 DATA3	Data3 Enables writing of data to Data (DATA).
23-16 DATA2	Data2 Enables writing of data to Data (DATA).
15-8 DATA1	Data1 Enables writing of data to Data (DATA).
7-0 DATA0	Data0 Enables writing of data to Data (DATA).

Chapter 69
Improved Inter-Integrated Circuit (I3C)

69.1 Chip-specific I3C information

Table 670. Reference links to related information

Topic	Related module	Reference
Full description	I3C	I3C
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

69.1.1 Module instances

This device has four instances of the I3C module, I3C0, I3C1, I3C2 and I3C3.

Table 671. I3C instances

	I3C0	I3C1	I3C2	I3C3
Located	VDD2_COMP	VDD2_COMP	VDD1_SENSE	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Need external pullup (working as controller)	Yes	Yes	No	No
Need external pullup (working as target)	Yes	Yes	Yes	Yes

69.1.2 High-Keeper methods

This device supports all High-Keeper methods.

69.1.3 Asynchronous wakeup control

When I3C is in controller mode, SYSCON controls I3C asynchronous wake-up enable. When I3C is in target mode, SINTSET[MATCHED] controls I3C asynchronous wake-up enable.

Table 672. I3C asynchronous wakeup control

I3C instances	I3C asynchronous wake-up enable register field	
	I3C in controller mode	I3C in target mode
I3C0	SYSCON0.I3C_ASYNC_WAKEUP_CTRL[IRQ_EN0]	I3C0.SINTSET[MATCHED]
I3C1	SYSCON0.I3C_ASYNC_WAKEUP_CTRL[IRQ_EN1]	I3C1.SINTSET[MATCHED]
I3C2	SYSCON1.I3C_ASYNC_WAKEUP_CTRL[IRQ_EN2]	I3C2.SINTSET[MATCHED]
I3C3	SYSCON1.I3C_ASYNC_WAKEUP_CTRL[IRQ_EN3]	I3C3.SINTSET[MATCHED]

69.2 Overview

I3C is a communications processor that improves upon the use and power of I2C, and provides an alternative to SPI for mid-speed applications.

The I3C bus protocol supports:

- In-band interrupts (IBI): these interrupts move from target to controller without extra wires, and the controller knows which target sent the interrupt
- Common command codes (CCC)
- Dynamic addressing
- Multi-controller and multi-drop
- Hot-Join (HJ)
- I2C compatibility

I3C supports all required and many optional features of the MIPI Alliance Specification for I3C, v1.0 and v1.1, except for ternary data rates (HDR-TSP and HDR-TSL). See Features, Target Capabilities (SCAPABILITIES), and Target Capabilities 2 (SCAPABILITIES2) for more information.

69.2.1 Block diagram

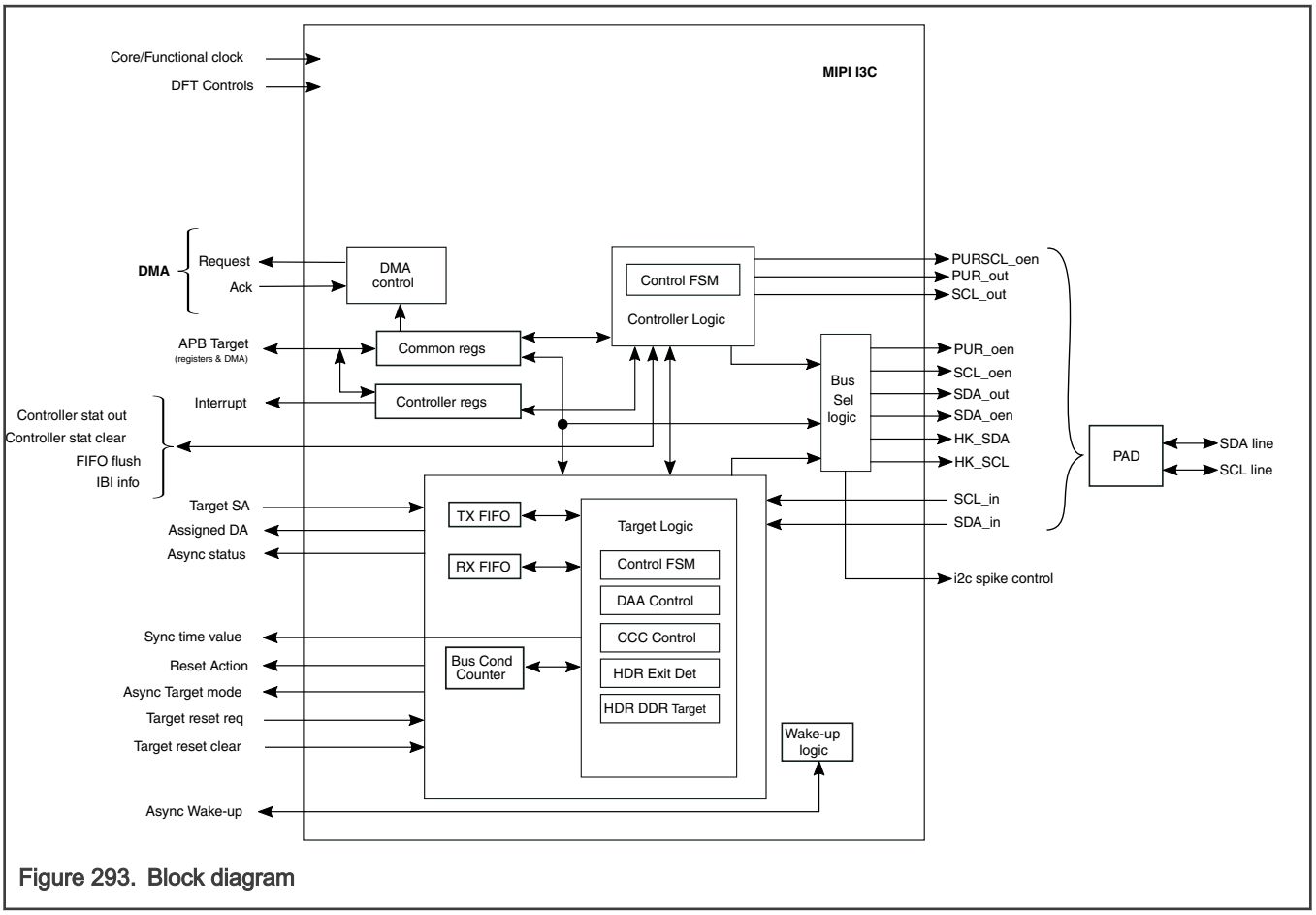


Figure 293. Block diagram

69.2.2 Features

- Two-wire multidrop bus that is capable of up to 12.5 MHz clock speeds with up to 11 devices:
  - Uses standard pads with 4 mA drive.
  - Dynamically assigns target addresses, and targets do not require static addresses (SAs). However, targets can have an I2C static address assigned at startup, so the target can operate on an I2C bus. By default, I3C supports 7-bit I2C-style addresses.
  - Allows targets to use the inbound SCL clock as the peripheral clock (instead of the clock from the controller) so that devices can have slow or inaccurate clocks internally.
  - Allows simple targets, such as temperature sensors, to have no internal clock.
  - I3C controller supports handoff from Open-Drain to Push-Pull mode for ACK to data transfer.
  - Generally, the controller terminates the read, but for I3C, the target can also end the read.
- IBIs that allow targets to send notifications to a controller:
  - Can be equivalent to a separate GPIO, but can also be directly data-bearing.
  - Can be prioritized. When multiple targets send interrupts to a controller at the same time, the order is resolved. Dynamic addresses (DAs) establish the priority of the targets, so the controller controls the priority of the targets.

    Targets with lower-value DAs are higher-priority level IBIs.

  - Can start interrupts even when the controller is not active on the bus. A free-running clock is not required, but starting an interrupt requires a Bus Available condition.
  - Can resolve an initial event via a time-stamping option, not requiring an interrupt.
- Built-in commands for applications created in software or firmware by using the register interface maintained in a separate space. These commands do not collide with normal controller-to-target messages and:
  - Control bus behavior, modes and states, low-power state, inquiries, and more.
  - Have an extra room for new built-in commands for other groups.
- Organized forms of multicontroller modes that are secondary controllers using clean handoffs between different controllers.
- Hot-Join onto I3C bus allows devices to connect to the bus later than when the bus starts. This:
  - Enables a device or module to access the I3C bus after power up or when physically inserted onto the I3C bus.
  - Provides a clean method for notification when new devices or modules access the I3C bus.
- I3C can use both I2C and I3C buses:
  - I3C supports specific legacy I2C devices on the bus.
  - I3C target devices can operate on I2C buses.
  - In I2C controller mode, it supports only Push-Pull mode on the SCL line.
  - A reset output is available for I2C mode SW reset for application usage. Asserted when the module is an I2C target and gets a I2C software General command from a controller. See the chip-specific section, for usage if this is available.
- Higher data rate modes are available:
  - I3C has a high data rate: double data rate (HDR-DDR) mode, which is double the data rate of SDR .
  - Only the controller and the specific targets, which are capable and configured to work with HDR-DDR, support the higher data rate (the other targets can ignore it).

I3C supports most of the I3C features (see Target Capabilities (SCAPABILITIES) and Target Capabilities 2 (SCAPABILITIES2)), except for the ternary data rates (HDR-TSP and HDR-TSL), HDR-BT modes, and peer-to-peer messaging.

69.3 Functional description

69.3.1 Operating modes

This section describes all functional operation modes of the I3C module.

69.3.1.1 Target and controller roles for I3C

The I3C protocol defines these roles for devices on the I3C bus:

- Main controller: initially configures the I3C bus and serves as the first active controller
- Secondary controller: accepts the controller role from any active controller to become the new active controller (the secondary controller may then pass the controller role along: either back to the previous active controller, or on to any other controller-capable device)
- Target: responds to commands from any I3C controller
- I2C target: responds to compatible commands from any I3C controller

The I3C peripheral contains both controller and target components and can be configured to be either controller or target, or as target with secondary controller capability. However, if I3C is chosen to be a controller, then the I3C peripheral supports Target mode to facilitate handoffs to multiple controllers.

In general, any I3C controller can be a controller or a target because a controller becomes a target when giving over control to another controller. The only exceptions to this rule occur when using point-to-point communication or when using a controller that never shares controllership.

69.3.1.1.1 Controller requirements

Controller mode uses software that supports the requirements of the controller (including as a secondary controller):

- Managing the enter dynamic address assignment (ENTDAA) assigning dynamic addresses (DAs) to each target. The I3C peripheral supports this process but requires you to make choices.
- Driving the serial data (SDA) signal in both Open-Drain and Push-Pull modes. After START command, SDA is in Open-Drain mode and after Repeated START command, it is in Push-Pull mode:
  - SDA is subject to arbitration because both controller and target can drive the SDA line (arbitration is also useful for handoff from controller to target).
  - The ninth bit of SDA controller write data is an odd parity bit.
- Building a table of targets and their capabilities to control which actions and commands may be sent to the targets.
- Managing requests such as IBI and controller requests (handoff).
- Adjusting clock speed or write-to-read timing to match the limitations of the target. This adjustment can be done in hardware using dividers and uneven duty cycles, but you must decide how.
- Adjusting the maximum data length.
- Being a target after the controller role handoff to another controller-capable device.

The controller needs an accurate clock capable of running at a frequency that is the multiple of a frequency between 11 MHz and 12.5 MHz. Following are the possible clock frequencies:

- 24 MHz (multiple of 12 MHz)
- 25 MHz (multiple of 12.5 MHz)

If required to use a lower I3C SCL value, for example as small as 5 MHz (although the lower value does not support a mixed bus system):

- The controller clock can be a multiplier of that value.
- A lower SCL can also be achieved using a higher PPBAUD value.

69.3.1.1.2 I3C target acts like I2C target on I3C buses

If the target is assigned an I2C static address, the I3C target acts like an I2C device when it first powers on. If the I3C target is placed on an I2C bus with an I2C controller, then the I3C target stays in I2C mode and operates normally. The software is aware that the I3C target is in I2C mode because:

- There has not been an interrupt (see SSTATUS[DACHG]) indicating that a dynamic address was assigned.

For full I2C support of Fast mode (Fm) and Fast-mode Plus (Fm+), the pads must support a 50-ns spike filter. This filter must be turned off when the I3C 7Eh broadcast address is received (indicating an I3C controller). You can turn off the spike filters via hardware using the raw net indicating that the address was received, or via software. I3C does not need a 50-ns spike filter to operate on an I2C bus. Therefore, the spike filter is not a requirement for the I3C peripheral.

Depending on the configuration, the module supports most I2C features. See Target Capabilities (SCAPABILITIES) and Target Capabilities 2 (SCAPABILITIES2) for more information.

69.3.1.1.3 How a target rejoins the I3C bus

When a target tries to rejoin the I3C bus following a power-up or hard reset, the target needs a new dynamic address (DA). The target can rejoin the I3C bus in these ways:

- If the DA is lost, Hot-Join is used.

- If the DA is retained in the peripheral (for example, with state retention flip-flops), SCONFIG[OFFLINE] is used.

When SCONFIG[SLVENA] is 1, SCONFIG[OFFLINE] may become 1. This setting causes the peripheral to rejoin the bus safely. It does so by ensuring that the I3C bus is not in HDR mode, using the same approach as TE0 or TE1 exit. The peripheral waits for the HDR exit pattern from the controller, or for 60 µs of SCL and SDA lines not changing, whichever occurs first.

After using SCONFIG[OFFLINE], the I3C peripheral cannot safely use IBI until SSTATUS[STOP] becomes 1. This status ensures that the next START is safe to use for IBI.

If the application has to perform an IBI, it must wait for SSTATUS[STOP] to become 1, or for SCL and SDA lines to remain high for 200 µs. You can perform this process using the SSTATUS and SINTSET controls:

- If Target Status (SSTATUS) indicates that the bus is not busy and the peripheral interrupts on START or STOP, use a timer to measure 200 µs. If the timer finishes with no START or STOP, it is safe to use IBI.
- If a START causes an interrupt, then the timer must be turned off and the application must wait for a STOP.
- If a STOP causes an interrupt, it is safe to use IBI.

69.3.1.2 Using the I3C controller for I2C and I3C

The I3C controller operates with the following set of built-in capabilities with the application flow:

- Built-in enter dynamic address assignment (ENTDAA) mechanism to simplify the assignment of dynamic addresses to targets. This feature is used when set dynamic address from static address (SETDASA) is not available.

  NOTE

  When SETDASA CCC is available, use SETDASA CCC before using ENTDAA CCC; all targets without assigned dynamic addresses respond to ENTDAA CCC.

- Request for START + address with IBI support, including both I2C and I3C modes.
- SDR write flow via FIFO, with automatic parity in I3C and ACK or NACK detection in I2C.
- SDR read flow via the FIFO, with an automatic NACK generator for I2C, and optional use of a terminate generator for I3C.
- Request for Repeated START + address or STOP when finished with the previous request, including both I2C and I3C.
- Auto-IBI mode, which responds immediately to a target-initiated IBI and can be used when in Sleep or Deep Sleep mode.
- Special message mode for SDR and DDR to simplify DMA use.
- Automated SCL, SDA, pullup, and High-Keeper controls.
- I2C and I3C frequency and duty cycle configurations from functional clock.

  NOTE

  You must configure Controller Configuration (MCONFIG) and Controller In-band Interrupt Registry and Rules (MIBIRULES) before using the controller.

69.3.1.3 Protocol modes and states

The following modes are activated in I3C:

- I2C mode is the default mode on startup:
  - If no I2C static address is used, this mode has no effect other than to track frames looking for I3C transitional frames.
  - If an I2C static address is used, the device can interact with the I2C bus controller using the address.
- When in I2C mode, I3C Transitory Frame mode occurs whenever the I3C broadcast address is received (7Eh). In particular:
  - This mode occurs when 7Eh is broadcast followed by SETDASA from the controller. If there is a static address match, the target is assigned a dynamic address and it enters I3C SDR mode.

- This mode occurs when 7Eh is broadcast followed by ENTDAA from the controller. If the target can send a 48-bit provisioned ID (48b ID), a dynamic address is assigned, and the target enters I3C SDR mode.
- If a Hot-Join arbitrated event occurs (sending 02h when the controller generates another address such as 7Eh), there is a conceptual Hot-Join mode. ENTDAA or SETDASA sets a dynamic address to resolve the event, and the target enters I3C SDR mode.

NOTE

An I3C target can operate as a normal I2C target only when it has a static address. Otherwise, it matches nothing in I2C and waits for the aforementioned events.

- I3C SDR mode is the standard mode after I3C activates, which is defined by a dynamic address being assigned. This mode is the resting mode of I3C, and all devices are normally in SDR mode:
  - RSTDAA CCC causes an exit from SDR mode and a return to I2C mode. This transition is not normally used.

    RSTDAA can be issued to ensure that all I3C targets participate in an upcoming dynamic address assignment process, before the I3C controller starts assigning the target address.
  - SETNEWDA does not affect the SDR mode; it only changes the dynamic address of an I3C device that already had a dynamic address assigned.
  - When in SDR mode, the device ignores messages to or from its original I2C static address.
- The I3C CCC command submode of type Direct or Broadcast. Following are the options to exit the CCC:
  - Exit the Broadcast CCC submode by a Repeated START or by a STOP.
  - Exit the Direct CCC submode by a 7Eh broadcast address after a Repeated START or by a STOP.
  - I3C dynamic address assignment (DAA) CCC command enters DAA mode. This mode is a special mode for dynamic addressing. Use STOP to exit. If a target has a dynamic address, the command is ignored.
- I3C SETDASA CCC command enters Static Address Match submode, which allows matching the static address of the device (if any). The command is ignored when a target has a dynamic address, or when the target in I2C mode has no static address.

NOTE

The special point-to-point address is also matched.

- I3C HDR modes are activated by ENTHDR*n* CCC commands, where *n* represents the type of HDR (0 to 7). This mode is valid until an HDR exit pattern is detected, irrespective of whether HDR mode for a target is supported.

  Exit-pattern detection must always be on to prevent errors. Alternatively, you could set it to be activated only on ENTHDR.
- A special HDR-DDR syntax is used for transmitting CCC in the HDR-DDR protocol.
- Internal states such as IBI or no-IBI and Low-Power mode are flagged internally (see SSTATUS[EVDET] for more information). These states are exported to the application and affect the engine to restrain it from performing a prohibited operation.

69.3.1.4 Address match

For an address match to occur, the target matches the I3C broadcast address and either the I2C-style static address or I3C dynamic address (see Table 673).

The address match is inactive (just listening) for all Repeated START commands and also for the START command, unless an IBI, CR, or Hot-Join has been activated. If inactive, it simply tries to match. If not matched, it waits for a Repeated START or STOP command. If an IBI, CR, or Hot-Join has been activated, the arbitration mechanism is used for START (but never for a Repeated START).

When matching the 7Eh broadcast address, it looks for a CCC until the next Repeated START or STOP command.

Table 673. Address match conditions

For	Match condition occurs when	Notes
I3C broadcast address	The address is 111 1110b (written as 7Eh).	
I2C-style static address	• The device has a static address. • The device is not in I3C mode.	When in I3C mode, the address matches only until the ENTDAA, SETDASA, or SETAASA command has a dynamic address assigned. SETDASA matches the static address or the special one-controller-to-one-target point-to-point address.
I3C dynamic address	ENTDAA or SETDASA assigns it, or SETNEWDA modifies it.	

69.3.1.5 I3C from low power mode and wake-up

A wake-up interrupt is available to bring the I3C module out of low power mode.

This interrupt works in both Controller and Target mode of operation as described below.

69.3.1.5.1 Controller mode

In Controller mode, I3C works with 2 clocks.

- Host interface clock for accessing registers and DMA
- Functional clock (FCLK) for controller logic operation and generating SCL

The Functional clock can be switched off to keep the I3C controller in low power mode if there is no data to be sent by the host and the i3C bus is in idle state (SDA = 1).

Before switching off the clock, take the below steps.

- Ensure that you enable the Controller mode MCONFIG[MSTENA] = 01b.
- Ensure that the bus is in the idle state.
- Switch off the Functional Clock.

If any agent on the bus pulls the SDA line low and at the SoC level the ENABLE bit of the I3C SDA interrupt is set to 1, then the interrupt goes high to inform the SoC logic to switch on the FCLK again.

69.3.1.5.2 Target mode

In Target mode, I3C works with only the Host interface clock (for accessing registers and DMA)

The main target control logic works on the external SCL clock.

The Host interface clock can be switched off to keep the I3C Target in low-power mode.

Before switching off the clock, take these steps.

- Ensure that you configure the Target mode, MCONFIG[MSTENA] = 00b along with all other normal target configuration as required.
- Clear the interrupts in Target Status (SSTATUS) after handling them.
- Enable DA matched interrupt in SINTSET[MATCHED].
- The host can now stop the Host interface clock to enter low power mode.

The interrupt asserts when any address match happens, either DA or SA or group address with this target.

The host then resumes the Host interface clock to serve and clear the interrupt.

69.3.2 Operations

This section describes the operations of the module.

69.3.2.1 Reading and writing I2C messages using the normal method

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when the module completes an MCTRL request.
   - COMPLETE: Indicates when data is finished sending or being received.
   - RXPEND: Used for read operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use this field to allow DMA to read out data.
   - TXNOTFULL: Used for write operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use this field to allow DMA to supply data.
   - IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Configure the following fields in Controller Control (MCTRL) simultaneously:
   a. Write 1 to REQUEST (EmitStartAddr).
   b. Write 1 to TYPE (I2C).
   c. Configure IBIRESP to respond to IBIs in your chosen manner.
   d. Write 1 to DIR for read, or write 0 for write.
   e. Write the static address of the I2C target to ADDR.
   f. Configure RDTERM to the maximum length for read operations, to auto-terminate, or set it to STOP as the data is read out. For example, write 1 (with MCTRL[REQUEST] = 0) to stop after the next character.
3. Write or read the data.
   - For write operations, write to Controller Write Data Byte (MWDATAB) for each byte before the last byte, and then write to Controller Write Data Byte End (MWDATABE) for the last byte:
     — You can perform or start this operation before setting up interrupts.
     — If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level. This interrupt allows the application to provide more data or to use DMA.
   - For read operations, wait for RXPEND, and then read out data via Controller Read Data Byte (MRDATAB). DMA may also be used.
4. After message transfer is complete (MSTATUS[COMPLETE] = 1), the message may end with a STOP condition, or a new message may start with a Repeated START condition:
   a. Write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for the MCTRLDONE status to be asserted for its completion. (When sending STOP in I2C mode, MCONFIG[ODSTOP] and MCTRL[TYPE] must be 1.)
   b. Write 1 to MCTRL[REQUEST] (EmitStartAddr to restart). IBI is not possible in this case.

NOTE

I2C Fm and I2C Fm+ modes are supported for legacy I2C devices as per the I3C standard specifications. See I2C configuration to meet timing requirements for Fm and Fm+ modes for more information.

69.3.2.2 Reading and writing I3C messages using the normal methods (SDR and HDR-DDR)

The normal method for I3C is the same as the method for I2C with a few differences:

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when the I3C module completes an MCTRL request.
   - COMPLETE: Indicates when data is finished sending or being received.
   - RXPEND: Used for read operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use it to allow DMA to read out bytes.
   - TXNOTFULL: Used for write operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use it to allow DMA to supply bytes.
   - IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Set up Controller Control (MCTRL).
   - Option 1: Configure the following fields of Controller Control (MCTRL) simultaneously in this way:
     a. Write 1 to REQUEST (EmitStartAddr).
     b. Write 0 to TYPE for I3C Single Data Rate (SDR) mode or write 2 for Double Data Rate (DDR) mode.
     c. Configure IBIRESP to respond to IBIs in your chosen manner.
     d. Write 1 to DIR for read, or write 0 for write.
     e. Write the dynamic address of the I3C target to ADDR.
     f. For read operations, you can configure RDTERM to the maximum length to terminate automatically.
     g. For write operations, prewriting the data (MWDATAB or MWDATAH) is preferred to ensure that there are no time delays waiting for the data.

        For DMA with MCTRL, use either Controller Write Byte Data 1 (to Bus) (MWDATAB1) or Controller Write Halfword Data (to Bus) (MWDATAH1).

     NOTE

     HDR-DDR mode requires writing an 8-bit command value for read or write. This value must be written into the TX FIFO via Controller Write Data Byte (MWDATAB). The END field is not used for this byte.

   - Option 2: This option is preferred when stopped (bus free condition) in SDR mode, and not in HDR-DDR mode. It allows any target to issue an IBI and avoids collisions with an IBI address. Also, it is faster (when the MSB of the dynamic address is always 0).

     Configure the following fields of Controller Control (MCTRL) in this way:
     a. Write 1 to REQUEST (EmitStartAddr). No prewritten transmit data can be in the FIFO.
     b. Write 0 to TYPE (I3C).
     c. Configure IBIRESP to respond to IBIs in your chosen manner.
     d. Write 0 to DIR.
     e. Write 7Eh to ADDR.
     f. Wait for MCTRLDONE (via interrupt, for example), then proceed as in option 1. The option 2 method has advantages for IBIs when 7Eh is sent on START (but not on Repeated START conditions).
3. Write or read the data.
   - For write operations, write to MWDATAB for each byte before the last byte, and then write to Controller Write Data Byte End (MWDATABE) for the last byte. For HDR-DDR, the byte with END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs:

— You can perform or start this operation before step 1 (REQUEST = 1) of the option 1 method, but not before the option 2 method.

— If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level. This interrupt allows the application to provide more data or to use DMA.

- For read operations, wait for RXPEND, and then read out data via Controller Read Data Byte (MRDATAB) or Controller Read Data Halfword (MRDATAH). DMA may also be used. When using DMA, read using the same registers.

4. After message transfer is complete (MSTATUS[COMPLETE] = 1), the message may be ended with STOP (or EXIT in HDR mode). Alternatively, a new message may be started with a Repeated START (or HDR-Restart in HDR mode):
   - In SDR mode, write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for the MCTRLDONE status to be asserted for its completion.
   - For HDR mode, write 6 to MCTRL[REQUEST] (ForceExit) to end HDR mode. Then wait for the MCTRLDONE status to be asserted for its completion. When sending the HDR exit pattern, MCONFIG[ODSTOP] must be 0.
   - Write 1 to MCTRL[REQUEST] (EmitStartAddr to start another message). IBI is not possible in this case.

69.3.2.3 Determining bus types and modes

The settings of MCTRL[TYPE] and MCTRL[REQUEST] determine bus types and modes.

Table 674. Determining bus types and modes

Value of MCTRL[TYPE]	Meaning when MCTRL[REQUEST] = 1 (EmitStartAddr)	Meaning when MCTRL[REQUEST] = 2 (EmitStop)	Meaning when MCTRL[REQUEST] = 6 (ForceExit)
0	SDR mode of I3C	SDR mode of I3C	Exit pattern
1	Standard I2C protocol	Standard I2C protocol	Reserved
2	The bus enters DDR mode (7E and then ENTHDR0), if the module is not in DDR mode already. The first byte written to the transmit FIFO must be a command and already in the FIFO. To end DDR mode, use ForceExit.	Reserved	Target reset
3	If not already in HDR-BT, the bus automatically enters BT mode (7E and then ENTHDR3). You must also configure the MHDRBTCFG register for multilane and CMD rules.	Reserved	Reserved

Table 675. Deriving the dynamic address

Value of MCTRL[TYPE]	Meaning when MCTRL[REQUEST] = 4 (ProcessDAA)
0	The dynamic address is derived from MWDATAB[VALUE]; in case of a NACK, ProcessDAA continues to the next 7E/R request.
1	Reserved
2	The dynamic address is derived from MWDATAB[VALUE]; in case of a NACK, a STOP condition is sent after the assignment.
3	Reserved

69.3.2.4 Sending a CCC to I3C targets

The normal common command code (CCC) method uses an I3C write operation with an address of 7Eh (the CCC is the first byte):

- For broadcast type, any remaining bytes are sent with the CCC. This operation ends with a STOP or a Repeated START and 7Eh.
- For direct type, only the CCC byte is sent (or by a CCC and a defining byte, if required by the CCC).

  These bytes are followed by a Repeated START and the address of the I3C target (for SETDASA, this address is its I2C static address). This sequence may be repeated with more Repeated START conditions and addresses until done. It may end in STOP or a Repeated START and 7Eh. After the Repeated START and target address, the values are read or written depending on the CCC.
- I3C provides interrupts (by hardware) for unhandled and handled CCC when received by the target. Its state is reflected in Target Status (SSTATUS).

I3C controllers require to emit a single START, 7E/W sequence with both SCL high and SCL low half periods at full open-drain timing (for example, 200 ns). This sequence allows I3C targets acting as I2C legacy devices to turn off their I2C 50-ns spike filters, if they have them.

NOTE

START, 7E/W is a notation indicating a START command, followed by the 7Eh broadcast address and then by a write command.

After that sequence, addresses following START may be sent with Open-Drain Low (for example, 200 ns) but high of Push-Pull timing (for example, 40 ns). The I3C controller must emit this START, 7E/W sequence with MCONFIG[ODHPP] = 0. ODHPP is Open-Drain High period at Push-Pull speeds, and MCONFIG[ODHPP] is usually 1.

NOTE

- MCONFIG[ODHPP] is ignored when sending a message to an I2C legacy device on an I3C bus.
- Each Repeated START is a new EmitStart request. This request can be chained by an interrupt or pushed by a message model using DMA.

69.3.2.5 IBI handling

An IBI occurs when a target sends its address after a START, and that address is numerically the lowest. That is, it is lower than the address sent by the controller and addresses sent by any other targets. When the controller sends 7Eh, the controller always loses the arbitration.

The IBI can occur unexpectedly when any new START (not a Repeated START) is sent. The IBI can also occur in response to a target pulling SDA low. This condition can occur in one of these ways:

- The controller has set the request to AutoIBI mode, so the IBI occurs automatically. The controller sends 7Eh to allow the target to win the arbitration.
- The application receives the SLVSTART (target START request) interrupt, so it sends 7Eh.

The IBI response is configured by MCTRL[IBIRESP], which can be set to:

- NACK (always reject the IBI).
- ACK (always accept the IBI):
  - You must configure Controller In-band Interrupt Registry and Rules (MIBIRULES) so that the engine knows whether IBI bytes follow.
  - If IBI bytes follow (also known as IBI mandatory byte), then the COMPLETE field does not become 1 when IBIWON becomes 1. RXPEND becomes 1 for one or more bytes in the receive FIFO. When the last byte is received, then the COMPLETE field becomes 1. The controller automatically stops IBI data after 9 bytes (including the mandatory data byte). MCTRL[RDTERM] can be used with MCTRL[REQUEST] = 0 to end the process of receiving data sooner. I3C

supports address ACK to mandatory byte transition during IBI from Open_Drain (controller acknowledges the target address) to Push Pull (target sends SDR data).

— I3C target supports up to 7 bytes (maximum limit is defined by IBIEXT1[MAX]) of extended IBI data following a mandatory data byte. Extended data to send, if any, is present in Extended IBI Data 1 (IBIEXT1) and Extended IBI Data 2 (IBIEXT2).

- Manual (allow the decision to be made by the application on a case-by-case basis):
  — The application rewrites it when stopped, pending an IBI.
  — The application can ACK or NACK the IBI based on the IBI address defined in MSTATUS.
  — This mode selects whether there is an IBI byte when accepting the IBI request from targets.

Accurate timestamping of I3C target data is supported in I3C though Async mode 0. In I3C Asynchronous Timing Control mode, a target device timestamp event occurs within that target. The target notifies the controller about the event by generating an IBI.

69.3.2.6 Assigning dynamic addresses to I3C devices

If dynamic addresses (DAs) are all assigned below 40h (7-bit values from 3Fh down to 03h, except where not allowed), then MIBIRULES[MSB0] can be 1. This setting optimizes the START timing. When dynamic addresses are assigned, you must program Controller In-band Interrupt Registry and Rules (MIBIRULES) based on which I3C target uses IBI bytes (present in SIDEXT[BCR]).

Any SETDASA assignments can be performed via the normal CCC model. These assignments cannot be performed when using the static address for the directed part.

There is a built-in mechanism to process ENTDAA mode:

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when a target has sent its ID and BCR or DCR, and a new DA is needed.
   - COMPLETE: Indicates when DAA is done (NACKed by all targets).
   - RXPEND: Indicates the reading of the IDs of the targets. You can also use this field to allow DMA to read out bytes.
   - IBIWON: Indicates when an IBI has occurred (normally there are no IBI conditions when assigning dynamic addresses to the I3C device).
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Configure Controller Control (MCTRL):
   a. Write 4 to MCTRL[REQUEST] (ProcessDAA).
   b. Set MCTRL[IBIRESP] to IBI response, if possible. If not possible, set to the first target assignment.
3. Wait for the MCTRLDONE interrupt when reading ID using the RXPEND interrupt. If MSTATUS[STATE] = 5 (DAA mode) and MSTATUS[BETWEEN] = 1, it is waiting to write a dynamic address:
   a. Write the dynamic address into Controller Write Data Byte (MWDATAB) using bits 6:0, such as 14h for DA = 7'h14.
   b. Write 4 to MSTATUS[STATE] (ProcessDAA again). This option writes the DA, then moves to the next DA.
   c. If MSTATUS[COMPLETE] = 1 and MSTATUS[STATE] = 0, then all targets are assigned.
   d. If MSTATUS indicates NACK after writing a new DA, the DA is not accepted by the target. The next step is to write 2 to MCTRL[REQUEST] (EmitStop) and start over. It is also acceptable to write 4 to MCTRL[REQUEST] (ProcessDAA again).

69.3.2.7 Using Controller Message mode

Controller Message mode is intended for use with DMA although Message mode can also be used by the processor. In Message mode, all writes are to the same location, including control and data. Reads occur from an associated location.

NOTE

The data writes for Message mode work in the same way as the halfword access registers, Controller Write Data Halfword (MWDATAH) and Controller Read Data Halfword (MRDATAH).

To send a message via Single Data Rate (SDR), follow these steps (from DMA or processor):

1. Write the control request to Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL). This request includes the following items:
   - The address.
   - I2C or I3C.
   - Read or write.
   - The count of bytes to process.
   - How to end (on STOP or ready for Repeated START).
2. Process the data:
   - For write operations, write the rest of the data to Controller Write Message Data in SDR mode (MWMSG_SDR_DATA). Use the DMA trigger (or interrupt) to keep the transmit FIFO full, and the controller stops using the data when it has already consumed the number of bytes configured in the LEN field.
   - For read operations, a DMA trigger (or interrupt) indicates when the RX FIFO is ready to be read out via Controller Read Message in SDR mode (MRMSG_SDR).
3. Select the in-band interrupt (IBI) behavior in MCTRL[IBIRESP].
4. Use END type STOP (MCTRL[REQUEST] = 2) with a zero-length message or by using Controller Control (MCTRL). To exit Message mode with the original message, use END type STOP (MCTRL[REQUEST] = 2) with a zero-length message or by using the MCTRL register.

To send a message via Double Data Rate (DDR), follow these steps (from DMA or processor):

1. Write the control request to Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL). This request includes:
   - The count of byte pairs.
   - How to end the message (through HDR exit or to wait for HDR restart).
2. Write the second control data to MWMSG_DDR_CONTROL. This second write includes:
   - The address.
   - Read or write.
   - The 7-bit command value.
3. Process the data:
   - For write operations, write the rest of the data to MWMSG_DDR_DATA. Use the DMA trigger (or interrupt) to keep the transmit FIFO full, and the controller stops using the data when the program count MWMSG_DDR_CONTROL register reaches 0.
   - For read operations, a DMA trigger (or interrupt) indicates when the receive FIFO is ready to be read out.
4. Select in-band interrupt (IBI) behavior in MCTRL[IBIRESP].
5. Use END type exit (MCTRL[REQUEST] = 6) with a zero-length message or by using the MCTRL register to exit DDR Message mode with the original message.

69.3.2.8 Handing off controllership to another target and getting it back

To hand off controllership, the controller can perform either of the following actions:

- Wait for a controller request (CR) (that is, MSTATUS[IBIWON] = 1 interrupt), indicating that the CR is using MSTATUS[IBITYPE].
- Push the request manually.

In either case, the controller sends a GETACCMST request, which is a directed GET informing the target that it is being assigned controllership. If the target accepts this request, it returns its dynamic address in bits 7:1 and the negative parity of its dynamic address in bit 0. A STOP must be issued, and MCONFIG[MSTENA] must be set to 2 (switching to Target mode).

To gain controllership, the target sends a CR using Target Control (SCTRL):

- If MCONFIG[MSTENA] is 2, the GETACCCR CCC is ACKed.
- If MCONFIG[MSTENA] is not 2, the GETACCCR CCC is NACKed.

After controllership is granted, MSTATUS[NOWMASTER] becomes 1. The application must enable MINTSET[NOWMASTER], so the application is interrupted when a controllership transfer occurs.

69.3.2.9 Controller engine flow diagram

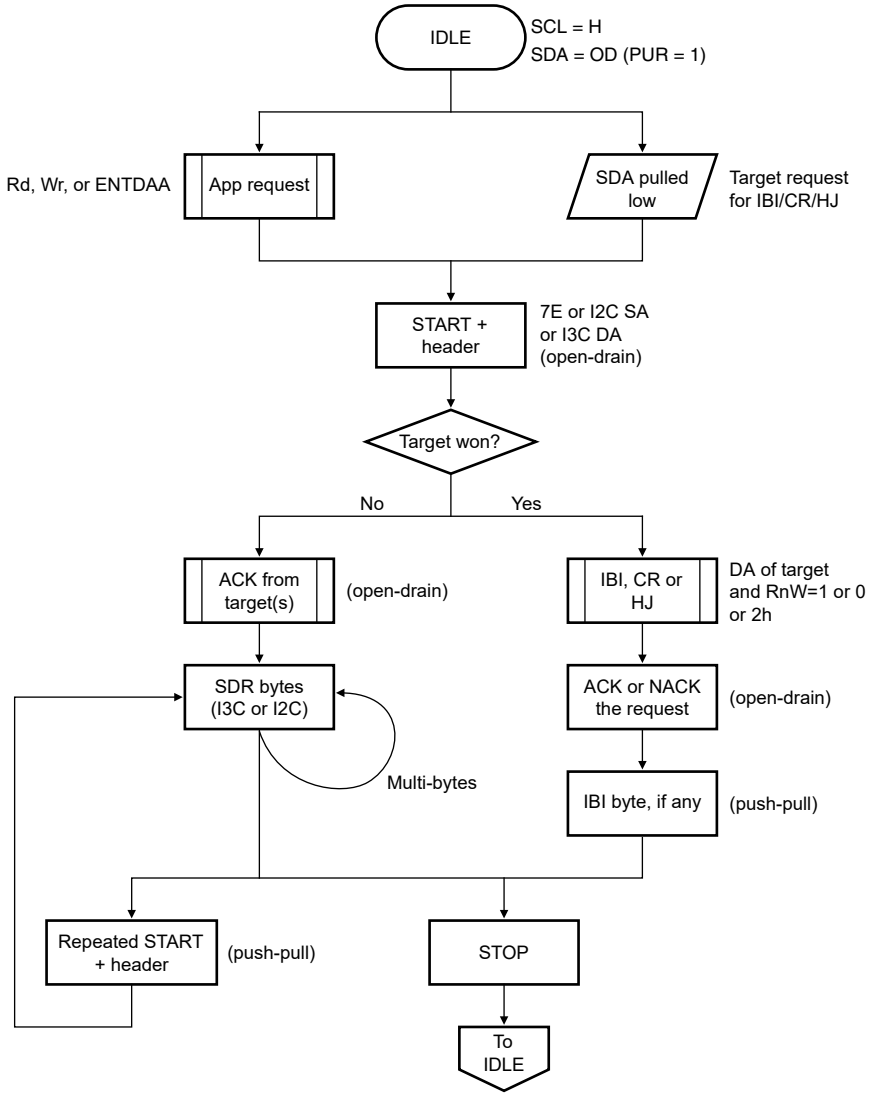


Figure 294. Controller engine flow diagram for SDR (HDR flow progresses from SDR bytes)

69.3.2.10 Target receives data from controller

When the address match is valid for the target dynamic address and the type is W (write from controller), the target ACKs the address. It does so unless some condition prevents it, such as a full buffer. On ACK, the Write state is entered. Once ACKed, it waits for each complete data byte. This process occurs in two steps:

1. Eight bits are clocked in and stored in the next buffer location when in the Write state.
2. On the ninth bit:
   - In I3C mode, the W9TH state (in I3C, the ninth data bit written by the controller is the parity of the preceding eight data bits) is used and the parity is checked. The buffer is marked as complete with or without a parity error.
   - In I2C mode, the W9TH state (in I2C, the ninth data bit written by the controller is an ACK by the target) is used and the data is ACKed and stored.

When the address match is valid for the 7Eh broadcast address, the target acknowledges it. After first data completion, it sets the in_ccc broadcast or direct flag (based on bit 7 of the command). If recognized, it parses the command, setting that bit as well. The recognized commands are for dynamic address work and modes. The CCC and DAA blocks handle the workload afterward. If not supported, then the commands are passed up to the system.

69.3.2.11 Target sends data to controller

When the address match is valid for the target dynamic address, and the type is R (read by controller), the target ACKs the address. The target does so unless there is no data waiting for the read. On ACK, the Read state is entered.

After being ACKed, it emits the data byte at a time, with the ninth bit using the R9TH state. In I2C, the ninth data bit from target to controller is an ACK by the controller. In I3C, this bit allows the target to end a read, and allows the controller to abort a read:

- In I3C mode, the T bit is emitted to allow the device to indicate whether this byte is the last. If it is not the last byte, the controller may terminate the read via the T bit.
- In I2C mode, the ninth bit allows the controller to terminate via NACK, or else it allows the read to continue via ACK. On completion of each byte read, the "done" signal is pulsed to get the next byte.

If the read is terminated by the controller unexpectedly (abort in I3C, NACK before END marked), the application is notified.

69.3.3 Clocking

The controller works on the following primary clocks:

- The system clock, which controls access to the memory-mapped registers.
- A functional clock (FCLK), which is used to generate the SCL clock rate on the I2C or I3C bus.

I3C supports adjusting clock frequency and accuracy via Target Time Control Clock (STCCLOCK).

FCLK is used to check conditions such as bus availability, bus free for IBI, or HJ:

- To determine the Bus Available condition for IBI, the target needs a clock to generate the ~1 µs timing. To support this requirement, a slow clock (CLK_SLOW) is provided to save on power.
- I3C supports a counter field, SCONFIG[BAMATCH], to calculate the Bus Available condition. BAMATCH provides the count of the slow clock . This field counts 1 µs or more to allow an IBI to drive SDA low when the controller is free. The I3C module controls the maximum width and maximum values.

I3C supports a timeout when stalled for too long in a frame. This timeout occurs when:

- The transmit FIFO or receive FIFO is not handled and the bus is stuck in the middle of a message.
- No STOP is issued and the bus is between messages.
- IBI manual is used and no decision is made.

69.3.4 Reset

- Global or system reset fed into the module: Everything is reset by this global or system reset. Release is assumed to be synchronized with the system clock. It is not synchronized with SCL or SDA. These signals must not be active when the reset is released; otherwise, I3C enters Hot-Join mode and remains inactive.
- Software reset

69.3.5 Interrupts

This section describes all the interrupts (IRQs) that this module generates. All status interrupts are updated in Controller Status (MSTATUS) and Target Status (SSTATUS).

Supported interrupts occur:

- For pending IBI, CR, or Hot-Join that have been sent by a target.
- When a CCC is received and is handled by the module (SSTATUS[CHANDLED]).
- When an error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR, or CRC error.

A wake-up interrupt is also available to bring the I3C module out of low power mode. See I3C from low power mode and wake-up .

69.4 External signals

Table 676. External signals

Signal	Description	Direction
SCL	Serial clock	Input or output
SDA	Serial data	Input or output
PUR	Pull up resistance. There is an internal pullup resistance on SDA, which the I3C controls. If the internal pullup is not enough, PUR can control an external pullup resistance on SDA actively. See the chip-specific I3C information on whether this external pullup is available on your chip.	Output

69.5 Initialization

69.5.1 Controller configuration

Certain configuration is performed when initializing this module. Controller Configuration (MCONFIG) controls the frequencies, duty cycle, optimizations for performance, and other parameters.

Table 677. Configuration for module initialization

Configuration parameter		Description
MSTENA	Controller enable	Determines whether the peripheral starts in Controller mode (main controller in I3C terms) or starts in Target mode (and switches to Controller mode later).
HKEEP	High-Keeper	Determines how the high-keeper (weak pullup) is implemented, depending on the device capabilities.
PPBAUD	Push-Pull baud	Sets the push-pull frequency as a divider from the FCLK (alternative clock fed to the peripheral). This frequency sets the SCL half-clock period baseline (used at least for the high time of SCL).

Table continues on the next page...

Table 677. Configuration for module initialization (continued)

Configuration parameter		Description
		The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL frequency (in MHz) = FCLK ÷ ((2 + 2 × PPBAUD) + PPLOW) If PPLOW is 0, then SCL high = SCL low (in ns) = (1 + PPBAUD) × 1000 ÷ FCLK (in MHz) Examples: • If FCLK = 24 MHz, then PPBAUD = 0 yields 12 MHz (42.67 ns per half period) • If FCLK = 50 MHz, then PPBAUD = 1 yields 12.5 MHz (20 + 20 = 40 ns per half period)
PPLOW	Push-Pull low	Changes the duty cycle for Push-Pull. It indicates how many more FCLK cycles to use for low. The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL frequency (in MHz) = FCLK ÷ ((2 + 2 × PPBAUD) + PPLOW) If PPLOW is a nonzero value, then: • SCL high (in ns) = (1 + PPBAUD) × 1000 ÷ FCLK (in MHz) • SCL low (in ns) = (1 + PPBAUD + PPLOW) × 1000 ÷ FCLK (in MHz) For example, when FCLK is 50 MHz, PPBAUD is 1, and PPLOW is 1, then the periods are: • (1 + 1) × 1000 ÷ 50 = 20 + 20 = 40 ns high • (1 + 1 + 1) × 1000 ÷ 50 = 20 + 20 + 20 = 60 ns low This timing is equivalent to 10 MHz SCL, but this timing maintains the 40 ns high needed; therefore, I2C devices do not see the high periods. **NOTE** The PPLOW value does not have any impact on Open-Drain mode and I2C mode SCL rate calculations.
ODBAUD	Open-Drain baud	Determines the number of PPBAUD periods to make up one I3C Open-Drain half-clock baseline. If ODHPP is 0, the ODBAUD half-clock time period = (ODBAUD + 1) × PPBAUD high period. If ODHPP is 0, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK ÷ (2 + 2 × PPBAUD) × (ODBAUD + 1) Target to get open-drain half-clock period is 200 ns. For example: • If PPBAUD yields 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 can be used to get 200 ns. See ODHPP for details on short high and long low. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4, then open-drain SCL = 50 ÷ (2 + 2 × 1) × 5 = 2.5 MHz.
ODHPP	Open-Drain High Push-Pull	Optional field that allows the I3C open drain to be long low and short high. The high period of SCL is the PPBAUD period. This period leaves enough time for the pullup resistor to pull the SDA high when SCL is low. It is also quick when SCL is high and there are no changes happening. ODBAUD low half-clock time period = (ODBAUD + 1) × PPBAUD high period.

Table continues on the next page...

Table 677. Configuration for module initialization (continued)

Configuration parameter		Description
		ODBAUD high half-clock time period = PPBAUD high period If ODHPP is 1, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK ÷ (1 + PPBAUD) × (ODBAUD + 1) + (1 + PBAUD) For example: • If PPBAUD produces 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 and ODHPP = 1. These settings provide a high period of 40 ns and a low period of 200 ns. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4, then open-drain SCL = 50 ÷ (1 + 1) × 5 + (1 + 1) = 4.16 MHz.
I2CBAUD	I2C Baud	Indicates the number of ODBAUD periods that are required to communicate with I2C devices (MCTRL[TYPE] = 2 or MWMSG_SDR_CONTROL[I2C] = 1). For example, if ODBAUD gives 200 ns, and the goal is Fm+ (Fast Mode, 1 MHz), then the sum must be 1 µs. I2CBAUD acts differently for odd and even values. For example: • If I2CBAUD = 3, it gives three ODBAUD periods low and two ODBAUD periods high. • If I2CBAUD = 4, it gives three ODBAUD periods for low and three ODBAUD periods for high. • Also, if I2CBAUD = 3, this yields 200 × 3 = 600 ns and 200 × 2 = 400 ns, with the sum 600 + 400 = 1000 ns = 1 µs.
SKEW	Skew	The normal SDA skew from SCL is handled by using the time for the SCL to reach its pad and return. This time is normally 2 ns to 5 ns (or sometimes more). If the skew is too fast, then add more delay. The skew allows specifying the number of FCLK to insert.

Additional optimizations:

- Use MIBIRULES[MSB0] to obtain faster start header times. If the controller application assigns all I3C dynamic addresses to be less than 40h (it does not have MSB set), you can write 1 to MIBIRULES[MSB0]. When the controller emits 7Eh (broadcast) and a target does not drive the first bit low, the rest of the header can be at push-pull speeds. This speed is two times faster or more, depending on optimizations.
- Auto-emit 7Eh speeds up the frame when used with MIBIRULES[MSB0]. It allows the processor to sleep when the frame starts automatically (in response to a target).

69.5.2 Interrupt service flow

Set up interrupts for Controller Status (MSTATUS) in Controller Interrupt Set (MINTSET).

Set up interrupts for Target Status (SSTATUS) in Target Interrupt Set (SINTSET).

69.6 Application information

69.6.1 I2C configuration to meet timing requirements for Fm and Fm+ modes

I2C Fm and I2C Fm+ modes are supported according to the I3C standard specifications.

Table 678 shows the configuration for Fm mode, where frequency is close to 400 kHz but timing requirement is not met (T_{SU_STA} is not met) because the actual value is 600 ns. Rest of the requirements are met per the I2C specifications in the chip data sheet.

Table 679 shows the configuration for Fm mode, where all timing requirements are met.

Table 678. Configuration for Fm mode with not all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	4	11	370 kHz	T_{SU_STA} - 353 ns

Table 679. Configuration for Fm mode with all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	1	2	9	363 kHz	All requirements met per the I2C specifications in the chip data sheet

Table 680 shows the configuration for Fm+ mode, where the frequency is close to 1 MHz. In this case, the T_{SU_STA} timing requirement is not met because the actual value is 260 ns. Rest of the requirements are met per the I2C specifications in the chip data sheet.

Table 681 shows the configuration for Fm+ mode, where all timing requirements are met. To meet all the timing specifications, frequency may differ from 1 MHz. Use the configurations listed in Table 681.

Table 680. Configuration for Fm+ mode with not all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	4	3	960 kHz	T_{SU_STA} - 228 ns

Table 681. Configuration for Fm+ mode with all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	3	6	749 kHz	All requirements met per the I2C specifications in the chip data sheet

NOTE

The controller module provides an option to slow down the clock for standard speed mode to support legacy I2C standard targets as well (although this option may not necessarily meet the exact timing requirement of the standard mode according to the standard I2C timing specification).

For the timing requirements under consideration, see the "I3C timing requirements when communicating with I2C legacy devices" table in the MIPI I3C v1.1.1 specification available on www.mipi.org.

69.7 I3C register descriptions

Writing to a read-only (RO) register, except to Controller Interrupt Mask (MINTMASKED), causes bus errors. This module does not verify whether programmed values in the registers are correct. You must ensure that valid programmed values are written.

However, the peripheral ignores writes to RO registers and returns 0 for reads from WO or nonexistent registers. It requires all 32-bit registers to be read and written as 32-bit and aligned to 32 bits. The peripheral uses the advanced peripheral bus (APB: AMBA 3 APB Protocol Specification v1.0); therefore, the peripheral does not know whether partial read or write operations (less than 32 bits) are used.

69.7.1 I3C memory map

I3C0 base address: 4003_7000h

I3C1 base address: 4003_8000h

I3C2 base address: 4005_6000h

I3C3 base address: 4005_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Controller Configuration (MCONFIG)	32	RW	0000_0000h
4h	Target Configuration (SCONFIG)	32	RW	0017_0000h
8h	Target Status (SSTATUS)	32	RW	0000_1400h
Ch	Target Control (SCTRL)	32	RW	0000_0000h
10h	Target Interrupt Set (SINTSET)	32	RW	0000_0000h
14h	Target Interrupt Clear (SINTCLR)	32	RW	See section
18h	Target Interrupt Mask (SINTMASKED)	32	R	0000_0000h
1Ch	Target Errors and Warnings (SERRWARN)	32	RW	0000_0000h
20h	Target DMA Control (SDMACTRL)	32	RW	0000_0010h
2Ch	Target Data Control (SDATACTRL)	32	RW	8000_0030h
30h	Target Write Data Byte (SWDATAB)	32	W	See section
34h	Target Write Data Byte End (SWDATABE)	32	W	See section
38h	Target Write Data Halfword (SWDATAH)	32	W	See section
3Ch	Target Write Data Halfword End (SWDATAHE)	32	W	See section
40h	Target Read Data Byte (SRDATAB)	32	R	0000_0000h
48h	Target Read Data Halfword (SRDATAH)	32	R	See section
54h	Target Write Data Byte (SWDATAB1)	32	W	0000_0000h
54h	Target Write Data Halfword (SWDATAH1)	32	W	0000_0000h
5Ch	Target Capabilities 2 (SCAPABILITIES2)	32	R	0000_0300h
60h	Target Capabilities (SCAPABILITIES)	32	R	E83F_FE70h
64h	Target Dynamic Address (SDYNADDR)	32	RW	0000_0000h
68h	Target Maximum Limits (SMAXLIMITS)	32	RW	0000_0000h
6Ch	Target ID Part Number (SIDPARTNO)	32	RW	3000_0000h
70h	Target ID Extension (SIDEXT)	32	RW	0066_EF00h
74h	Target Vendor ID (SVENDORID)	32	RW	0000_011Bh
78h	Target Time Control Clock (STCCLOCK)	32	RW	0000_3014h
7Ch	Target Message Map Address (SMSGMAPADDR)	32	R	0000_0000h
80h	Controller Extended Configuration (MCONFIG_EXT)	32	RW	0000_0000h
84h	Controller Control (MCTRL)	32	RW	0000_0000h
88h	Controller Status (MSTATUS)	32	RW	0000_1000h
8Ch	Controller In-band Interrupt Registry and Rules (MIBIRULES)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
90h	Controller Interrupt Set (MINTSET)	32	RW	0000_0000h
94h	Controller Interrupt Clear (MINTCLR)	32	RW	See section
98h	Controller Interrupt Mask (MINTMASKED)	32	R	0000_0000h
9Ch	Controller Errors and Warnings (MERRWARN)	32	RW	0000_0000h
A0h	Controller DMA Control (MDMACTRL)	32	RW	0000_0010h
ACh	Controller Data Control (MDATACTRL)	32	RW	8000_0030h
B0h	Controller Write Data Byte (MWDATAB)	32	W	See section
B4h	Controller Write Data Byte End (MWDATABE)	32	W	See section
B8h	Controller Write Data Halfword (MWDATAH)	32	W	See section
BCh	Controller Write Data Halfword End (MWDATAHE)	32	W	See section
C0h	Controller Read Data Byte (MRDATAB)	32	R	0000_0000h
C8h	Controller Read Data Halfword (MRDATAH)	32	R	0000_0000h
CCh	Controller Write Byte Data 1 (to Bus) (MWDATAB1)	32	W	0000_0000h
CCh	Controller Write Halfword Data (to Bus) (MWDATAH1)	32	W	0000_0000h
D0h	Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL)	32	W	See section
D0h	Controller Write Message Data in SDR mode (MWMSG_SDR_DATA)	32	W	0000_0000h
D4h	Controller Read Message in SDR mode (MRMSG_SDR)	32	R	0000_0000h
D8h	Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL)	32	W	See section
D8h	Controller Write Message in DDR Mode Control 2 (MWMSG_DDR_CONTROL2)	32	W	See section
D8h	Controller Write Message Data in DDR mode (MWMSG_DDR_DATA)	32	W	0000_0000h
DCh	Controller Read Message in DDR mode (MRMSG_DDR)	32	R	0000_0000h
E4h	Controller Dynamic Address (MDYNADDR)	32	RW	0000_0000h
11Ch	Map Feature Control 0 (SMAPCTRL0)	32	R	0000_0000h
140h	Extended IBI Data 1 (IBIEXT1)	32	RW	0000_0070h
144h	Extended IBI Data 2 (IBIEXT2)	32	RW	0000_0000h
FFCh	Target Module ID (SID)	32	R	EDCB_0100h

69.7.2 Controller Configuration (MCONFIG)

Offset

Register	Offset
MCONFIG	0h

Function

Controls all controller states when the controller operation is enabled. You must not change this register during an active transaction.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	I2CBAUD				SKEW			ODHPP	ODBAUD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPLOW				PPBAUD				Reserved	ODSTOP	HKEEP		DISTO	0	MSTENA	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 I2CBAUD	I2C Baud Rate Specifies the I2C low and high times in ODBAUD counts. See I2CBAUD in Controller configuration.
27-25 SKEW	Skew Specifies the number of FCLK counts for an SDA change after SCL for I3C push-pull. This skew is in addition to the roundtrip of the SCL line from the pad back to the module (6 ns or more): • SKEW is normally not needed, so assign SKEW = 0. • SKEW is only used if SDA is not naturally skewing from an SCL change. • I2C automatically skews SDA (but not PUR) to match I2C rules. Use the following values: • In I2C Controller mode (MCONFIG[MSTENA] = 11b) — SKEW > 0 when clock stretching feature is not required. — SKEW = 0 when clock stretching feature is required. • In I3C Controller mode, use SKEW = 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 ODHPP	Open-drain High Push-Pull Enables ODHPP. If you write 0 to this field, open-drain SCL high half-clock period is the same as the open-drain low SCL half period. If you write 1 to this field, open-drain high SCL half-lock period is one PPBAUD count for I3C messages. This setting is faster and works for I3C devices. Any legacy I2C devices on the bus do not see the SCL high level at all (less than the spike filter period). For open-drain timing, check upon the first 7Eh broadcast allowing I3C peripherals acting as I2C legacy devices to turn off their I2C 50-ns spike filters. See Sending a CCC to I3C targets for more information. 0b - Disable 1b - Enable
23-16 ODBAUD	Open-drain Baud Rate Specifies the open-drain baud rate. ODBAUD is configured in terms of number of PPBAUD counts - 1. This field must not be 0; this setting is not the same as push-pull. See Controller configuration for more information. The open-drain baud rate is usually 200 ns (see I2CBAUD for I2C counts). When used with MCONFIG[ODHPP], this setting produces 250 ns per clock in I3C. In this case, if MCONFIG[PPBAUD] provides 12 MHz, then one PPBAUD count is half of 12 MHz, or 41.67 ns. To obtain a rate around 200 ns, use a value of 5 - 1 = 4 for ODBAUD.
15-12 PPLOW	Push-Pull Low Acts as an adder for push-pull low to create a duty cycle with a longer low period, with up to 15 more FCLK cycles low than high. PPLOW = 0 produces a 50/50 duty cycle.
11-8 PPBAUD	Push-Pull Baud Rate Specifies the push-pull baud rate. The number of FCLK counts makes each push-pull low and normally high period. PPBAUD = 0 when run at 1/2 input FCLK speed. For example, a 24 MHz FCLK produces a 12 MHz SCL because each FCLK is SCL low or SCL high. MCONFIG[PPLOW] has an effect on the duty cycle. For example, 24 MHz with 50/50 duty cycle is 12 MHz. However, when PPLOW adds three more low beats, the push-pull baud rate becomes 4.8 MHz (from 24 MHz or 5 beats).
7 —	Reserved
6 ODSTOP	Open-drain Stop Enables open-drain stop. If you write 0 to this field, open-drain stop is disabled. ODSTOP must be disabled when sending an HDR exit pattern. If you write 1 to this field, open-drain stop is enabled. STOP condition is emitted at open-drain speeds even for I3C messages. In legacy devices, this feature can ensure that the legacy devices see the STOP condition. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
5-4 HKEEP	High-Keeper Indicates how High-Keeper is supported. If this field is 00b, use pull-up resistor (PUR). No separate pin_HK_SDA or pin_HK_SCL is used. Only pin_PUR_oena is used for SDA. The pin_SCL_oena pin is held high in this mode, SCL clock is push-pull, and pin_SCL_out toggles, which is the actual clock. If this field is 01b, High-Keeper controls are in place; pin_HK_SDA or pin_HK_SCL (High-Keeper) controls are used. SCL may use an HK or that signal (pin_HK_SCL) may only OR into pin_SCL_oena. The pin_HK_SDA pin is used as well for SDA high keeper, along with pin_PUR_oena. Use HKEEP = 10b or 11b for I2C modes. Use HKEEP = 00b or 01b for I3C modes. **NOTE** Your chip may not support any or all High-Keeper methods. You must check the availability of High-Keeper methods. 00b - None 01b - WIRED_IN 10b - PASSIVE_SDA (I2C mode, no clock stretches mode). SDA is open drain, and will be Hi-Z (high-impedance) for Bus Free (IDLE) and hold. The pin pin_HK_SDA is not used in this option, only a combination of SDA_oena and pin_PUR_oena are used. SCL is push-pull. The pin pin_HK_SCL is not used in this option, only pin_SCL_oena is used. Since the controller does not support multi-controller mode, there is no need to have open-drain SCL for I2C. For a single controller application, the controller's SCL output can be a push-pull driver design if there are no devices on the bus which would stretch the clock. 11b - PASSIVE_ON_SDA_SCL. For I2C clock stretching mode, where both SCL and SDA are open-drain. Open-Drain on both SDA and SCL, can Hi-Z (high-impedance) both for Bus Free (IDLE), and can Hi-Z SDA for hold. Does not support multi-controller mode, SCL open-drain is only to support clock stretching.
3 DISTO	Disable Timeout Disables the timeout that produces application errors. If the controller is left in a state other than Stopped for more than 100 µs (because 10 kHz is the slowest allowed I3C speed), the timeout sends a MERRWARN interrupt. To prevent the MERRWARN interrupt during development or testing, write 1 to DISTO to disable the timeout. In systems that support timeouts, timeout is disabled automatically during debug. To obtain a clock configuration with 100 µs timeout, see SCONFIG[BAMATCH]. 0b - Enabled 1b - Disabled
2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
1-0 MSTENA	Controller Enable Indicates whether the controller is enabled and the states it can use. If this field is 0b, the controller is disabled. The module can only use Target mode. If this field is 1b, the controller is enabled. When used upon start-up, this module is the main controller by default. I3C controls the bus unless the controller is handed off. When this happens, MSTENA must become 2 so that it has the capability to become the controller again. Performing the handoff means emitting the GETACCMST CCC command. If the command is accepted, I3C emits a STOP condition and sets this field to 2 (or 0). If this field is 10b, I3C is controller-capable, but is operating as a target now. When used from the start, I3C starts as a target, but is prepared to switch to Controller mode. To switch to this mode, the target emits a controller request (CR) or receives a GETACCMST CCC command and accepts it (to switch on the STOP condition). Legacy I2C—if this field is 11b, I2C Controller mode uses an open-drain clock (to allow clock stretching) and relies on passive pull-up resistors (PURs) on both SCL and SDA. This mode uses only I2CBAUD. START requests can only be made for I2C type, and I3C requests are not accepted. If there are only I2C targets on the bus and no clock stretching, use type 1 instead. Type 1 has a push-pull SCL clock, which is faster and uses less power. 00b - CONTROLLER_OFF 01b - CONTROLLER_ON 10b - CONTROLLER_CAPABLE 11b - I2C_CONTROLLER_MODE

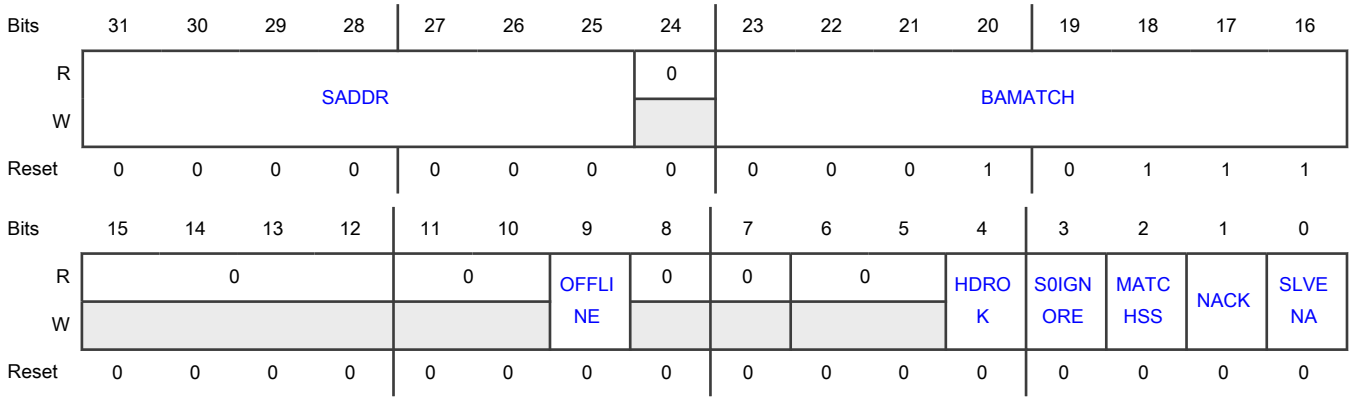
69.7.3 Target Configuration (SCONFIG)

Offset

Register	Offset
SCONFIG	4h

Function
Contains fields that must be configured before the module is activated.

Diagram



Fields

Field	Function
31-25 SADDR	Static Address Sets the I2C 7-bit static address, which otherwise must be 0.
24 —	Reserved
23-16 BAMATCH	Bus Available Match Specifies the bus available condition match value for the current slow clock. BAMATCH provides the count of the slow clock to count out 1 µs (or more). The BAMATCH should be set as (slowClk(MHz) - 1) to generate a 1 µs clock cycle. Use BAMATCH = 1 to generate 1 µs clock cycle if slow clock is 1 MHz. A value of 0 would not give a correct match indication. The default value applies to the default slow clock setting of the chip. If the I3C slow clock is updated to a new value, BAMATCH must be updated, too. When I3C operates in Controller mode, BAMATCH is used to generate a 1 µs clock cycle basic time to count 100 µs timeout. When I3C operates in Target mode, BAMATCH allows the IBI to drive SDA low when the controller is not doing so.
15-12 —	Reserved
11-10 —	Reserved
9 OFFLINE	Offline Enables wait to ensure that the bus is not in HDR mode.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If this field is 1 when SCONFIG[SLVENA] is 1, then I3C waits for either 60 µs of bus quiet or an HDR exit pattern. This waiting ensures that the bus is not in HDR mode, and so can safely monitor the next activity in Single Data Rate (SDR) mode. 0b - Disable 1b - Enable
8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 HDROK	HDR OK Enables HDR OK. If you write 1 to this field, it allows HDR-DDR and/or HDR-BT messaging, if available, by writing 1 to the corresponding SIDEXT[BCR] field to indicate HDR is available, and the corresponding GETCAPS field for DDR and/or BT bit permitting use. This is a deprecated field; use the SIDEXT[BCR] field instead. 0b - Disable HDR OK 1b - Enable HDR OK
3 S0IGNORE	Ignore TE0 or TE1 Errors Ignores TE0 or TE1 errors. If you write 1 to this field, the target does not detect TE0 or TE1 errors, so it does not lock up waiting on an exit pattern. You must not use this setting when the bus does not use HDR mode. 0b - Do not ignore TE0 or TE1 errors 1b - Ignore TE0 or TE1 errors
2 MATCHSS	Match Start or Stop Enables matched START, that is Repeated START, or STOP condition. If you write 1 to this field, SINTSET[START] and SINTSET[STOP] become 1 only when SSTATUS[MATCHED] is 1. This setting allows the START and STOP fields to be used to detect the end of a message to or from this target. **NOTE** SSTATUS[START] does not assert for START of transaction if SCONFIG[MATCHSS] = 1. SSTATUS[START] only asserts for Repeated Start within that transaction, because SSTATUS[START] only asserts after SSTATUS[MATCHED] occurs, which is after START of transaction. 0b - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enable
1 NACK	Not Acknowledge Controls the ACK or NACK capability. If you write 1 to this field, the target rejects all requests, except for a CCC broadcast. NACK = 1 must be used with caution because the controller may decide that the target is missing, if NACK is overused. 0b - Always disable NACK mode 1b - Always enable NACK mode (works normally)
0 SLVENA	Target Enable Enables the target. If you write 0 to this field, the target ignores the I2C or I3C bus. If you write 1 to this field, the target can operate on the I2C or I3C bus. You must not write 1 to this field before registers such as Target Configuration (SCONFIG), Target ID Part Number (SIDPARTNO), Target ID Extension (SIDEXT), and others are configured because these registers affect the data to and from the controller. Target enable is configured just once before the bus comes up. If target enable is used at other times, SCAPABILITIES[IBI_MR_HJ] must be 1 before writing 1 to SLVENA so that the device does not see a START or STOP condition incorrectly. 0b - Disable 1b - Enable

69.7.4 Target Status (SSTATUS)

Offset

Register	Offset
SSTATUS	8h

Function

Indicates the sticky status for interrupts, states, and modes related to the I3C bus. Not all fields of the register are used if the module only acts as a target. The fields are divided into current activity, interrupt maskable actions, and then states and modes on the bus.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIMECTRL		ACTSTATE		HJDIS	0	MRDIS	IBIDIS	0		EVDET		0	EVENT	CHANDLED	HDRMATCH
W														W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	CCC	DACHG	TXNOTFU...	RX_PEND	STOP	MATCHED	START	0	STHDR	STDAA	STREQWR	STREQRD	STCCCH	STMSG	STNOTST...
W		W1C	W1C			W1C	W1C	W1C								
Reset	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 TIMECTRL	Time Control Indicates whether time control is enabled, and in which mode. 00b - NO_TIME_CONTROL (no time control is enabled) 01b - SYNC_MODE (Synchronous mode is enabled) 10b - ASYNC_MODE (Asynchronous standard mode (0 or 1) is enabled) 11b - BOTHSYNCASYNC (both Synchronous and Asynchronous modes are enabled)
29-28 ACTSTATE	Activity State from Common Command Codes (CCC) Indicates the activity state from CCC. 00b - NO_LATENCY (normal bus operations) 01b - LATENCY_1MS (1 ms of latency) 10b - LATENCY_100MS (100 ms of latency) 11b - LATENCY_10S (10 seconds of latency)
27 HJDIS	Hot-Join Disabled Indicates whether hot-join is disabled. When hot-join is disabled, CTRL requests are not responded to. 0b - Enabled 1b - Disabled
26 —	Reserved
25 MRDIS	Controller Requests Disable Indicates whether controller requests are disabled. When controller requests are disabled, CTRL requests are not responded to.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Enabled 1b - Disabled
24 IBIDIS	In-Band Interrupts Disable Indicates whether in-band interrupts are disabled. When in-band interrupts are disabled, CTRL requests are not responded to. 0b - Enabled 1b - Disabled
23-22 —	Reserved
21-20 EVDET	Event Details Indicates current details of the last (EVENT = 1) or pending event. 00b - NONE (no event or no pending event) 01b - NO_REQUEST (request is not sent yet; either there is no START condition yet, or is waiting for Bus-Available or Bus-Idle (HJ)) 10b - NACKed (not acknowledged, request sent and rejected); I3C tries again 11b - ACKed (acknowledged; request sent and accepted), so done (unless the time control data is still being sent)
19 —	Reserved
18 EVENT	Event Flag Indicates, for a target, whether a pending in-band interrupt (IBI), controller request (CR), or hot-join (HJ) has been sent as requested. See the upper status register fields for details. This field becomes 1 only when acknowledged by a controller. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No event occurred 1b - IBI, CR, or HJ occurred When writing 0b - No effect 1b - Clear the flag
17 CHANDLED	Common Command Code Handled Flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
	Indicates whether an HDRMATCH CCC is being handled by I3C. This field is for notification only, but it may result in updates to this register. **NOTE** This field behaves differently for register reads and writes. When reading 0b - CCC handling not in progress 1b - CCC handling in progress When writing 0b - No effect 1b - Clear the flag
16 HDRMATCH	High Data Rate Command Match Flag Indicates whether the HDR command matched the I3C dynamic address of this device. The HDR command is available as the first byte, with RXPEND = 1. The MSB of the command byte indicates whether it is a read or a write command. If the HDR command is a read, and there are to-bus bytes waiting, then the command is ACKed and the data is sent back. Otherwise, the HDR command is NACKed. **NOTE** When this field is 1, you must check SSTATUS[ERRWARN] because the HPAR error may occur after signaling this HDR command. The parity occurs after the destination address and command. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Did not match 1b - Matched the I3C dynamic address When writing 0b - No effect 1b - Clear the flag
15 ERRWARN	Error Warning Indicates that an error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR CRC error, or other error or warning condition (see Target Errors and Warnings (SERRWARN) for more information).
14 CCC	Common Command Code Flag Indicates whether a CCC has been received, and is not handled by I3C. There are two types of common command codes: • Broadcast CCC, which corresponds with RXPEND, and the first byte is the CCC (command).

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• Direct CCC, which may never be directed to this device. If the direct CCC is directed to this device, then TXSEND or RXPEND are triggered, and RXPEND contains the command. **NOTE** This field behaves differently for register reads and writes. When reading 0b - CCC not received 1b - CCC received When writing 0b - No effect 1b - Clear the flag
13 DACHG	Dynamic Address Change Flag Indicates an occurrence of dynamic address (DA) change. Actual DA can be seen in the DYNADDR register. If this field is 1, DA change is detected. The target DA has been assigned, reassigned, or reset (lost) and is now in the state of being valid or none. This field is also used when the MAP auto feature is configured, changing one or more MAP items. See DYNADDR and MAPCTRL*n* for more information. DYNAADDR for the main DA (0) indicates whether the last change was because of Auto-MAP. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No DA change detected 1b - DA change detected When writing 0b - No effect 1b - Clear the flag
12 TXNOTFULL	Transmit Buffer Not Full Indicates that the transmit buffer is not full. If DMA is enabled for transmitting, then it is also signaled to provide more data. To-bus buffer or FIFO can accept more data to be transmitted. For all but external FIFOs, this process uses SDATACTRL[TXTRIG], which defaults to "not full". 0b - Transmit buffer full 1b - Transmit buffer not full
11 RX_PEND	Received Message Pending Indicates whether a received message is pending. The field indicates when a message from the controller that is not being handled by I3C (not a CCC message) is received. I3C processes such messages internally. For all but external FIFOs, this process uses SDATACTRL[RXTRIG], which

Table continues on the next page...

Table continued from the previous page...

Field	Function
	defaults to "not empty". If DMA is enabled for receiving, then DMA is signaled as well. This field automatically becomes 0 if data is read (from FIFO and non-FIFO sources). 0b - No received message pending 1b - Received message pending
10 STOP	Stop Flag Indicates whether the Stopped state is detected. The STNOTSTOP state also indicates when the module is in Stop mode. A fast STOP and START combination may not trigger the Stop status. In that case, the Start status is always set. If this field is 1, it indicates that the Stop state was present on the bus since the bus was last cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stopped state detected 1b - Stopped state detected When writing 0b - No effect 1b - Clear the flag
9 MATCHED	Matched Flag Indicates whether an incoming header matched the I3C dynamic or I2C static address (if any) of this device since the bus was last cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Header not matched 1b - Header matched When writing 0b - No effect 1b - Clear the flag
8 START	Start Flag Indicates whether a START or Repeated START was detected after this flag was last cleared. This flag is not usually needed, but can be used for wake events.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
7 —	Reserved
6 STHDR	Status High Data Rate Indicates whether the I3C bus is in HDR-DDR mode, regardless of whether HDR mode is supported by this module, and regardless of whether the message is intended for this module or some other module. 0b - I3C bus not in HDR-DDR mode 1b - I3C bus in HDR-DDR mode
5 STDAA	Status Dynamic Address Assignment Indicates whether the I3C bus is in Enter Dynamic Address Assignment (ENTDAA) mode, regardless of whether this bus target has a dynamic address. 0b - Not in ENTDAA mode 1b - In ENTDAA mode
4 STREQWR	Status Request Write Indicates whether the REQ in process is SDR write data from the controller to this bus target (or all targets), but not in ENTDAA mode. See status high data rate (STHDR) for double data rate (DDR) handling. 0b - Not an SDR write 1b - SDR write data from the controller, but not in ENTDAA mode
3 STREQRD	Status Request Read Indicates whether the REQ in process is an SDR read from this target. See status high data rate (STHDR) for double data rate (DDR) handling. 0b - Not an SDR read 1b - SDR read from this target or an IBI is being pushed out
2	Status Common Command Code Handler

Table continues on the next page...

Table continued from the previous page...

Field	Function
STCCCH	Indicates whether a CCC message is being handled automatically. 0b - No CCC message handled 1b - Handled automatically
1 STMSG	Status Message Indicates whether the bus target is busy (listening to the bus traffic or responding). If STNOSTOP = 1, STMSG is 0 when a nonmatching address is seen, until the next Repeated START or STOP condition occurs. 0b - Idle 1b - Busy
0 STNOTSTOP	Status not Stop Indicates the status of the bus. If this field is 0, I3C is in a STOP condition. If this field is 1, the bus is busy (has activity). Other fields of this register may also be set when busy. STNOTSTOP can also become 1 after a TE0 or TE1 error, when I3C is waiting for an exit pattern. 0b - In STOP condition 1b - Busy

69.7.5 Target Control (SCTRL)

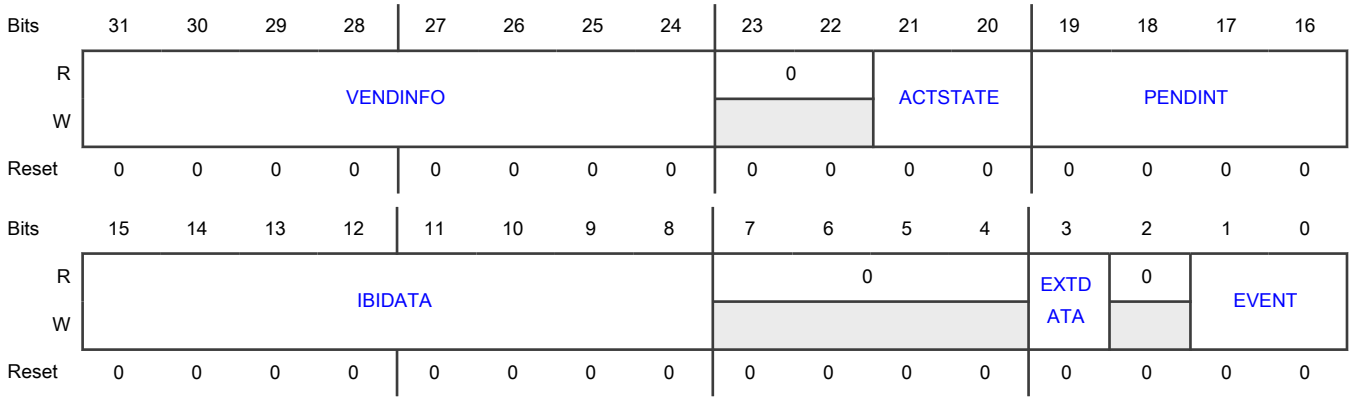
Offset

Register	Offset
SCTRL	Ch

Function

Contains controls for the active use of the I3C bus (for example, event generation such as interrupts to the controller). Only if I3C is configured to support various special operations for the target, this register is used to activate those operations. These events include IBI and GETSTATUS fields (except the protocol error, which is automatically set).

Diagram



Fields

Field	Function
31-24 VENDINFO	Vendor Information Controls vendor information that the GETSTATUS CCC returns. You must set this field to the Vendor Reserved field that the GETSTATUS CCC returns. The application must maintain the vendor information because the controller reads this field. If this field is not configured, then the GETSTATUS field always returns 0.
23-22 —	Reserved
21-20 ACTSTATE	Activity State of Target Controls the activity state of the target. You must set this field to the activity state of the target that the GETSTATUS CCC command returns as activity mode. The application must maintain the activity state because the controller reads this field. If you do not configure the activity state, then the GETSTATUS command always returns 0.
19-16 PENDINT	Pending Interrupt Specifies whether an IBI interrupt is pending. You must set this field to the pending interrupt that the GETSTATUS CCC command returns. The application must maintain the pending interrupt because the controller reads this field. If PENDINT = 0: • The GETSTATUS command returns 1 if an IBI interrupt is pending. • The GETSTATUS field returns 0 if an IBI interrupt is not pending.
15-8 IBIDATA	In-Band Interrupt Data Controls the data byte accompanying the IBI, if the module is enabled for IBI. If SCTRL[IBIDATA] is enabled, then IBI is required.
7-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
3 EXTDATA	Extended Data Enables extended data. After IBIDATA is emitted, extended data is acquired from IBIEXT1 and IBIEXT2, if configured. If extended data is used with time control, the data follows the time information. See IBIEXT1[MAX] for more information. 0b - Disable 1b - Enable
2 —	Reserved
1-0 EVENT	Event Requests an event. If this field is 0b, it indicates Normal mode, in which if this field becomes 0 from a nonzero value and event processing has not yet started, the event processing is canceled. However, event processing is not canceled if it has already started. If this field is 1b, an IBI is pushed onto the I3C bus. If there is data associated with the IBI, the data is read from SCTRL[IBIDATA]. If time control is enabled, this data includes any time-control-related bytes. Additionally, SCTRL[IBIDATA][7] becomes 1 automatically (as is required for time control). The IBI interrupt occurs after the first (mandatory) IBIDATA, if any. If this field is 10b, a controller request is started; the meaning depends on SIDEXT[BCR] configured in I3C. If this field is 11b, a hot-join (HJ) request is started, which is used when the device is powered on after the I3C bus is already powered up. It is also used when the device is connected using hot-insertion methods (the device is powered up when it is physically inserted in the powered-up I3C bus). The HJ waits for Bus Idle, and SCTRL[EVENT] = HOT_JOIN_REQUEST must be set before the target enable (SCONFIG[SLVENA]). If this field is a nonzero value, it requests an event. After the request, SSTATUS[EVENT] and SSTATUS[EVDET] show the status as it progresses. After completion, this field automatically returns to 0. After this field becomes a nonzero value, you can write only 0 it (to cancel) until the event processing is finished. 00b - NORMAL_MODE 01b - IBI 10b - CONTROLLER_REQUEST 11b - HOT_JOIN_REQUEST

69.7.6 Target Interrupt Set (SINTSET)

Offset

Register	Offset
SINTSET	10h

Function

Sets interrupt enables for select Target Status (SSTATUS) fields. Reading this register (SINTSET) returns the status of the interrupt enable:

- To activate an interrupt enable, write 1 to its corresponding field in this register (SINTSET).
- To disable an interrupt, write 1 to its corresponding field in Target Interrupt Clear (SINTCLR). Writing 0 to the interrupt enable in this register (SINTSET) does not disable the interrupt.

The Interrupt registers allow the masking of interrupt sources. They also allow the checking of which interrupts are activated. The normal method is to enable an interrupt, and then after the interrupt occurs, clear the interrupt either by writing to Target Status (SSTATUS) or by performing an action on the corresponding data register. The interrupt is level-held, meaning the interrupt stays set until the cause is cleared by some method. The module prevents races; if a new event occurs, that new event is not lost:

- SINTSET sets interrupt enables for Target Status (SSTATUS) fields. Reading the SINTSET register returns the status of the interrupt enables.
- SINTCLR clears interrupt enables for the SSTATUS fields.
- SINTMASKED returns the value of the SSTATUS fields ANDed with their interrupt enables.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	EVENT	CHANDLED	DDRMATC...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	CCC	DACHG	TXSEND	RXPEND	STOP	MATCHED	START	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19 —	Reserved
18 EVENT	Event Interrupt Enable Enables event interrupts. This field indicates, for a target, whether a pending in-band interrupt (IBI), controller request (CR), or hot-join (HJ) has been sent as requested. The field is configured to support events. See SSTATUS[EVDET] for more information.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
17 CHANDLED	Common Command Code (CCC) Interrupt Enable Enables CCC interrupts. This field specifies whether CCC is being handled by I3C. Target Status (SSTATUS) shows new results. You can use this field to track when activity states and masks on events (for example, IBIs) occur. The field is used for CCCs that are enabled. 0b - Disable 1b - Enable
16 DDRMATCHED	Double Data Rate Interrupt Enable Enables DDR match for read or write command. This field indicates when DDR matches for read or write command, and the field is used only if HDR is enabled. 0b - Disable 1b - Enable
15 ERRWARN	Error or Warning Interrupt Enable Enables error or warning interrupts. This field indicates whether an error or warning has occurred, such as data underrun, data overrun, parity error, or HDR-DDR CRC error, or any other error or warning condition. See Target Errors and Warnings (SERRWARN) for cause of error. Only available for errors related to configured features. 0b - Disable 1b - Enable
14 CCC	Common Command Code (CCC) Interrupt Enable Enables CCC interrupts. This field indicates whether a CCC has been received, and is not handled by I3C. For CCCs that the block does not handle, RXPEND also interrupts, and SSTATUS[STREQRD] indicates that it is a CCC sending a read request. 0b - Disable 1b - Enable
13 DACHG	Dynamic Address Change Interrupt Enable Enables dynamic address change interrupts: interrupt on dynamic address defined (SETDASA or ENTDAA) or lost (RSTDAA). See Controller Dynamic Address (MDYNADDR) for more information.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
12 TXSEND	Transmit Interrupt Enable Enables transmit interrupts. This field indicates whether to-bus buffer or FIFO can accept more data to be transmitted. The corresponding interrupt occurs when the controller requests data to read from this target. This interrupt occurs on the first request (header) as well as when it is ready for more data. The corresponding interrupt occurs when the controller requests data to be read from this target. If this interrupt is for a FIFO, it triggers on the transmit emptiness trigger. If this interrupt is for DMA, then it indicates message end (DMA end or termination). 0b - Disable 1b - Enable
11 RXPEND	Receive Interrupt Enable Enables receive interrupts. This field specifies when a message from the controller that is not being handled by the module is received. For example, where data is consumed by hardware directly when it goes into the FIFO (excludes CCCs being handled automatically). If this interrupt is for a FIFO, it indicates a receive fullness trigger. If this interrupt is for DMA, then it indicates message end. 0b - Disable 1b - Enable
10 STOP	Stop Interrupt Enable Enables stop interrupts. This field detects the Stopped state present on the bus since the bus was last cleared. Use SINTSET[START] as the preferred interrupt when needed. This interrupt may not trigger for a quick STOP and START combination because it relates to the state of being stopped. 0b - Disable 1b - Enable
9 MATCHED	Match Interrupt Enable Enables match interrupts. Incoming header matched the I3C dynamic or I2C static address of this device since the bus was last cleared. If configured and if no dynamic address is set, this interrupt is also for matching a header on an I2C static address. See Controller Dynamic Address (MDYNADDR) for related information. 0b - Disable 1b - Enable
8	Start Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
START	Enables start interrupts. A START or Repeated START (such as wakeup) is seen after this field last became 0. See SINTSET[STOP] for related information. 0b - Disable 1b - Enable
7-0 —	Reserved

69.7.7 Target Interrupt Clear (SINTCLR)

Offset

Register	Offset
SINTCLR	14h

Function

Clears interrupt enables for select Target Status (SSTATUS) fields. To clear an interrupt enable, write 1 to the corresponding field in this register (SINTCLR). Writing 0 has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	EVENT	CHANDLED	DDRMATC...
W														W1C	W1C	W1C
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	CCC	DACHG	TXSEND	RXPEND	STOP	MATCHED	START	0							
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
19 —	Reserved
18 EVENT	EVENT Interrupt Enable Clear Flag
17 CHANDLED	CHANDLED Interrupt Enable Clear Flag
16 DDRMATCHED	DDRMATCHED Interrupt Enable Clear Flag
15 ERRWARN	ERRWARN Interrupt Enable Clear Flag
14 CCC	CCC Interrupt Enable Clear Flag
13 DACHG	DACHG Interrupt Enable Clear Flag
12 TXSEND	TXSEND Interrupt Enable Clear Flag
11 RXPEND	RXPEND Interrupt Enable Clear Flag
10 STOP	STOP Interrupt Enable Clear Flag
9 MATCHED	Matched Interrupt Enable Clear Flag
8 START	START Interrupt Enable Clear Flag
7-0 —	Reserved

69.7.8 Target Interrupt Mask (SINTMASKED)

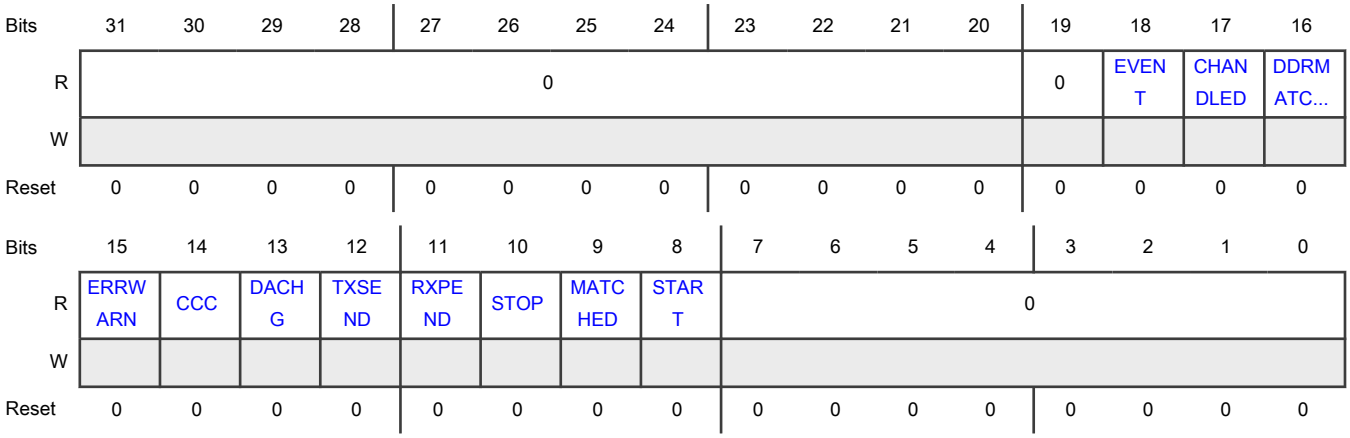
Offset

Register	Offset
SINTMASKED	18h

Function

Returns the status of enabled interrupts (the value of Target Status (SSTATUS) ANDed with the value of Target Interrupt Set (SINTSET)).

Diagram



Fields

Field	Function
31-20 —	Reserved
19 —	Reserved
18 EVENT	EVENT Interrupt Mask
17 CHANDLED	CHANDLED Interrupt Mask
16 DDRMATCHED	DDRMATCHED Interrupt Mask
15	ERRWARN Interrupt Mask

Table continues on the next page...

Table continued from the previous page...

Field	Function
ERRWARN	
14 CCC	CCC Interrupt Mask
13 DACHG	DACHG Interrupt Mask
12 TXSEND	TXSEND Interrupt Mask
11 RXPEND	RXPEND Interrupt Mask
10 STOP	STOP Interrupt Mask
9 MATCHED	MATCHED Interrupt Mask
8 START	START Interrupt Mask
7-0 —	Reserved

69.7.9 Target Errors and Warnings (SERRWARN)

Offset

Register	Offset
SERRWARN	1Ch

Function

Contains errors and warnings from I3C and I2C protocols. This includes internal issues such as overrun and underrun, detected errors and conditions like parity errors, CRC errors, and read terminations by the controller. See SSTATUS[ERRWARN] and SINTSET[ERRWARN] for related information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										0		0		OWRITE	OREAD
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	0	S0S1	HCRC	HPAR	SPAR	0			INVSTART	TERM	URUNNACK	URUN	ORUN
W					W1C	W1C	W1C	W1C				W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-22 —	Reserved
21 —	Reserved
20-18 —	Reserved
17 OWRITE	Over-Write Error Flag Indicates that Target Write Data Byte (SWDATAB) or Target Write Data Byte End (SWDATABE) was written to when full. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overwrite error 1b - Overwrite error When writing 0b - No effect 1b - Clear the flag
16 OREAD	Over-Read Error Flag Indicates that Target Read Data Byte (SRDATAB) was read for more bytes than were available, by the application. This field also indicates over-read errors for Target Read Data Halfword (SRDATAH), if it is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No over-read error 1b - Over-read error When writing 0b - No effect 1b - Clear the flag
15-14 —	Reserved
13 —	Reserved
12 —	Reserved
11 S0S1	TE0 or TE1 Error Flag Indicates whether a TE0 or TE1 error occurred and the target is locked and waiting for an HDR exit pattern. Writing 1 to S0S1 causes I3C to release the lock, but this method must be used with great care. S0S1 becomes 0 automatically when an exit pattern is detected, so writing 1 to S0S1 must be used under controlled circumstances to avoid problems. Before starting to operate normally after this error, I3C waits for a START (or Repeated START) or STOP state. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No TE0 or TE1 error occurred 1b - TE0 or TE1 error occurred When writing 0b - No effect 1b - Clear the flag
10 HCRC	HDR-DDR CRC Error Flag Indicates whether an HDR-DDR CRC error occurred on a message from the controller. Error reasons include an HDR restart and an exit being issued before an HDR-DDR message from the controller has finished. This error calls into question the data from the entire DDR command frame.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No HDR-DDR CRC error occurred 1b - HDR-DDR CRC error occurred When writing 0b - No effect 1b - Clear the flag
9 HPAR	HDR Parity Error Flag Indicates when an HDR parity error or framing error on a message from the controller occurs. The corresponding command or data that has the error is usually in the RX buffer, which can be read using Target Read Data Byte (SRDATAB). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No HDR parity error 1b - HDR parity error When writing 0b - No effect 1b - Clear the flag
8 SPAR	SDR Parity Error Flag Indicates when an SDR parity error on a message from the controller occurs. This error also sets the GETSTATUS protocol error field (which becomes 0 after a GETSTATUS read). For read operations, this field becomes 1 when a read abort (timeout) occurs because of the controller not driving clock for more than 100 µs during an SDR read. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No SDR parity error 1b - SDR parity error When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-5 —	Reserved
4 INVSTART	Invalid Start Error Flag Indicates an invalid condition with SCL falling before SDA falls, so there is no start. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No invalid start error 1b - Invalid start error When writing 0b - No effect 1b - Clear the flag
3 TERM	Terminated Error Flag Indicates when the controller terminates a read from a target, if an END is not set by the target (on the same read or the previous read). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No terminated error 1b - Terminated error When writing 0b - No effect 1b - Clear the flag
2 URUNNACK	Underrun and Not Acknowledged (NACKed) Error Flag Indicates when the internal to-bus buffer or FIFO is underrun in the read header and so the module NACKed the header. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underrun; not acknowledged error 1b - Underrun; not acknowledged error When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
1 URUN	Underrun Error Flag Indicates when the internal to-bus buffer or FIFO is underrun during data read (the application is not providing the data fast enough). The END field or register must be used if the read was the last one. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underrun error 1b - Underrun error When writing 0b - No effect 1b - Clear the flag
0 ORUN	Overrun Error Flag Indicates when the internal from-bus buffer or FIFO has overrun (arrival of too many characters that you cannot process fast enough). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overrun error 1b - Overrun error When writing 0b - No effect 1b - Clear the flag

69.7.10 Target DMA Control (SDMACTRL)

Offset

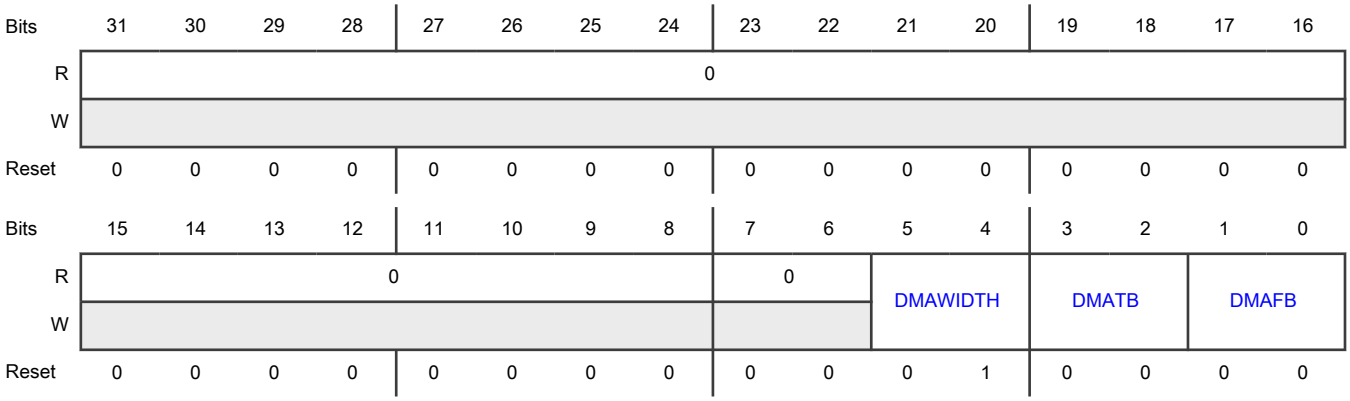
Register	Offset
SDMACTRL	20h

Function

Allows DMA to be used for inbound and outbound messages. This register is limited in value for target use because the target must be reactive. These are the two common use case models:

- To avoid an overrun in from-bus collection: if SCONFIG[MATCHSS] becomes 1, then the processor enables the interrupts for START and STOP and enables the DMA to collect the data. The START or STOP interrupt only occurs after a message is directed to the target (MATCHED is 1). The DMA copied data can then be examined.
- To perform larger reads from the to-bus: I3C and I2C reads are preceded by a write that indicates what will be read (or in response to an IBI from the target). Because of this process, the DMA can be used to push through the data:
  - For I3C, the processor must manage the last value, unless the DMA moves wider words and is able to write 1 to the END field. These values are 16-bit values when in Byte mode or 32-bit values when in Halfword mode.
  - For I2C, the controller determines the last value, so the DMA may end early or run out when the controller expects more values.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-6 —	Reserved
5-4 DMAWIDTH	Width of DMA Operations Indicates the width of DMA operations, if configured to allow halfword data accesses. 00b,01b - Byte 10b - Halfword (16 bits) (this value ensures that two bytes are available in the FIFO) 11b - Reserved
3-2 DMATB	DMA Write (To-Bus) Trigger Enables DMA writes. If this field is enabled, it starts a request DMA on a transmit trigger (see Target Data Control (SDATACTRL)). DMATB requests until full, unless the DMA is set up as a trigger. DMATB becomes 0 when MSTATUS[ERRWARN] becomes 1.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If you write 1b to this field, DMA is enabled for one frame (ended by DMA or terminated). DMATB automatically becomes 0 after a STOP or repeated START. See SCONFIG[MATCHSS] for more information. If you write 10b to this field, DMA is enabled until turned off. The value must only be used with Controller Message mode. 00b - DMA not used 01b - DMA enabled for one frame 10b - DMA enabled until turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11b - Reserved
1-0 DMAFB	DMA Read (From-Bus) Trigger Enables DMA reads. If this field is enabled, DMAFB requests DMA on receive trigger (see SDATACTRL). It requests until empty, unless the DMA is set up as a trigger. This field becomes 0 when SSTATUS[ERRWARN] becomes 1. If this field is 1b (DMA is enabled for one frame), it automatically becomes 0 on STOP or repeated START. See SCONFIG[MATCHSS] for more information. **NOTE** Do not enable DMA after a transaction starts. It must be enabled only when enabling the target and interrupts to avoid any kind of RX FIFO overrun. 00b - DMA not used 01b - DMA enabled for one frame 10b - DMA enabled until turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11b - Reserved

69.7.11 Target Data Control (SDATACTRL)

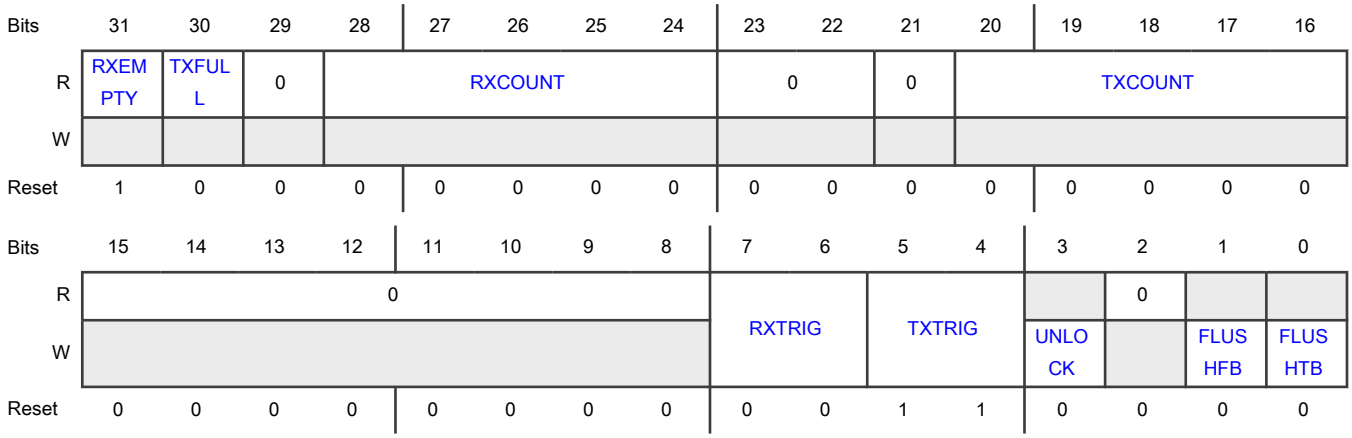
Offset

Register	Offset
SDATACTRL	2Ch

Function

Assists in data control when no FIFO is used. Also assists in data control of FIFO when the FIFO is available (regardless of size) allowing some control over the FIFO behavior. This register controls when to interrupt based on fullness or emptiness of a buffer or FIFO. It also controls behavior related to width, when the buffer or FIFO is not 1 byte wide.

Diagram



Fields

Field	Function
31 RXEMPTY	Receive is Empty Indicates whether the receive FIFO or buffer is empty. 0b - Not empty 1b - Empty
30 TXFULL	Transmit is Full Indicates whether the transmit FIFO or buffer is full. 0b - Not full 1b - Full
29 —	Reserved
28-24 RXCOUNT	Count of Entries in Receive Contains the count of entries in the receive FIFO or buffer. Entry size for SDR and HDR-DDR is in bytes.
23-22 —	Reserved
21 —	Reserved
20-16 TXCOUNT	Count of Entries in Transmit Contains the count of entries waiting in the TXFIFO. This count is the number of entries that the application has written to the transmit FIFO that have not yet gone onto the bus. Entry size for SDR and HDR-DDR is in bytes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-8 —	Reserved
7-6 RXTRIG	Receive Trigger Level Indicates the trigger level for receive fullness when using a FIFO. The field affects the RXPEND interrupt. 00b - Trigger when not empty (default) 01b - Trigger when 1/4 or more full 10b - Trigger when 1/2 or more full 11b - Trigger when 3/4 or more full
5-4 TXTRIG	Transmit Trigger Level Indicates the trigger level for transmit emptiness when using a FIFO. The field affects the TXNOTFULL interrupt. 00b - Trigger when empty 01b - Trigger when 1/4 full or less 10b - Trigger when 1/2 full or less 11b - Default (trigger when 1 less than full or less)
3 UNLOCK	Unlock Indicates whether the RXTRIG and TXTRIG fields can be changed on a write. This field must be 1 in the same cycle while writing to TXTRIG or RXTRIG. 0b - Cannot be changed 1b - Can be changed
2 —	Reserved
1 FLUSHFB	Flush From-Bus Buffer or FIFO Not normally used. **NOTE** Before using FLUSHFB, disable the DMA (SDMACTRL[DMAFB] = 0). Then use FLUSHFB to flush the FIFO and then re-enable the DMA. 0b - No action 1b - Flush the buffer
0 FLUSHTB	Flush To-Bus Buffer or FIFO Indicates when the controller terminates a to-bus (read) message prematurely.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE Before using FLUSHTB, disable the DMA (SDMACTRL[DMATB] = 0). Then use FLUSHTB to flush the FIFO and then re-enable the DMA. 0b - No action 1b - Flush the buffer

69.7.12 Target Write Data Byte (SWDATAB)

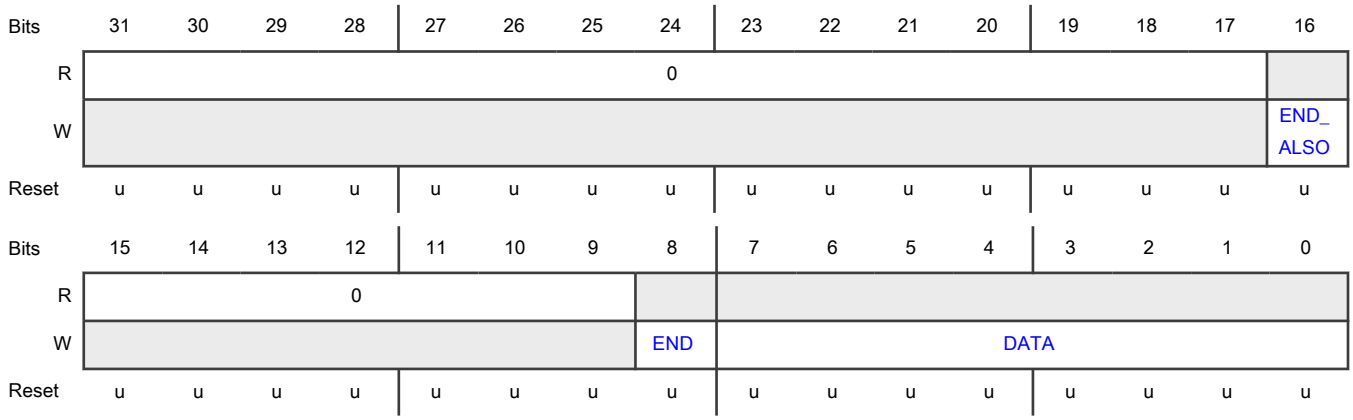
Offset

Register	Offset
SWDATAB	30h

Function

Allows writing a byte to the bus (to the controller) unless an external FIFO is used. Writing a byte requires a byte plus an end-of-data (last) marker bit. A byte must not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 END_ALSO	End Also Marks the last byte of the message. This field is required for I3C, but is optional for I2C. If this field is 0, it indicates that there are more bytes in the message.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	For HDR-DDR, the byte with END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0b - Not the end 1b - End
15-9 —	Reserved
8 END	End Marks the last byte of the message. If this field is 0, it indicates that there are more bytes in the message. This field is required for I3C, but is optional for I2C. For HDR-DDR, the byte with END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0b - Not the end 1b - End
7-0 DATA	Data Indicates the data byte to be sent to the controller.

69.7.13 Target Write Data Byte End (SWDATABE)

Offset

Register	Offset
SWDATABE	34h

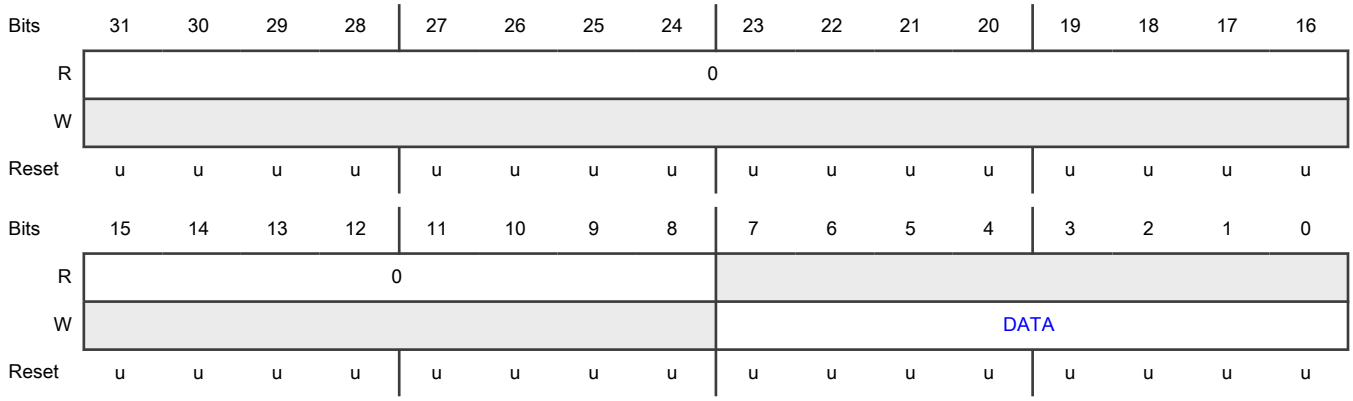
Function

Allows writing a byte to the bus (to the controller) unless an external FIFO is used.

Unlike Target Write Data Byte (SWDATAB), writing a byte only requires the byte itself, and is marked as end-of-data (last byte). A byte must not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.

For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 DATA	Data Indicates the data byte to be sent to the controller.

69.7.14 Target Write Data Halfword (SWDATAH)

Offset

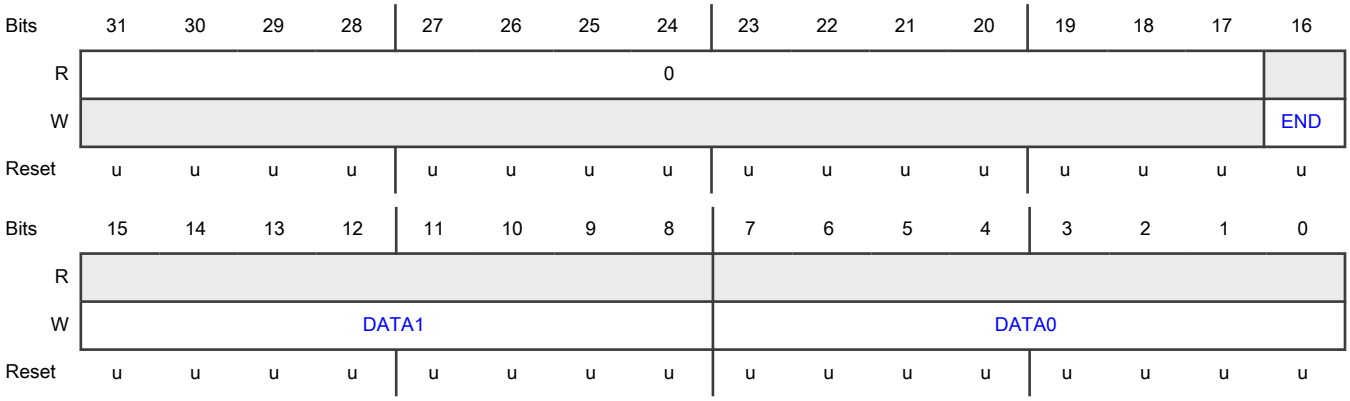
Register	Offset
SWDATAH	38h

Function

Allows writing a halfword (pair of bytes) to the bus unless an external FIFO is used. The low byte is followed by the high byte. The 16th bit marks the end; that is, the last byte of the halfword is the end.

An end-of-data (last) marker bit is allowed (or must be 0). A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or SDATACTRL[TXCOUNT].

Diagram



Fields

Field	Function
31-17 —	Reserved
16 END	End of Message Marks the last byte of the message. This field always marks DATA1 as the end. The field is required for I3C, but is optional for I2C. If this field is 0, it indicates that there are more bytes in the message. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0b - Not the end 1b - End
15-8 DATA1	Data 1 Indicates the second byte to be sent to the controller.
7-0 DATA0	Data 0 Indicates the first byte to be sent to the controller.

69.7.15 Target Write Data Halfword End (SWDATAHE)

Offset

Register	Offset
SWDATAHE	3Ch

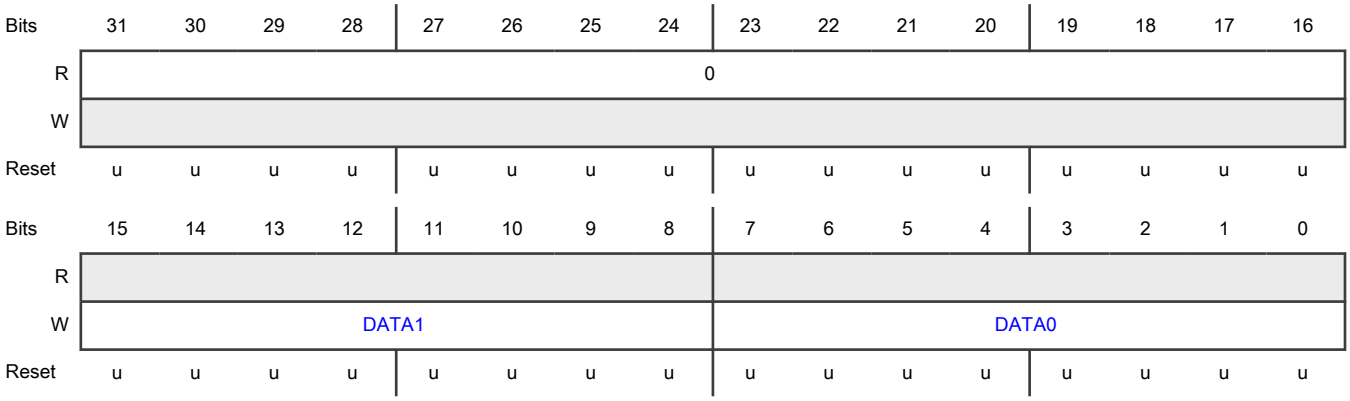
Function

Allows writing a halfword of data, which is the end (the last byte of the halfword is the end). The target writes the halfword (byte pair) in the same way it writes to Target Write Data Halfword (SWDATAH), but marks the second byte as end-of-data (last byte).

For HDR-DDR, the byte with the END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs.

A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or SDATACTRL[TXCOUNT].

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DATA1	Data 1 Indicates the second byte to be sent to the controller.
7-0 DATA0	Data 0 Indicates the first byte to be sent to the controller.

69.7.16 Target Read Data Byte (SRDATAB)

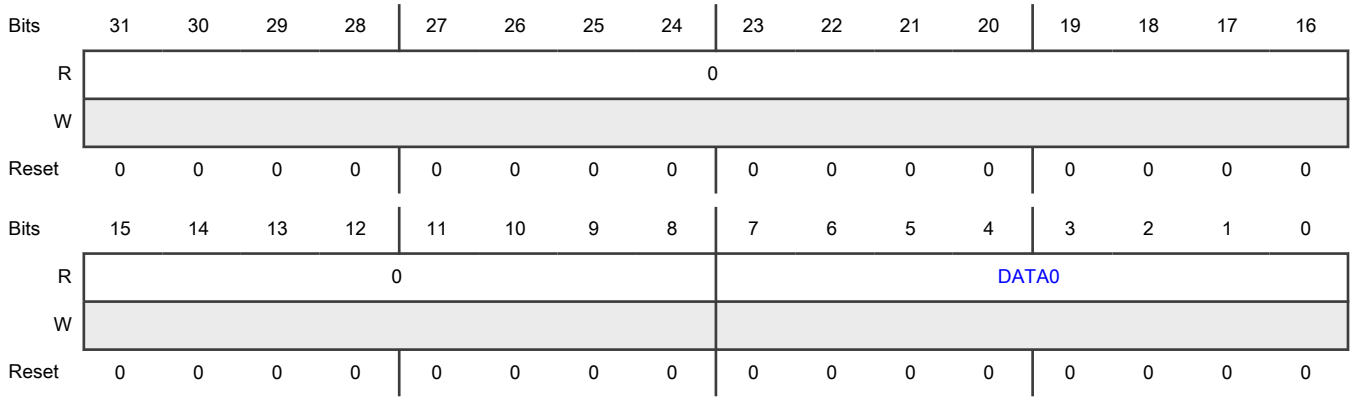
Offset

Register	Offset
SRDATAB	40h

Function

Allows reading a byte from the bus (controller). A byte must not be read unless there is data waiting, as indicated by SSTATUS[RX_PEND] = 1.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 DATA0	Data 0 Indicates the byte read from the controller.

69.7.17 Target Read Data Halfword (SRDATAH)

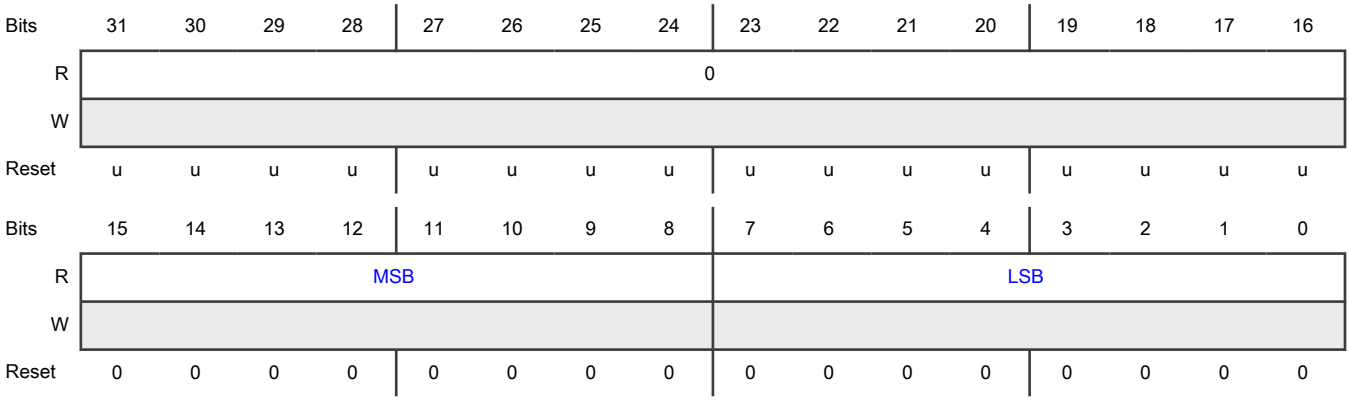
Offset

Register	Offset
SRDATAH	48h

Function

Allows reading a halfword (byte pair) written by the target after an SDR read or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively. A halfword must not be read unless there are at least two bytes of data waiting, as indicated by the use of receive FIFO level trigger or SDATACTRL[RXCOUNT].

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 MSB	High Byte Indicates the second byte read from the target.
7-0 LSB	Low Byte Indicates the first byte read from the target.

69.7.18 Target Write Data Byte (SWDATAB1)

Offset

Register	Offset
SWDATAB1	54h

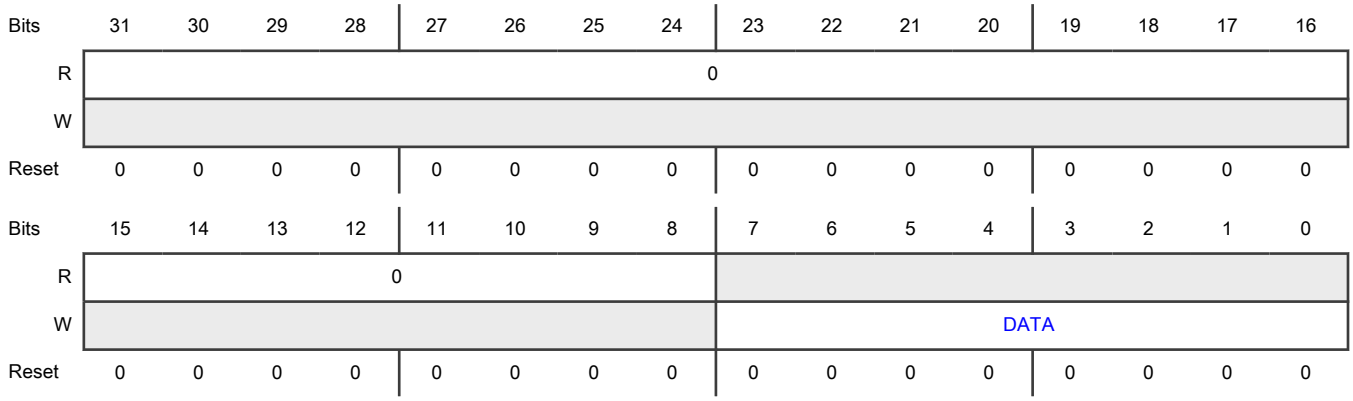
Function

Allows writing a single byte to the bus (to the controller) in a way that only bits 7:0 are used.

This register is intended to be used by DMAs (the upper bytes of the APB word are ignored).

A byte must not be written unless there is room, as indicated by SSTATUS[TXNOTFULL] = 1.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 DATA	Data Indicates the byte to be sent to the controller.

69.7.19 Target Write Data Halfword (SWDATAH1)

Offset

Register	Offset
SWDATAH1	54h

Function

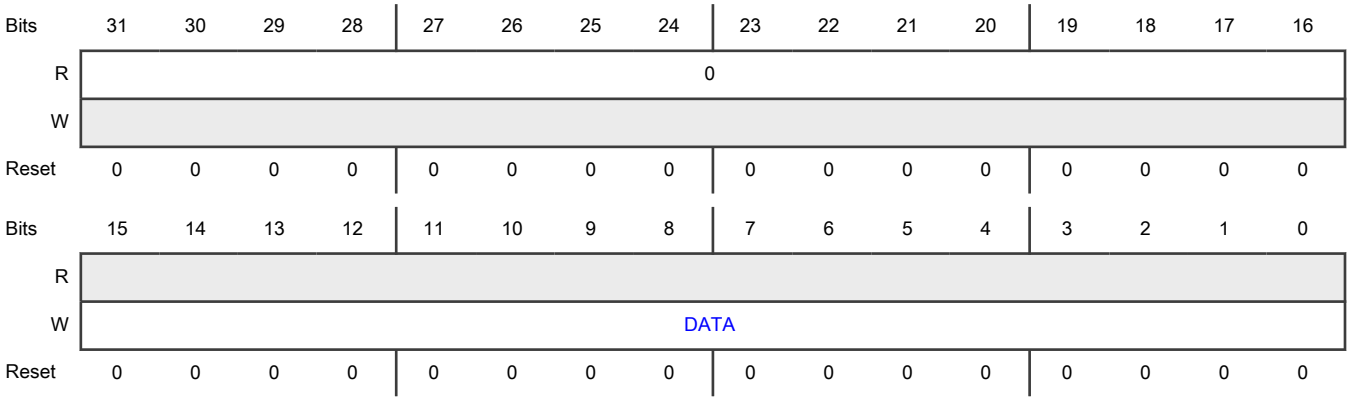
Shares the same address as Target Write Data Byte (SWDATAB1), and SDMACTRL[DMAWIDTH] determines which one is set.

This register allows writing a single-byte pair (halfword) to the bus (to the controller) in a way that only bits 15:0 are used.

This register is intended for use by DMAs, which do not format the upper part of the APB word.

A halfword must not be written unless there is room, as indicated by SSTATUS[TXNOTFULL] = 1.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Data Indicates the byte pair (halfword) to be sent to the controller.

69.7.20 Target Capabilities 2 (SCAPABILITIES2)

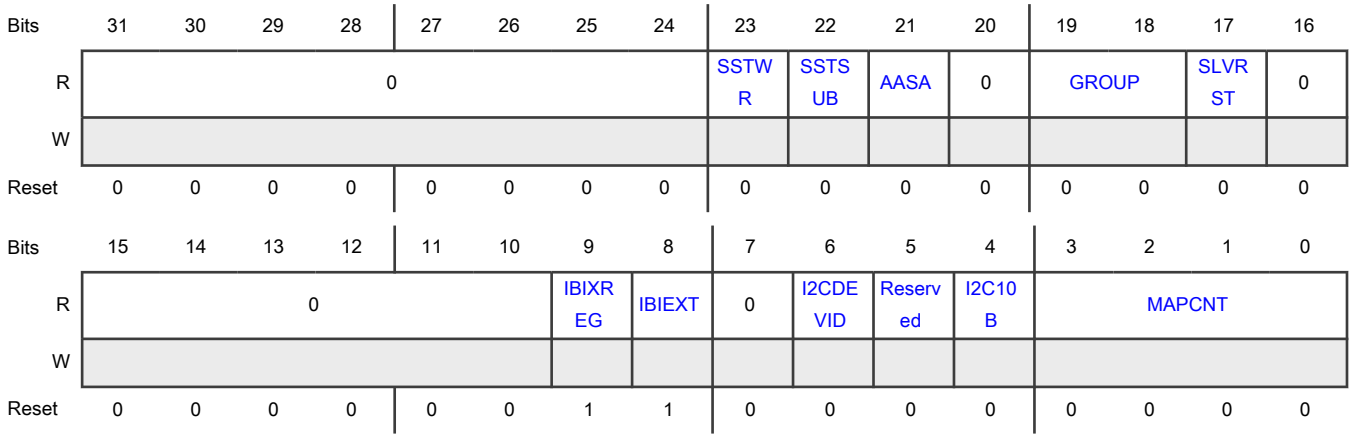
Offset

Register	Offset
SCAPABILITIES2	5Ch

Function

Indicates which features are available and supported in this module, including controller and target capabilities, HDR modes, and others.

Diagram



Fields

Field	Function
31-24 —	Reserved
23 SSTWR	Target-Target(s)-Tunnel Write Capable Indicates whether target-target(s)-tunnel is write capable. 0b - Not write capable 1b - Write capable
22 SSTSUB	Target-Target(s)-Tunnel Subscriber Capable Indicates whether target-target(s)-tunnel is subscriber capable. 0b - Not subscriber capable 1b - Subscriber capable
21 AASA	SETAASA Supports the set static address as dynamic address CCC (SETAASA) feature. 0b - SETAASA not supported 1b - SETAASA supported
20 —	Reserved
19-18 GROUP	Group Indicates whether v1.1 group addressing is supported. Groups use mapping, so MAPCNT must be the same or greater than GROUP. 00b - v1.1 group addressing not supported 01b - One group supported

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10b - Two groups supported 11b - Three groups supported
17 SLVRST	Target Reset Indicates whether v1.1 target reset is supported. 0b - Not supported 1b - Supported
16 —	Reserved
15-10 —	Reserved
9 IBIXREG	In-Band Interrupt Extended Register Supports extended registers for IBIs. This field indicates whether Extended IBI Data 1 (IBIEXT1) is available. Extended IBI Data 2 (IBIEXT2) is available if IBIEXT1[MAX] > 3. 0b - Not supported 1b - Supported
8 IBIEXT	In-Band Interrupt EXTDATA Supports SCTRL[EXTDATA] to allow data beyond the mandatory data byte (MDB) for IBIs. 0b - Not supported 1b - Supported
7 —	Reserved
6 I2CDEVID	I2C Device ID Indicates whether I2C device ID is supported. 0b - Not supported 1b - Supported
5 —	Reserved
4 I2C10B	I2C 10-bit Address Supports 10-bit I2C address in MAP 1. If this field is 1, MAPCNT must be 1 or greater.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not supported 1b - Supported
3-0 MAPCNT	Map Count Indicates the number of maps that are allowed. If there is no mapping, this field is 0. (DYNADDR is used to add more static and/or dynamic addresses to act as virtual targets.)

69.7.21 Target Capabilities (SCAPABILITIES)

Offset

Register	Offset
SCAPABILITIES	60h

Function

Indicates which features are available and supported in this module, including controller and/or target capabilities, HDR modes, and others.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA	INT	FIFORX		FIFOTX		EXTFIFO			0	TIMECTRL		IBI_MR_HJ			
W																
Reset	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CCCHANDLE				SADDR		MASTER	0	HDRSUPP		IDREG				IDENA	
W																
Reset	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0

Fields

Field	Function
31 DMA	Direct Memory Access Indicates whether DMA is supported. 0b - Not supported 1b - Supported

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 INT	Interrupts Indicates whether interrupts are supported. 0b - Not supported 1b - Supported
29-28 FIFORX	FIFO Receive Indicates the size of receive (from-bus) FIFO. **NOTE** Controller and target use the same receive FIFO, so size is the same for the controller receive FIFO as well. FIFO size of SDR and HDR-DDR is in bytes. 00b - Two or three 01b - Four 10b - Eight 11b - 16 or larger
27-26 FIFOTX	FIFO Transmit Indicates the size of transmit (to-bus) FIFO. **NOTE** Controller and target use the same transmit FIFO, so size is the same for the controller transmit FIFO as well. FIFO size of SDR and HDR-DDR is in bytes. 00b - Two 01b - Four 10b - Eight 11b - 16 or larger
25-23 EXTFIFO	External FIFO Indicates whether external FIFOs are enabled. If they are not enabled, then check FIFOTX and FIFORX for the internal FIFO. 000b - No external FIFO available 001b - Standard available or free external FIFO 010b - Request track external FIFO All other values are reserved.
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
21 TIMECTRL	Time Control Specifies whether any time-control type is supported. 0b - No time control supported 1b - At least one time-control type supported
20-16 IBI_MR_HJ	In-Band Interrupts, Controller Requests, Hot-Join Events Indicates which events (IBI, CR, and HJ) are allowed. For example, if this field is 00011b, IBI (bit 0) and IBI_HAS_DATA (bit 1) functionality are both enabled. 0_0000b - Application cannot generate IBI, CR, or HJ 1_xxxxb - Application can use SCONFIG[BAMATCH] for bus-available timing x_1xxxb - Application can generate a Hot-Join event x_x1xxb - Application can generate a controller request for a secondary controller x_xx1xb - When bit 0 = 1, the IBI has data from the SCTRL register x_xxx1b - Application can generate an IBI
15-12 CCCHANDLE	Common Command Codes Handling Indicates what manages CCC between I3C and the application. 0000b - All handling features disabled 1xxxb - GETSTATUS CCC returns the value of SCTRL[VENDINFO] x1xxb - GETSTATUS CCC returns the values of SCTRL[PENDINT] and SCTRL[ACTSTATE] xx1xb - The I3C module manages maximum read and write lengths, and max data speed xxx1b - The I3C module manages events, activities, status, HDR, and if enabled for it, ID and static-address-related items
11-10 SADDR	Static Address Indicates how the static address is managed. 00b - No static address 01b - Static address is fixed in hardware 10b - Hardware controls the static address dynamically (for example, from the pin strap) 11b - SCONFIG register supplies the static address
9 MASTER	Controller Specifies whether controller capability is supported. 0b - Not supported 1b - Supported
8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
7-6 HDRSUPP	High Data Rate Support Indicates which HDR modes are supported. 00b - No HDR modes supported 01b - DDR mode supported All other values are reserved.
5-2 IDREG	ID Register Indicates which ID features are available in the configurable registers. 0000b - All ID register features disabled 1xxxb - A Bus Characteristics Register (BCR) is available x1xxb - A Device Characteristic Register (DCR) is available xx1xb - An ID Random field is available xxx1b - ID Instance is a register; used if there is no PARTNO register
1-0 IDENA	ID 48b Handler Indicates what handles the 48-bit ID value. 00b - Application 01b - Hardware 10b - Hardware, but the I3C module instance handles ID 48b 11b - A part number register (PARTNO)

69.7.22 Target Dynamic Address (SDYNADDR)

Offset

Register	Offset
SDYNADDR	64h

Function

Fills with the assigned address after the controller has assigned it via SETDASA or ENTDAA CCC commands. It clears if the RESETDAA CCC is used. The current validity state is also indicated through SSTATUS[DAVALID], which can be used to interrupt the processor. The DCAUSE field can be used to determine the changes, if enabled.

NOTE

If mapped addresses are enabled, you can access them through the SMAPCTRL*n* registers.

The first dynamic address (DA[0]) may be written irrespective of whether the mapping is enabled, if configured to allow writes used to restore the DA after a power-down (an ultra-low power state that loses power to peripherals but retains the DA somewhere else). This is not required if state-retention flops are used for the DA. This mechanism only allows writes when target is disabled

(and is ignored otherwise). If the controller uses RSTDAA or SETNEWDA, then it overrides this mechanism and yields to the controller-assigned DA. When enabling the target, SCONFIG[OFFLINE] must be 1. This waits for evidence that the bus is not in I3C HDR mode and exits when an HDR exit pattern is seen or when 60 µs has expired. This makes it safe to monitor START and STOP. If the application needs to do an IBI, then the application must either wait for a STOP (see STATUS) or make sure that 200 µs have gone by with no activity (no START or STOP) before the application emits the IBI.

The MAPIDX or MAPSA model allows writing additional DAs (and SAs) into a list (the number in the list is preconfigured); any DA or SA with DAVALID = 0 are never matched. The additional DAs may be based on the bridge target CCC or done by a move (read current DA and then copy into the upper map, and then invalidate the 0 map) when matching ENTDAA over and over. Any mapped location can be invalidated by writing MAPIDX and DAVALID = 0. The mapped ones are not reset by the RSTDA CCC (see Target Message Map Address (SMSGMAPADDR)).

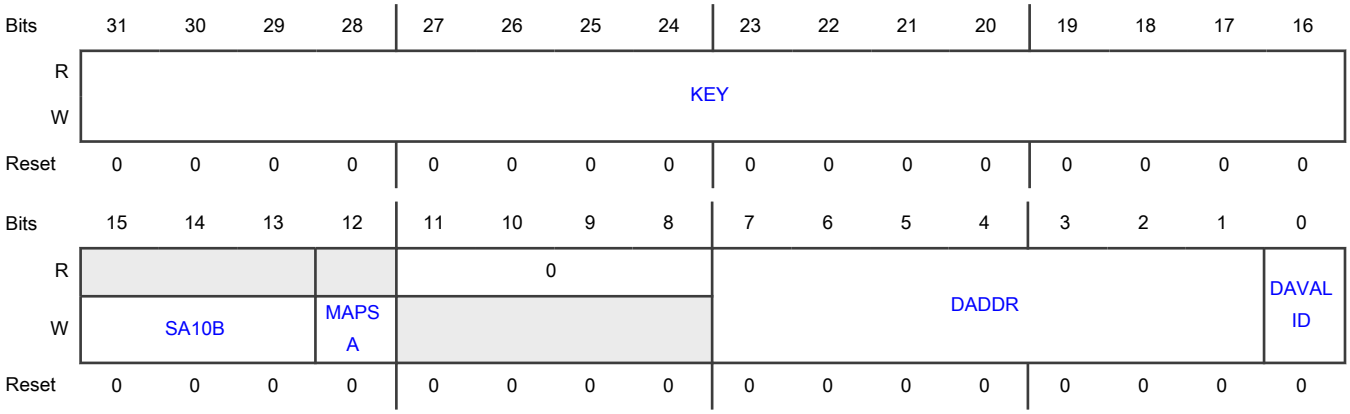
To copy the base DA to the mapped set, you must perform the configuration to allow it; otherwise it leads to distinct mechanisms:

- If used in I3C mode, a base DA is required, so the last one must not be cleared (or writing a dynamic address with DAVALID = 1 is necessary).

NACK/ACK of mapped DAs/SAs: The register also permits specific DAs (above the base) to NACK or ACK writes or reads, such as for bridges when the endpoint is busy. The NACK must not be set for too long, or the controller may consider the target in an error state. The model is to write with KEY = A731h and the MAPIDX and DAVALID to select ACK or NACK.

Mapping: If MAPIDX ≠ 0 and KEY = 0, then the next read returns details on the static or dynamic address at that location. If the read does not have MAPIDX ≠ 0, then it is not an acceptable location.

Diagram



Fields

Field	Function
31-16 KEY	Key Specifies the value of KEY. Must be set to A4D9h to write to the DADDR field (and 1 to the DAVALID field). Write only when a target is not enabled (for restoring after power-down sleep with auto-restore). The mapped locations and base may be written when a target is enabled, but do not write when there are transactions on the I3C bus. KEY reads back 1 if overwritten, else KEY is 0 if assigned by the controller including when the controller changes it. If address mapping is allowed, then writing with KEY = CB19h is used to clear the base DA.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	If mapping is allowed, then writing with the key of A731h means that MAPIDX indicates which mapped address to set to ACK or NACK, such that DAVALID = 1 if ACK, DAVALID = 0 if NACK. This allows having the address refuse write and read requests when NACK. When using read back from map index (write with MAPIDX ≠ 0, KEY = 0, read) bit [16] = 1 if NACK, else 0.
15-13 SA10B	10-Bit Static Address Stores a 10-bit static address composed of {SA10B,DADDR}, if not 0 when MAPSA is set. Only one static address may be stored this way, and it must be MAPIDX == 1.
12 MAPSA	Map a Static Address Sets a static address into the list if MAPSA = 1 on a write with MAPIDX ≠ 0; otherwise, a dynamic address is used.
11-8 —	Reserved
7-1 DADDR	Dynamic Address Specifies the dynamic address (DA). This is the assigned dynamic address when DAVALID is 1 and is a static address if the MAPSA field is 1.
0 DAVALID	Dynamic Address Valid Determines whether a dynamic address is assigned. This field is 1 for ACK and 0 for NACK when using the A731h KEY. 0b - DANOTASSIGNED: a dynamic address is not assigned 1b - DAASSIGNED: a dynamic address is assigned

69.7.23 Target Maximum Limits (SMAXLIMITS)

Offset

Register	Offset
SMAXLIMITS	68h

Function

Indicates the limits set by the controller (or the originally requested limits). The maximum limits are not enabled in the hardware design, including maximum read and write lengths. If the maximum read and write lengths are enabled, then the current setting (including the default request) shows up in this register (SMAXLIMITS).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				MAXWR											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MAXRD											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 MAXWR	Maximum Write Length Indicates the maximum write length, which must be between 8 to 4095 (saturation). The application must not set the maximum write length to a higher value than the maximum write length set by the controller.
15-12 —	Reserved
11-0 MAXRD	Maximum Read Length Indicates the maximum read length, which must be between 16 to 4095 (saturation). The application must not set the maximum read length to a higher value than the maximum read length set by the controller.

69.7.24 Target ID Part Number (SIDPARTNO)

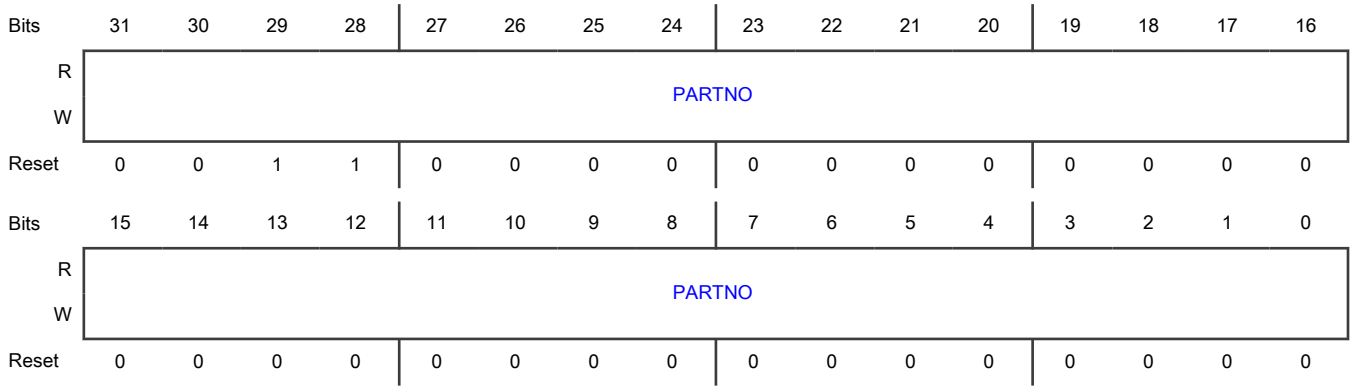
Offset

Register	Offset
SIDPARTNO	6Ch

Function

Allows you to write the ID part number. You must write a nonzero value into the PARTNO field because 0 is not valid.

Diagram



Fields

Field	Function
31-0 PARTNO	Part Number

69.7.25 Target ID Extension (SIDEXT)

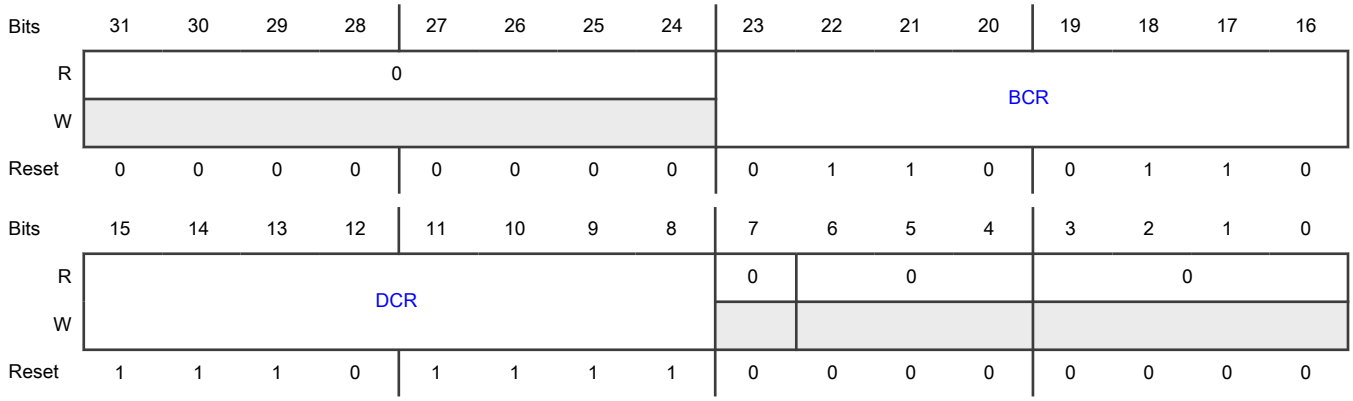
Offset

Register	Offset
SIDEXT	70h

Function

Allows you to write the ID extension of DCR and/or BCR.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 BCR	Bus Characteristics Register Sets the value for BCR, if this field is configured. Otherwise, the default value is considered. This field controls features such as secondary controller and slow-speed requirements.
15-8 DCR	Device Characteristic Register Sets the value for DCR, if this field is configured. Otherwise, the default value is considered.
7 —	Reserved
6-4 —	Reserved
3-0 —	Reserved

69.7.26 Target Vendor ID (SVENDORID)

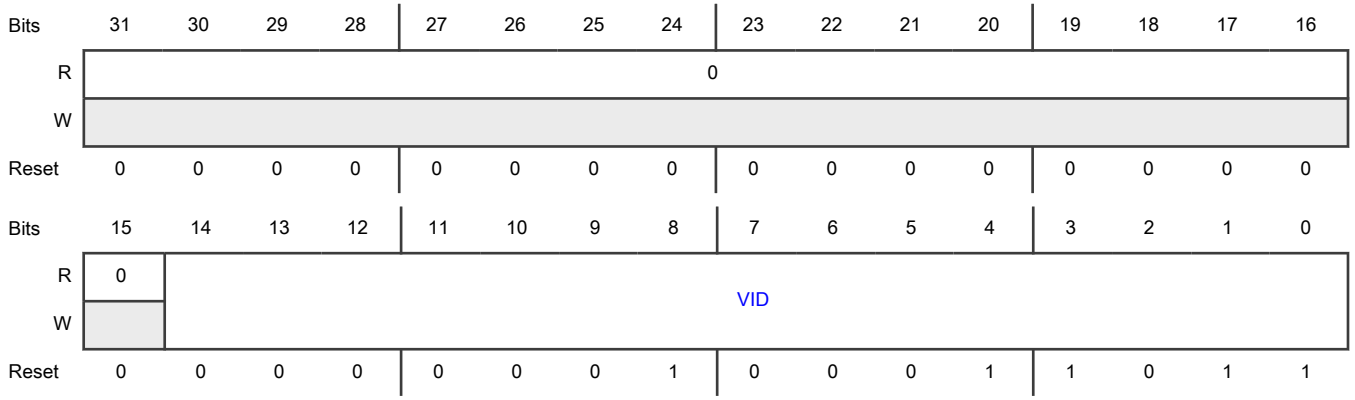
Offset

Register	Offset
SVENDORID	74h

Function

Allows you to write the vendor ID. The default value is the chip vendor ID, and is set from the constant field. When using the chip vendor ID, the part number (PARTNO) does not collide with other uses. The MIPI vendor ID is available to all companies (MIPI membership is not required). To get a vendor ID, make a request at the mipi.org website.

Diagram



Fields

Field	Function
31-15 —	Reserved
14-0 VID	Vendor ID Configures the 15-bit MIPI vendor ID. If not configured, the default value is considered.

69.7.27 Target Time Control Clock (STCCLOCK)

Offset

Register	Offset
STCCLOCK	78h

Function

Allows you to dynamically set the time control clock and accuracy information. The clock frequency and accuracy are constants set by the hardware. If the clock can be adjusted (that is, divided) or if the accuracy could vary with knowable information, then the clock may be set via this register, which must be updated whenever the clock source is changed.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FREQ								ACCURACY							
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Function
31-16 —	Reserved
15-8 FREQ	Clock Frequency Indicates the clock frequency in 0.5 MHz steps. For example, a value of 20 in this field indicates a frequency of 10 MHz. Default set by parameters if configured.

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-0 ACCURACY	Clock Accuracy Indicates the clock accuracy in 1/10ths of %. For example, a value of 15 indicates an accuracy of 1.5%. Default set by parameters if configured.

69.7.28 Target Message Map Address (SMSGMAPADDR)

Offset

Register	Offset
SMSGMAPADDR	7Ch

Function

Allows you to determine which address is matched when STATUS[MATCHED] = 1. This register is used when the DYNADDR register builds a list of extra DAs or SAs to match. It holds the last three matches.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												MAPLASTM2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MAPLASTM1				0			LASTSTA...	MAPLAST			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-16 MAPLASTM2	Matched Previous Index 2 Indicates index 2 of the previous matched address (0 for the base address).
15-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-8 MAPLASTM1	Matched Previous Address Index 1 Indicates index 1 of the previous matched address (0 for the base address).
7-5 —	Reserved
4 LASTSTATIC	Last Static Address Matched Indicates whether the last matched address was an I2C static address or an I3C dynamic address. 0b - I3C dynamic address 1b - I2C static address
3-0 MAPLAST	Matched Address Index Indicates the matched address index for current or last matched message (0 for the base address).

69.7.29 Controller Extended Configuration (MCONFIG_EXT)

Offset

Register	Offset
MCONFIG_EXT	80h

Function

Allows special extended configurations, including for I2C use.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												I3C_CASR_DEL		I3C_CAS_DEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0		0	0	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-18 I3C_CASR_DEL	I3C CAS Delay After Repeated START Increases SCL clock period by a certain value. You can configure this field to delay the clock after a Repeated START to provide more time for slow I3C devices. The maximum value may be less if the internal clock is very fast. 00b - No delay 01b - Increases SCL clock period by 1/2 10b - Increases SCL clock period by 1 11b - Increases SCL clock period by 1 1/2
17-16 I3C_CAS_DEL	I3C CAS Delay After START Increases SCL clock period by a certain value. By increasing the SCL clock period, you can delay the clock to be longer after the START. The maximum value may be less if the internal clock is very fast. 00b - No delay 01b - Increases SCL clock period by 1/2 10b - Increases SCL clock period by 1 11b - Increases SCL clock period by 3/2
15-12 —	Reserved
11-8 —	Reserved
7-6 —	Reserved
5 —	Reserved
4 —	Reserved
3-0 —	Reserved

69.7.30 Controller Control (MCTRL)

Offset

Register	Offset
MCTRL	84h

Function

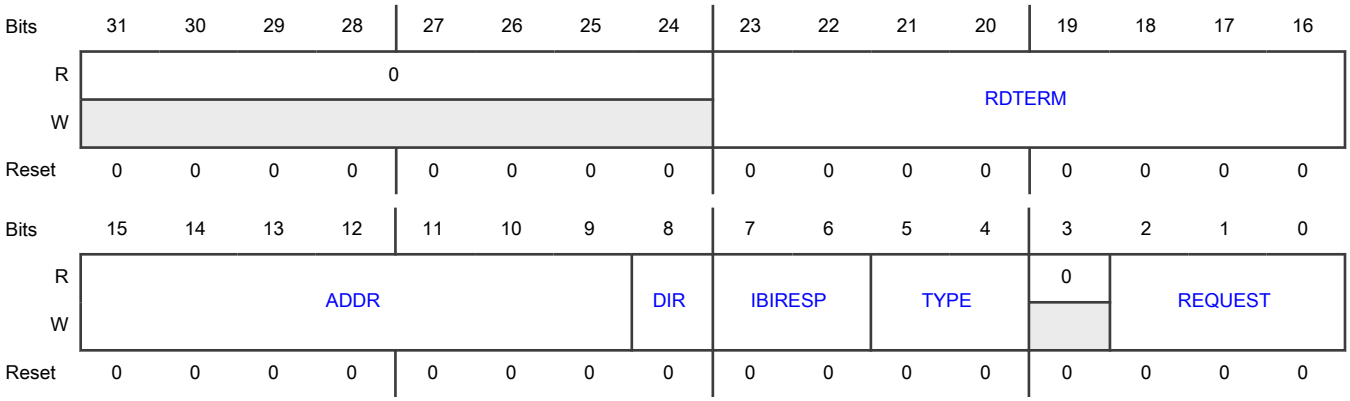
Starts activities on the I3C or I2C bus (see the MWMSG registers). A request cannot be changed when a message is in progress; the REQUEST field becomes 0 automatically.

You must write to MCTRL fields as per use case or as mentioned in Operating modes. Bit read and write checking may not work independently.

NOTE

If MCONFIG[MSTENA] is configured for I2C Controller mode (legacy I2C), only REQUEST = 1 and REQUEST = 2 are accepted. Also, fields are constrained to I2C-supported fields.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 RDTERM	Read Terminate Counter Determines when to terminate a read operation: • For I2C, this field controls when to NACK a read. • For I3C, this field can be used to terminate (end) a read: — If RDTERM is 0, it has no effect. — If RDTERM is 1, the read terminates after the next character.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	— If RDTERM is 2, the read terminates after the next two characters. The field supports up to 255 characters. In DDR mode, RDTERM terminates the read based on word counts (for DDR) instead of byte counts (for SDR). The field self-clears when MSTATUS[COMPLETE] becomes 1. **NOTE** You may write 1 to this field anytime, but a value greater than 1 must be written when starting EmitStartAddress.
15-9 ADDR	Address Indicates the type of address: I3C dynamic address or I2C static address. Some values are not allowed based on the bus that is used (I3C or I2C).
8 DIR	Direction Indicates the direction, write or read. 0b - Write 1b - Read
7-6 IBIRESP	In-Band Interrupt Response Indicates the response to use when you get an IBI from START, and when to force using an IBI ACK NACK request when completing a manual IBI. Completion of a manual IBI means that the target DA is known, and so the mandatory byte (or not) is specified by the application when acknowledging. This field is also used when a message is emitted in Message mode using the MWMSG_SDR or MWMSG_DDR registers. If an IBI with MDB (mandatory byte) is ACKed, the controller limits it to a maximum of eight more bytes after the MDB. RDTERM = 1 can be used with REQUEST = NONE to terminate data from a target sooner. If this field is 0b, it indicates ACK (acknowledge). When REQUEST = 1 (EmitStartAddr) or REQUEST = 7 (AutoIBI), Controller In-band Interrupt Registry and Rules (MIBIRULES) decides whether to ACK with mandatory byte. When REQUEST = 3 (IBIAckNack), ACK with no mandatory byte. If this field is 1b, it indicates NACK (reject) or no acknowledgement. If this field is 10b, it indicates acknowledge with mandatory byte. When REQUEST = 1 or REQUEST = 7, ignore Controller In-band Interrupt Registry and Rules (MIBIRULES). Do not use this setting unless only targets with a mandatory byte can cause an IBI. When REQUEST = 3, ACK with mandatory byte. If this field is 11b, it indicates manual. When REQUEST = 1 or REQUEST = 7, stop and wait for a decision using the IBI ACK NACK request. When REQUEST = 3, this field is reserved. **NOTE** A CR and HJ always cause IBIRESP to become 3 (manual) so the application must decide. 00b - ACK (acknowledge) 01b - NACK (reject) 10b - Acknowledge with mandatory byte

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11b - Manual
5-4 TYPE	Bus Type with EmitStartAddr Works with REQUEST to determine the bus type. See Determining bus types and modes for more information. The following values are valid only if REQUEST is 1. 00b - I3C 01b - I2C 10b - DDR 11b - Reserved
3 —	Reserved
2-0 REQUEST	Request Emits the requested operation when performing in pieces, instead of performing by message. You must check Controller Status (MSTATUS) because some requests can only be made in some states. For example, the system cannot enter SDR mode from DDR mode, and it cannot use an incorrect request in DAA mode. If this field is 0b, it indicates that no request is present (NONE). The REQUEST field returns to NONE after finishing a request. Controller Status (MSTATUS) indicates the state of the controller. See AutoIBI mode. NONE is written as 0 only in these cases: when writing 1 to MCTRL[RDTERM] (to stop a read in progress) or when configuring MCTRL[IBIRESP] for MSG use. If this field is 1b, it emits START with address and direction (EMITSTARTADDR), either from Stopped state or in the middle of a single data rate (SDR) message. If from Stopped state (Idle), then EmitStart may be prevented by an event (such as IBI, CR, HJ). In this case, an appropriate interrupt is signaled. EmitStart can be resubmitted. If this field is 10b, it emits a STOP on bus (EMITSTOP). Must be in SDR mode. In DAA mode, emitting STOP exits DAA mode. If this field is 11b, it indicates manual IBI ACK or NACK (IBIACKNACK). When MCTRL[IBIRESP] indicates a hold on an IBI to allow a manual decision, this request completes it. The field uses MCTRL[IBIRESP] to provide the information. If this field is 100b, and if currently not in DAA mode, it emits START, 7E, ENTDAA sequence, and then emits 7E/R to process the first target (PROCESSDAA). Stops just before the new dynamic address (DA) is to be emitted. The DA is written using Controller Write Data Byte (MWDATAB), then process DAA is requested again to write the new address, and then it starts the next unless marked to STOP. An MSTATUS indicating NACK means DA was not accepted (for example, parity error). If PROCESSDAA is NACKed on the 7E/R request, meaning no more targets need a DA, then a COMPLETE is signaled (along with DONE) and a STOP issued. If TYPE = 2 or 3, the DA is assigned and then it emits a STOP (instead of starting a new 7E/R request). If TYPE = 1 or 3, then the DA is taken from the ADDR field (bits 6:0). If this field is 110b, it emits an exit pattern from any state (force exit and target reset). End DDR mode (including MSGDDR), including a STOP afterward. If MCTRL[TYPE] is 2, perform a target reset action

Table continues on the next page...

Table continued from the previous page...

Field	Function
	(RSTACT can prevent the reset in the target). Target reset may follow immediately after RSTACT CCC or STOP. If this field is 111b, the target is held in a Stopped state, but autoemits a START, 7E sequence when the target holds SDA low for an IBI (AUTOIBI). Actual IBI handling is defined by MCTRL[IBIRESP]. 000b - NONE 001b - EMITSTARTADDR 010b - EMITSTOP 011b - IBIACKNACK 100b - PROCESSDAA 101b - Reserved 110b - Force Exit and Target Reset 111b - AUTOIBI

69.7.31 Controller Status (MSTATUS)

Offset

Register	Offset
MSTATUS	88h

Function

Indicates the controller status, including which events cause interrupts. The peripherals share the IRQ (called parallel-to-target status).

Because a peripheral can either be in Controller or Target mode, but not both at the same time, only one (target or controller peripheral) can cause the IRQ. If there is an IRQ and the peripheral is a controller, then this register (MSTATUS) indicates the status.

If there is an IRQ and the peripheral is a target, then Target Status (SSTATUS) indicates the status.

This register self-clears if the COMPLETE field becomes 1.

If the MCONFIG register is configured only for I2C, some states and fields never remain active (for example, DAA or IBI).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	IBIADDR							0				NOWMAST...	0		
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	0	IBIWON	TXNOTFU...	RXPEND	COMPLETE	MCTRLDO...	SLVSTART	IBITYPE		NACKED	BETWEEN	0	STATE		
W			W1C			W1C	W1C	W1C								
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-24 IBIADDR	IBI Address Indicates the address of: • The IBI when MSTATUS[IBITYPE] = 1. • The CR, when MSTATUS[IBITYPE] = 2. • 7'h2 when HJ, when MSTATUS[IBITYPE] = 3.
23-20 —	Reserved
19 NOWMASTER	Module is now Controller Flag Indicates whether the module is now a controller. That is, it was previously a target, and controllership acceptance was requested from the previous controller and controllership was accepted. The reverse operation (controller becomes a target) does not need an interrupt because the application grants it through the GETACCMST CCC. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not a controller 1b - Controller When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
18-16 —	Reserved
15 ERRWARN	Error or Warning Indicates whether an error occurred, such as improper register use, overrun, or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read. See Controller Errors and Warnings (MERRWARN) for more information. 0b - No error or warning 1b - Error or warning
14 —	Reserved
13 IBIWON	In-Band Interrupt (IBI) Won Flag Indicates whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed. Arbitration requires manual intervention for CR and HJ, and optionally requires it for IBI if MCTRL[IBIRESP] = 3 (manual). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No IBI arbitration won 1b - IBI arbitration won When writing 0b - No effect 1b - Clear the flag
12 TXNOTFULL	TX Buffer or FIFO Not Full Indicates whether the buffer, FIFOm, or message register can accept another byte or halfword. FIFO uses trigger level. If DMA is enabled for transmitting, it transfers data as long as it is not full. 0b - Receive buffer or FIFO full 1b - Receive buffer or FIFO not full
11 RXPEND	RXPEND Indicates whether a message is being received from a target and bytes are in the input buffer or FIFO: • When using a FIFO, this message is at least one FIFO trigger's worth (a minimum of one byte in the FIFO). • When DMA is enabled for receiving, the DMA is signaled. RXPEND becomes 0 when the data is read.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No receive message pending 1b - Receive message pending
10 COMPLETE	Complete Flag Indicates whether a message is complete: • With MWMSG_SDR or MWMSG_DDR, the COMPLETE condition occurs when MWMSG_SDR_CONTROL[LEN] or MWMSG_DDR_CONTROL[LEN] reaches 0. • When MCTRL[REQUEST] = 1 (EmitStartAddr), this COMPLETE condition occurs after the end of a write operation, or after a read operation has terminated or ended. • With an IBI (AutoIBI or EmitStartAddr), the COMPLETE condition occurs at the end of IBI data (if any). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
9 MCTRLDONE	Controller Control Done Flag Indicates whether the module has completed an MCTRL request. MCTRLDONE automatically becomes 0 when writing a new control. If MCTRL[REQUEST] = 1 (EmitStartAddr), MCTRLDONE becomes 1 when the address goes out (and is ACKed, NACKed, or ended in an IBI). If ACKed, MSTATUS[COMPLETE] becomes 1 after the write or read data is complete. If MCTRL[REQUEST] = 4 (ProcessDAA), MCTRLDONE becomes 1 when the module is ready to emit the dynamic address (DA) for the target, or when no more targets are ACKing. This condition can be determined by using the MSTATUS[BETWEEN] and MSTATUS[STATE] fields. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not done 1b - Done When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 SLVSTART	Target Start Flag Indicates whether a target is or was requesting a START by holding SDA low. Handling starts automatically when MCTRL[REQUEST] = 7 (AutoIBI). **NOTE** This field behaves differently for register reads and writes. When reading 0b - Target not requesting START 1b - Target requesting START When writing 0b - No effect 1b - Clear the flag
7-6 IBITYPE	In-Band Interrupt (IBI) Type Indicates the type of IBI of the last event that won the arbitration, whether the interrupt was ACKed, NACKed, or pending. 00b - NONE (no IBI: this status occurs when MSTATUS[IBIWON] becomes 0) 01b - IBI 10b - CR 11b - HJ
5 NACKED	Not Acknowledged Indicates whether the last start and address sequence was NACKed (was not ACKed by the addressed target). 0b - Not NACKed 1b - NACKed (not acknowledged)
4 BETWEEN	Between Indicates whether the controller is active between messages or dynamic address assignments (DAA). It is active when: • MSTATUS[STATE] is MSGSDR, DDR, or DAA, or NORMACT, and the state is between messages or DAAs. It is expecting a new message or DAA to start (or STOP or exit). • MSTATUS[STATE] is NORMACT. The module is waiting for the transmit FIFO to be not empty or the receive FIFO to be not full. 0b - Inactive (for other cases) 1b - Active
3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
2-0 STATE	State of the Controller Indicates the current controller state. If this field is 0b, it indicates that the controller is idle (the bus has stopped.) If this field is 1b, it indicates target request. The bus has stopped but a target is holding SDA low. When using auto-emit IBI (MCTRL[REQUEST] = 7), the controller does not remain in this state. If this field is 10b, it indicates entry into Single Data Rate Message mode, using MWMSG_SDR. If this field is 11b, it indicates entry into normal active SDR mode, using MCTRL, MWDATA*n*, and MRDATA*n* registers. The controller remains in this state until a STOP is issued. If this field is 100b, it indicates entry into Double Data Rate Message mode, using MWMSG_DDR or the normal method with DDR. The controller remains in the DDR state until the controller exits using EXIT (emitting the Exit pattern). If this field is 101b, it indicates entry into Dynamic Address Assignment (ENTDAA) mode. If this field is 110b, it indicates wait for an IBI ACK/NACK decision. If this field is 111b, it indicates receiving an IBI. This state is used after IBI, CR, or HJ has won an arbitration. The IBIRCV state is also used for IBI mandatory byte (if any) and any bytes that follow. 000b - IDLE (bus has stopped) 001b - SLVREQ (target request) 010b - MSGSDR 011b - NORMACT 100b - MSGDDR 101b - DAA 110b - IBIACK 111b - IBIRCV

69.7.32 Controller In-band Interrupt Registry and Rules (MIBIRULES)

Offset

Register	Offset
MIBIRULES	8Ch

Function

Contains the rules for using IBI, and keeps a registry of the targets that use the IBI byte.

This register defines the IBI mandatory byte rules for the targets and determines which targets do (or do not) have a mandatory byte.

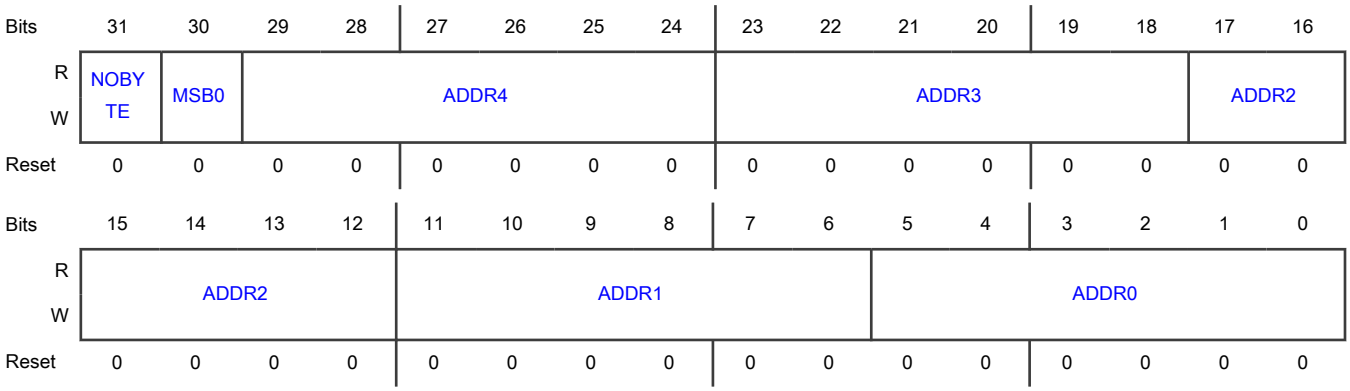
Concerning ADDR*n* fields: The address has 6 bits. Assuming that MIBIRULES[MSB0] = 1, the most significant bit of each address is 0. In this case, each address is 7 bits (usually written as A7 to A1) and A7 must be 0. If the application does not use that optimal convention, the "manual" method of IBI ACK handling must be used (see MCTRL[IBIRESP] for more information).

By default, the ADDR*n* values indicate the targets with a mandatory byte. If MIBIRULES[NOBYTE] = 1, the ADDR*n* values indicate targets that do not have a mandatory IBI byte.

NOTE

A7 = 0 is only needed for targets that use IBI. For legacy I2C devices, A7 can use any valid value because I2C devices cannot cause an IBI.

Diagram



Fields

Field	Function
31 NOBYTE	No IBI byte Specifies whether the ADDR*n* fields refer to targets with or without a mandatory IBI byte. 0b - With mandatory IBI byte 1b - Without mandatory IBI byte
30 MSB0	Most Significant Address Bit is 0 Specifies whether MSB is 0 for all I3C dynamic addresses. Assigning 1 to this field allows the START header to be optimized. 0b - MSB is not 0 1b - MSB is 0
29-24 ADDR4	ADDR4 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
23-18 ADDR3	ADDR3 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
17-12 ADDR2	ADDR2 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.

Table continues on the next page...

Table continued from the previous page...

Field	Function
11-6 ADDR1	ADDR1 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
5-0 ADDR0	ADDR0 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.

69.7.33 Controller Interrupt Set (MINTSET)

Offset

Register	Offset
MINTSET	90h

Function

Returns the status of the interrupt enables when read. This register sets interrupt enables for select fields in Controller Status (MSTATUS):

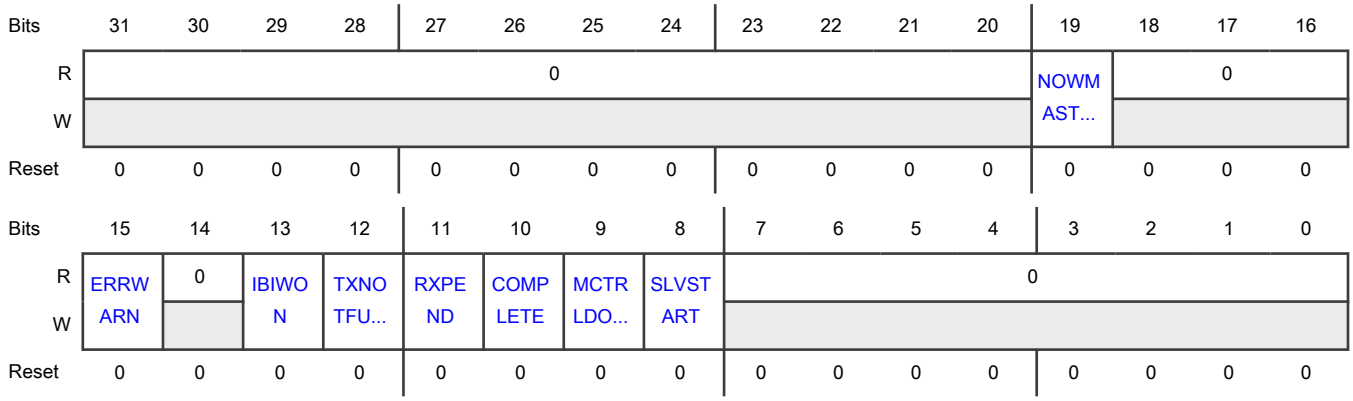
- To activate an interrupt enable, write 1 to its corresponding field.
- To disable an interrupt enable, write 1 to the appropriate field in Controller Interrupt Clear (MINTCLR). Writing 0 to the interrupt enable in MINTSET does not disable the interrupt.

The interrupt registers allow masking of interrupt sources as well as checking which interrupts are activated in Controller Status (MSTATUS). The normal method is to enable an interrupt and then after that interrupt fires, clear the interrupt either by writing to Controller Status (MSTATUS) or via an action on the corresponding data register. The interrupt is level-held, meaning that the interrupt remains 1 until the cause is cleared by some method. The module prevents races; if a new event comes in, that new event is not lost.

These interrupts are parallel to Target Interrupt Set (SINTSET) and Target Interrupt Clear (SINTCLR), and only one interrupt set is active depending on the state (controller or target operation):

- MINTSET: Sets interrupt enables for MSTATUS fields. Reading the MINTSET register returns the status of the interrupt enables.
- MINTCLR: Clears interrupt enables for MSTATUS fields.
- MINTMASKED: Returns the value of the MSTATUS fields ANDed with their interrupt enables.

Diagram



Fields

Field	Function
31-20 —	Reserved
19 NOWMASTER	Now Controller Interrupt Enable Enables the corresponding interrupt to indicate whether the module is now a controller (now this I3C module is a controller). That is, it was previously a target; controllership acceptance was requested from the previous controller and it was accepted. 0b - Disable 1b - Enable
18-16 —	Reserved
15 ERRWARN	Error or Warning (ERRWARN) Interrupt Enable Enables the corresponding interrupt to indicate whether an error occurred, such as improper register use, overrun or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read. 0b - Disable 1b - Enable
14 —	Reserved
13 IBIWON	IBI Won Interrupt Enable Enables the corresponding interrupt to indicate whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed. 0b - Disable 1b - Enable
12	Transmit Buffer/FIFO Not Full Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Function
TXNOTFULL	Enables the corresponding interrupt to indicate whether the buffer, FIFOm, or message register can accept another byte or halfword. 0b - Disable 1b - Enable
11 RXPEND	Receive Pending Interrupt Enable Enables the corresponding interrupt to indicate whether a message is being received from a target and bytes are in the input buffer or FIFO.
10 COMPLETE	Completed Message Interrupt Enable Enables the corresponding interrupt to indicate whether a message is complete. 0b - Disable 1b - Enable
9 MCTRLDONE	Controller Control Done Interrupt Enable Enables the corresponding interrupt to indicate whether the module has completed an MCTRL request. 0b - Disable 1b - Enable
8 SLVSTART	Target Start Interrupt Enable Enables the corresponding interrupt to indicate whether a target is or was requesting a START by holding SDA low. 0b - Disable 1b - Enable
7-0 —	Reserved

69.7.34 Controller Interrupt Clear (MINTCLR)

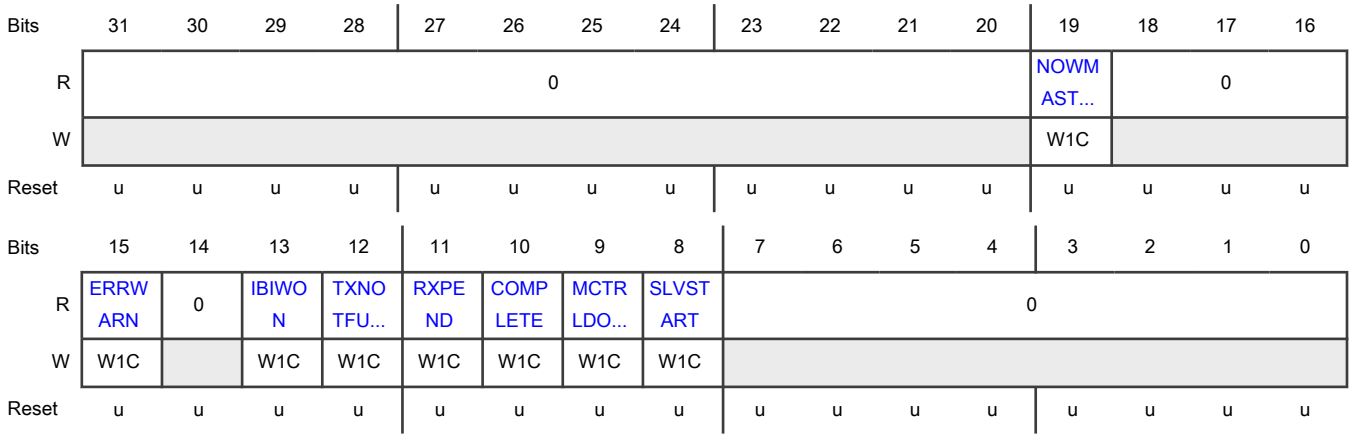
Offset

Register	Offset
MINTCLR	94h

Function

Clears interrupt enables for select fields in Controller Status (MSTATUS). Writing 1 clears the corresponding interrupt enable. Writing 0 has no effect.

Diagram



Fields

Field	Function
31-20 —	Reserved
19 NOWMASTER	NOWCONTROLLER Interrupt Enable Clear Flag Clears the corresponding NOWCONTROLLER interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
18-16 —	Reserved
15 ERRWARN	ERRWARN Interrupt Enable Clear Flag Clears the corresponding ERRWARN interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared

Table continues on the next page...

Table continued from the previous page...

Field	Function
	When writing 0b - No effect 1b - Clear the flag
14 —	Reserved
13 IBIWON	IBIWON Interrupt Enable Clear Flag Clears the corresponding IBIWON interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
12 TXNOTFULL	TXNOTFULL Interrupt Enable Clear Flag Clears the corresponding TXNOTFULL interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
11 RXPEND	RXPEND Interrupt Enable Clear Flag Clears the corresponding RXPEND interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
10 COMPLETE	COMPLETE Interrupt Enable Clear Flag Clears the corresponding COMPLETE interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
9 MCTRLDONE	MCTRLDONE Interrupt Enable Clear Flag Clears the corresponding MCTRLDONE interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing 0b - No effect 1b - Clear the flag
8 SLVSTART	SLVSTART Interrupt Enable Clear Flag Clears the corresponding SLVSTART interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No effect 1b - Interrupt enable cleared When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
7-0 —	Reserved

69.7.35 Controller Interrupt Mask (MINTMASKED)

Offset

Register	Offset
MINTMASKED	98h

Function

Returns the status of enabled interrupts (the value of Controller Status (MSTATUS) ANDed with the value of Controller Interrupt Set (MINTSET)).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												NOWMAST...	0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	0	IBIWON	TXNOTFU...	RXPEND	COMPLETE	MCTRLDO...	SLVSTART	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19 NOWMASTER	NOWCONTROLLER Interrupt Mask Indicates whether the NOWCONTROLLER interrupt is enabled and active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disabled 1b - Enabled
18-16 —	Reserved
15 ERRWARN	ERRWARN Interrupt Mask Indicates whether the ERRWARN interrupt is enabled and active. 0b - Disabled 1b - Enabled
14 —	Reserved
13 IBIWON	IBIWON Interrupt Mask Indicates whether the IBIWON interrupt is enabled and active. 0b - Disabled 1b - Enabled
12 TXNOTFULL	TXNOTFULL Interrupt Mask Indicates whether the TXNOTFULL interrupt is enabled and active. 0b - Disabled 1b - Enabled
11 RXPEND	RXPEND Interrupt Mask Indicates whether the RXPEND interrupt is enabled and active.
10 COMPLETE	COMPLETE Interrupt Mask Indicates whether the COMPLETE interrupt is enabled and active. 0b - Disabled 1b - Enabled
9 MCTRLDONE	MCTRLDONE Interrupt Mask Indicates whether the MCTRLDONE interrupt is enabled and active. 0b - Disabled 1b - Enabled
8 SLVSTART	SLVSTART Interrupt Mask Indicates whether the SLVSTART interrupt is enabled and active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Disabled 1b - Enabled
7-0 —	Reserved

69.7.36 Controller Errors and Warnings (MERRWARN)

Offset

Register	Offset
MERRWARN	9Ch

Function

Contains errors and warnings. When any errors or warnings are not 0, then MSTATUS[ERRWARN] is 1.

Parallel-to-target ERRWARN:

- In Controller mode, use this register (MERRWARN).
- In Target mode, use Target Errors and Warnings (SERRWARN).

The error fields in both registers (MERRWARN and SERRWARN) are similar in meaning.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										0	TIMEOUT	INVREQ	MSGERR	OWRITE	OREAD
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			0	0	HCRC	HPAR	0				TERM	WRABT	NACK	URUN	0
W						W1C	W1C					W1C	W1C	W1C	W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
21 —	Reserved
20 TIMEOUT	Timeout Error Flag Indicates an error caused by the module stalling for too long in a frame. This stalling occurs when: • The transmit FIFO or receive FIFO is not handled, and the bus is stuck in the middle of a message. • No STOP is issued after data transfer (there is a gap between messages). • IBI manual is used and no decision has been made. The maximum stall period is 10 kHz or 100 µs. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
19 INVREQ	Invalid Request Error Flag Indicates an error caused by an invalid use of a request: • Not using IBI ACK NACK when stopped in manual hold for IBI acknowledgment. • Using a request other than ForceStop or ForceExit when in a message. Other requests are valid when the message is done. • Other mismatched uses (for example, IBI ACK NACK in normal states). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
18 MSGERR	Message Error Flag Indicates an error caused by:

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• Trying to write to or read from the MWMSG_SDR register when in a DDR message. • Trying to write to or read from the MWMSG_DDR register when in an SDR message. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
17 OWRITE	Overwrite Error Flag Indicates an error caused by trying to write to Controller Write Data Byte (MWDATAB) when the FIFO is full. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
16 OREAD	Overread Error Flag Indicates an error caused by: • Trying to read from Controller Read Data Byte (MRDATAB) when the FIFO is empty. • Trying to read from the MRMSG_SDR register or the MRMSG_DDR register when no message has yet started. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
15-13 —	Reserved
12 —	Reserved
11 —	Reserved
10 HCRC	High Data Rate CRC Error Flag Indicates that a cyclic redundancy check (CRC) error occurred from a DDR read. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
9 HPAR	High Data Rate Parity Flag Indicates a parity error from a DDR read, including a bad preamble on a read. This does not stop the read because it is not safe to terminate; the read data may become misframed. Ends on a run of 1 second. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
8-5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
4 TERM	Terminate Error Flag Indicates an error when this controller terminates a target read because the read exceeded the count for the message. This error is valid only when using the MWMSG_SDR or MWMSG_DDR register. If you write to the MWMSG_SDR or MWMSG_DDR register, this field automatically becomes 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
3 WRABT	Write Abort Error Flag Indicates an error caused by the I2C target NACKing the write data, terminating the message. For example, the controller is writing in I2C and the target NACKed the write. If you write to the MCTRL register, this field automatically becomes 0. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
2 NACK	Not Acknowledge Error Flag Indicates an error caused by the target or targets NACKing (not acknowledging) the last address. If 7Eh is the address, then it indicates all targets NACKed the last address. If you write to the MCTRL register, this field automatically becomes 0. **NOTE** In HDR mode, this error occurs when an address is not accepted (as opposed to NACKed). **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
1 URUN	Underrun Error Flag Indicates an underrun for HDR-BT. This error occurs when attempting to write from an empty transmit FIFO. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No error 1b - Error When writing 0b - No effect 1b - Clear the flag
0 —	Reserved

69.7.37 Controller DMA Control (MDMACTRL)

Offset

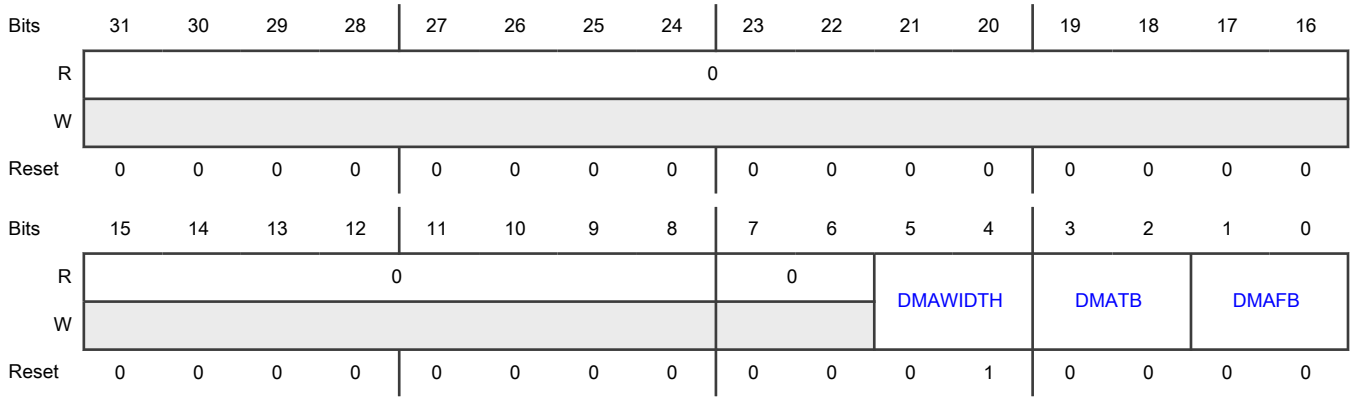
Register	Offset
MDMACTRL	A0h

Function

Allows DMA to be used for inbound and outbound messages. DMA is much more useful for a controller than for a target because the controller directs the bus traffic and actions. DMA can be used with a controller in these ways:

- Push or pull data for MCTRL[REQUEST] = 1 (EmitStartAddr) request written by the processor.
- Implementing message mode, completely controlled by DMA.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-6 —	Reserved
5-4 DMAWIDTH	DMA Width Specifies the data width of DMA operations. The halfword (16 bits) setting ensures that two bytes are free or available in FIFO. 00b,01b - Byte 10b - Halfword (16 bits) 11b - Reserved
3-2 DMATB	DMA to Bus Represents the DMA write (to-bus) trigger. When DMATB = 1 or DMATB = 2, I3C requests DMA when a transmit trigger occurs (see Controller Data Control (MDATACTRL)). I3C requests until full, unless the DMA is set up as a trigger. DMAFB becomes 0 when MSTATUS[ERRWARN] = 1. If this field is 1b, STOP or repeated START causes DMATB to become 0 (see SCONFIG[MATCHSS]). Normally, DMA enable must be used only in Controller Message mode. 00b - DMA not used 01b - Enable DMA for one frame (ended by DMA or terminated) 10b - Enable DMA until DMA is turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11b - Reserved
1-0	DMA from Bus

Table continues on the next page...

Table continued from the previous page...

Field	Function
DMAFB	Represents the DMA read (from-bus) trigger. When DMAFB = 1 or DMAFB = 2, I3C requests DMA when a receive trigger occurs (see Controller Data Control (MDATACTRL)). I3C requests until empty unless the DMA is set up as a trigger. DMAFB becomes 0 when MSTATUS[ERRWARN] = 1. If this field is 1b, STOP or Repeated START causes DMAFB to automatically become 0 (see SCONFIG[MATCHSS]). **NOTE** Do not enable DMA after a transaction starts. It must be enabled only when enabling the controller and interrupts to avoid any kind of RX FIFO overrun. 00b - DMA not used 01b - Enable DMA for one frame 10b - Enable DMA until DMA is turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11b - Reserved

69.7.38 Controller Data Control (MDATACTRL)

Offset

Register	Offset
MDATACTRL	ACh

Function

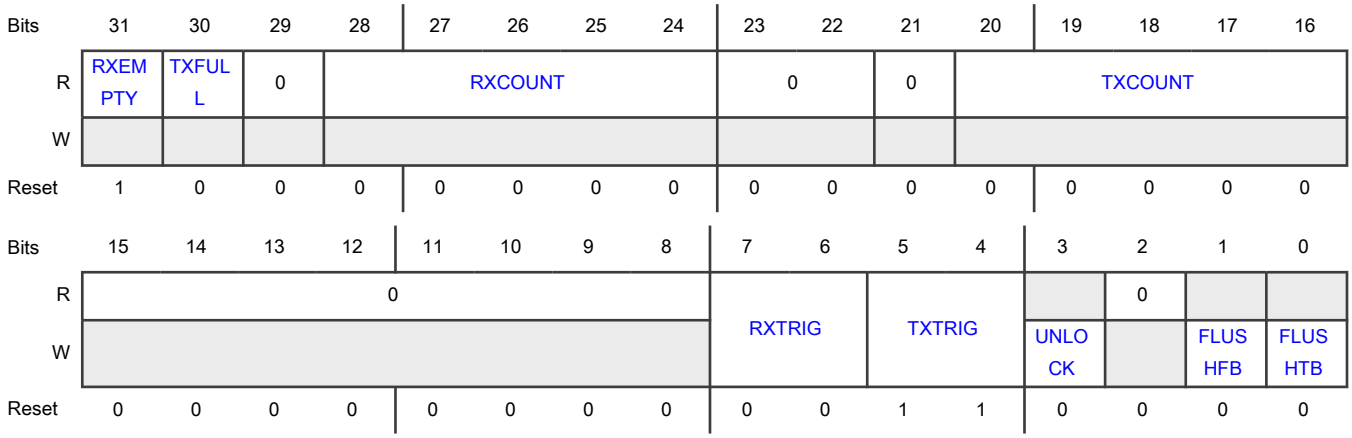
Assists in data control when there is no FIFO. This register also assists the FIFO when FIFO is available (regardless of size), and this assistance allows some control over the FIFO behavior. In particular, the register provides control over when to interrupt when a particular state of fullness or emptiness is reached. It also controls behavior related to width, when the width is not 1 byte wide.

This register acts as an alias of Target Data Control (SDATACTRL).

NOTE

When flushing a FIFO if DMA is in use, disable the DMA channel first. You must not use FIFO flush when a message (in that direction) is in flight.

Diagram



Fields

Field	Function
31 RXEMPTY	Receive is Empty Indicates whether the receive FIFO or buffer is empty. 0b - Not empty 1b - Empty
30 TXFULL	Transmit is Full Indicates whether the transmit FIFO or buffer is full. 0b - Not full 1b - Full
29 —	Reserved
28-24 RXCOUNT	Receive Entry Count Contains the count of entries in the receive FIFO or buffer. Entry size for SDR and HDR-DDR is in bytes.
23-22 —	Reserved
21 —	Reserved
20-16 TXCOUNT	Transmit Entry Count Contains the count of entries waiting in the TXFIFO. This count is the number of entries that the application has written to the transmit FIFO that have not yet gone onto the bus. Entry size for SDR and HDR-DDR is in bytes.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-8 —	Reserved
7-6 RXTRIG	Receive Trigger Level Indicates the trigger level for receive fullness when using a FIFO. This field affects the RXPEND interrupt. 00b - Trigger when not empty (default) 01b - Trigger when 1/4 full or more 10b - Trigger when 1/2 full or more 11b - Trigger when 3/4 full or more
5-4 TXTRIG	Transmit Trigger Level Indicates the trigger level for transmit emptiness when using a FIFO. This field affects the TXNOTFULL interrupt. 00b - Trigger when empty 01b - Trigger when 1/4 full or less 10b - Trigger when 1/2 full or less 11b - Trigger when 1 less than full or less (default)
3 UNLOCK	Unlock Unlocks the functionality so that the RXTRIG and TXTRIG fields can be changed on a write. This field must be 1 (unlocked) in the same cycle when writing to TXTRIG or RXTRIG. If this field is 0 (locked), RXTRIG and TXTRIG fields cannot be changed on a write. 0b - Locked 1b - Unlocked
2 —	Reserved
1 FLUSHFB	Flush From-Bus Buffer or FIFO Flushes from-bus buffer or FIFO. You normally do not use this field. **NOTE** Before using FLUSHFB, disable the DMA (MDMACTRL[DMAFB] = 0). Then use FLUSHFB to flush the FIFO and then re-enable the DMA. 0b - No action 1b - Flush the buffer
0	Flush To-Bus Buffer or FIFO

Table continues on the next page...

Table continued from the previous page...

Field	Function
FLUSHTB	Flushes to-bus buffer or FIFO. This field is used when the controller terminates a to-bus message (read) prematurely. **NOTE** Before using FLUSHTB, disable the DMA (MDMACTRL[DMATB] = 0). Then use FLUSHTB to flush the FIFO and then re-enable the DMA. 0b - No action 1b - Flush the buffer

69.7.39 Controller Write Data Byte (MWDATAB)

Offset

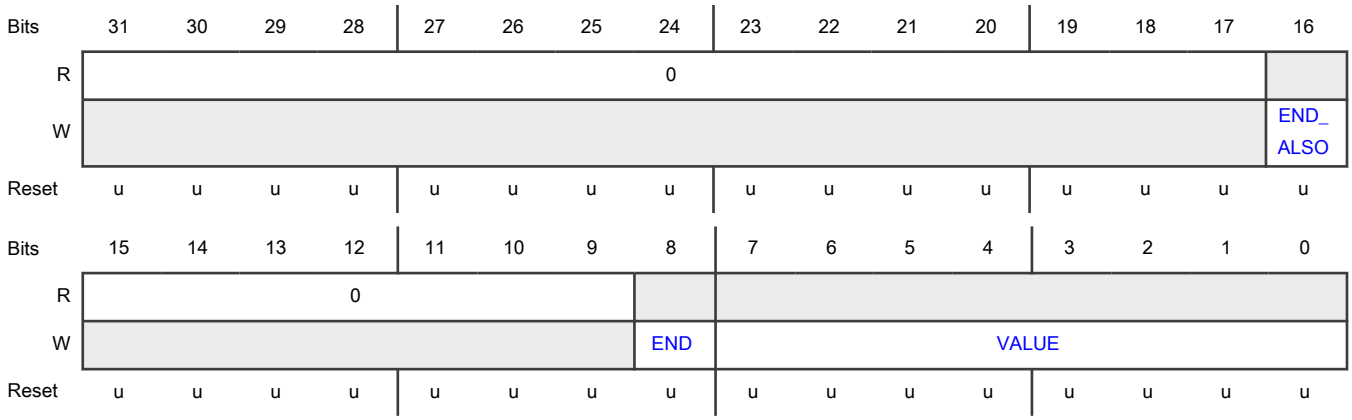
Register	Offset
MWDATAB	B0h

Function

Allows writing bytes to send onto the bus. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Write Data Byte (SWDATAB).

Diagram



Fields

Field	Function
31-17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 END_ALSO	End of Message ALSO Indicates end of message, used to end an outbound message normally. Every message must indicate when it is the last message to be sent. This method can be used with the MDATABE register. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0b - Not the end 1b - End
15-9 —	Reserved
8 END	End of Message Indicates the end of a message. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0b - Not the end 1b - End
7-0 VALUE	Data Byte Represents the byte written to the target (stored in transmit FIFO): • I3C computes the parity. • I2C manages the ACK or NACK.

69.7.40 Controller Write Data Byte End (MWDATABE)

Offset

Register	Offset
MWDATABE	B4h

Function

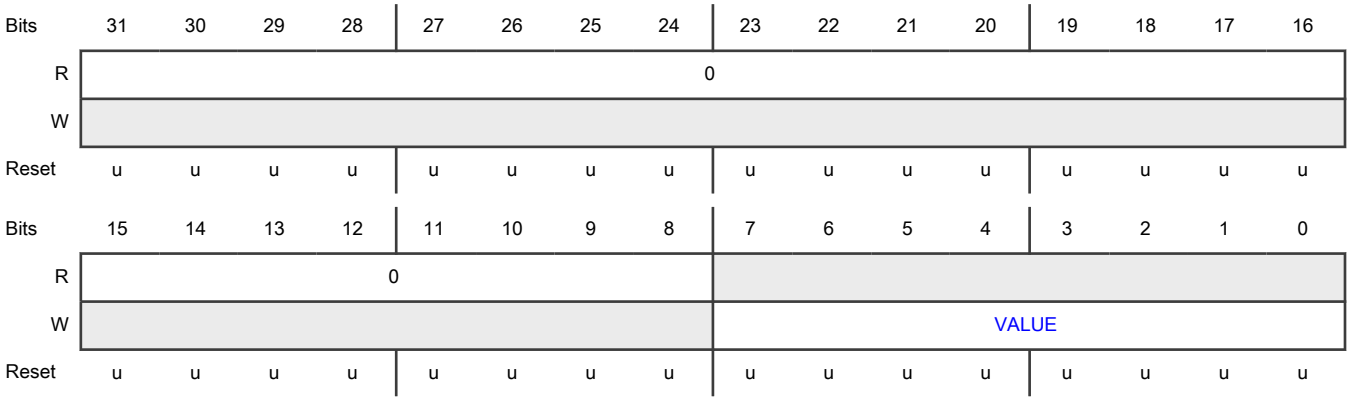
Allows writing the last byte to send onto the bus. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Write Data Byte End (SWDATABE).

NOTE

MWDATABE can also indicate END using bit 8.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 VALUE	Data Represents the last byte written to the target (stored in transmit FIFO): • I3C computes the parity. • I2C manages the ACK or NACK.

69.7.41 Controller Write Data Halfword (MWDATAH)

Offset

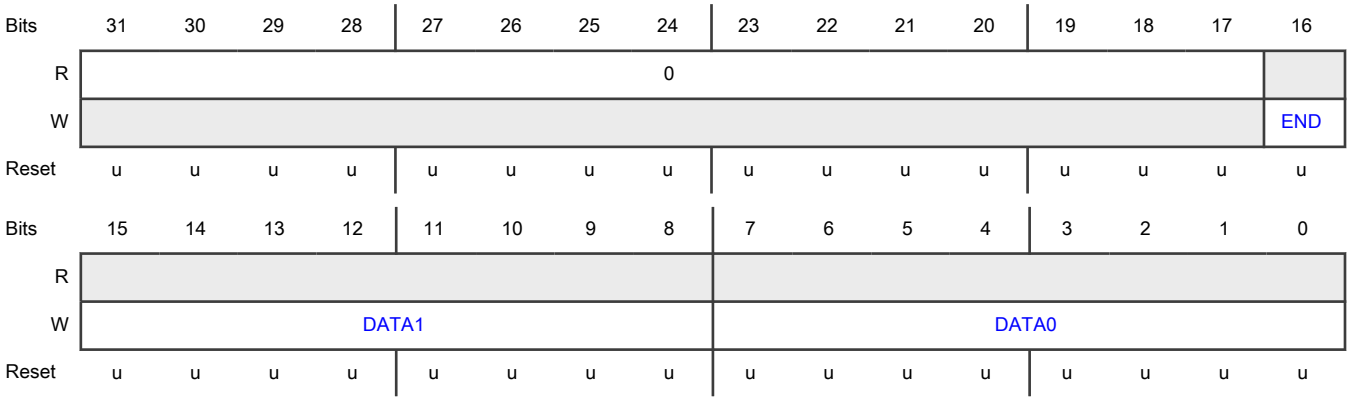
Register	Offset
MWDATAH	B8h

Function

Allows writing a halfword (pair of bytes) to the bus unless an external FIFO is used, sending the low byte followed by the high byte. An end-of-data (last) marker bit is allowed (or must be 0). A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or MDATACTRL[TXCOUNT].

This register acts as an alias of Target Write Data Halfword (SWDATAH).

Diagram



Fields

Field	Function
31-17 —	Reserved
16 END	End of Message Indicates the end of a message. For this register, this field always marks DATA1 as the end. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0b - Not the end 1b - End
15-8 DATA1	Data Byte 1 Represents the second byte that is sent to the target (written to transmit FIFO).
7-0 DATA0	Data Byte 0 Represents the first byte that is sent to the target (written to transmit FIFO).

69.7.42 Controller Write Data Halfword End (MWDATAHE)

Offset

Register	Offset
MWDATAHE	BCh

Function

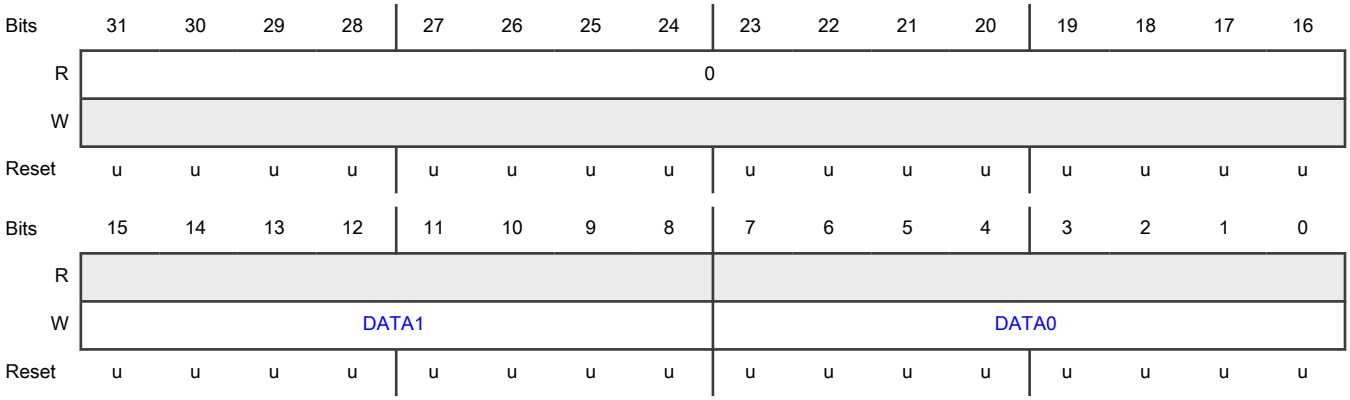
Allows writing a halfword of data, which is the end (the last byte of the halfword is the end). The target writes the halfword (byte pair) in the same way as it does to Controller Write Data Halfword (MWDATAH), but marks the second byte as end-of-data (last byte).

For HDR-DDR, the byte with the END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs.

You must not write a halfword unless there is room for both, as indicated by the use of transmit FIFO level trigger or MDATACTRL[TXCOUNT].

This register acts as an alias of Target Write Data Halfword End (SWDATAHE).

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 DATA1	Data Byte 1 Represents the second byte that is sent to the target (written to transmit FIFO).
7-0 DATA0	Data Byte 0 Represents the first byte that is sent to the target (written to transmit FIFO).

69.7.43 Controller Read Data Byte (MRDATAB)

Offset

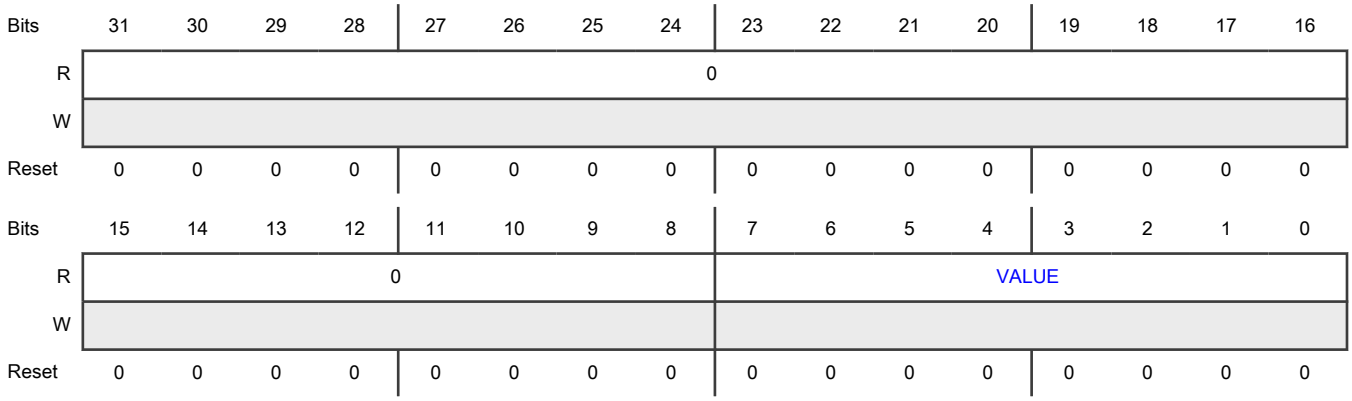
Register	Offset
MRDATAB	C0h

Function

Allows reading bytes written by the target after an SDR read, or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Read Data Byte (SRDATAB).

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 VALUE	Value Represents the byte read from the controller (and written by the target).

69.7.44 Controller Read Data Halfword (MRDATAH)

Offset

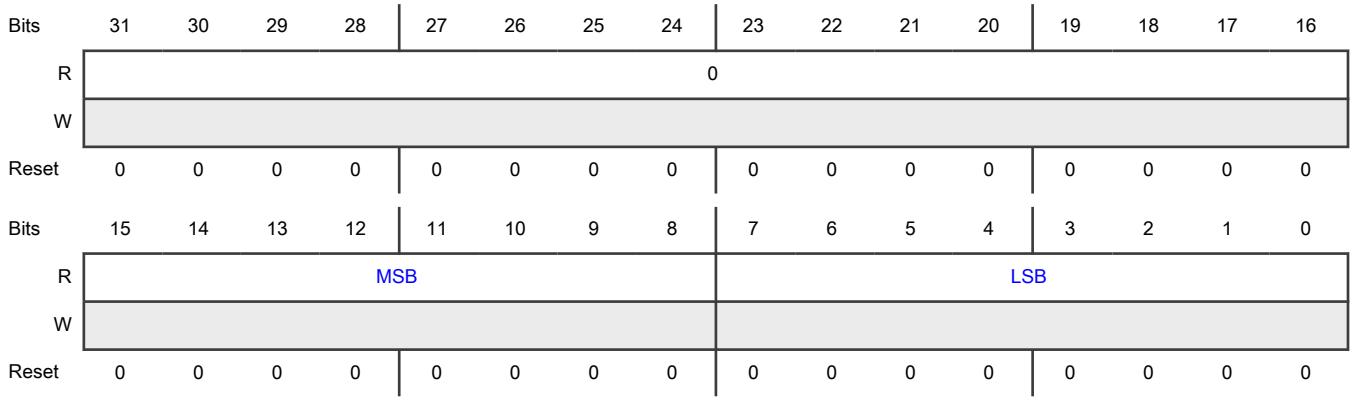
Register	Offset
MRDATAH	C8h

Function

Allows reading a halfword (byte pair) written by the target after an SDR read or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

A halfword must not be read unless there are at least two bytes of data waiting, as indicated by the receive FIFO level trigger or MDATACTRL[RXCOUNT].

Diagram



Fields

Field	Function
31-16 —	Reserved
15-8 MSB	High Byte Represents the second byte read from the controller (and written by the target).
7-0 LSB	Low Byte Represents the first byte read from the controller (and written by the target).

69.7.45 Controller Write Byte Data 1 (to Bus) (MWDATAB1)

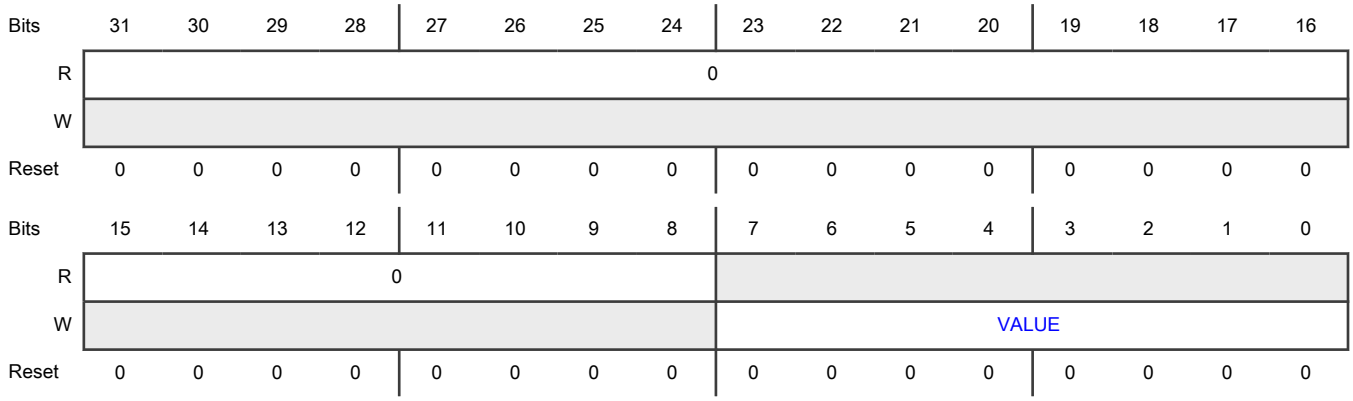
Offset

Register	Offset
MWDATAB1	CCh

Function

Allows writing bytes to send onto the bus, and is intended for DMAs that do not clear the upper bits of the word. This register does not have the END bits and is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 VALUE	Value Represents the byte to write out. The I3C module computes the parity for I3C, or manages the ACK or NACK for I2C.

69.7.46 Controller Write Halfword Data (to Bus) (MWDATAH1)

Offset

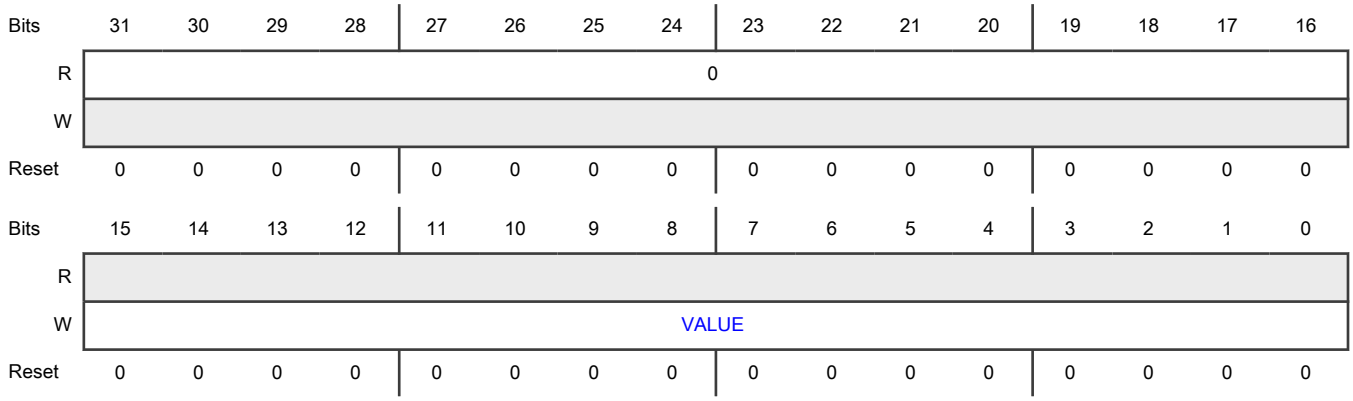
Register	Offset
MWDATAH1	CCh

Function

Allows writing a byte pair (halfword) to send onto the bus, and is intended for DMAs that do not clear the upper bits of the word. It does not have the END bits as present in Controller Write Data Halfword (MWDATAH). This register is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

MWDATAH1 shares the same address as MWDATAB1. MDMACTRL[DMAWIDTH] determines which one is set.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 VALUE	Value Represents the byte to write out. The I3C module computes the parity for I3C, or manages the ACK or NACK for I2C.

69.7.47 Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL)

Offset

Register	Offset
MWMSG_SDR_CONTROL	D0h

Function

Allows setting up and writing 16-bit words in Single Data Rate (SDR) mode. The MWMSG_SDR_ register is modal and has two modes—control and data:

- For the first write to set up a new message, this register functions as the MWSMSG_SDR_CONTROL register.
- For subsequent writes, this register functions as MWMSG_SDR_DATA.
- The control information is not pipelined. You must write to control information registers for a start and do not write to them until data is sent.

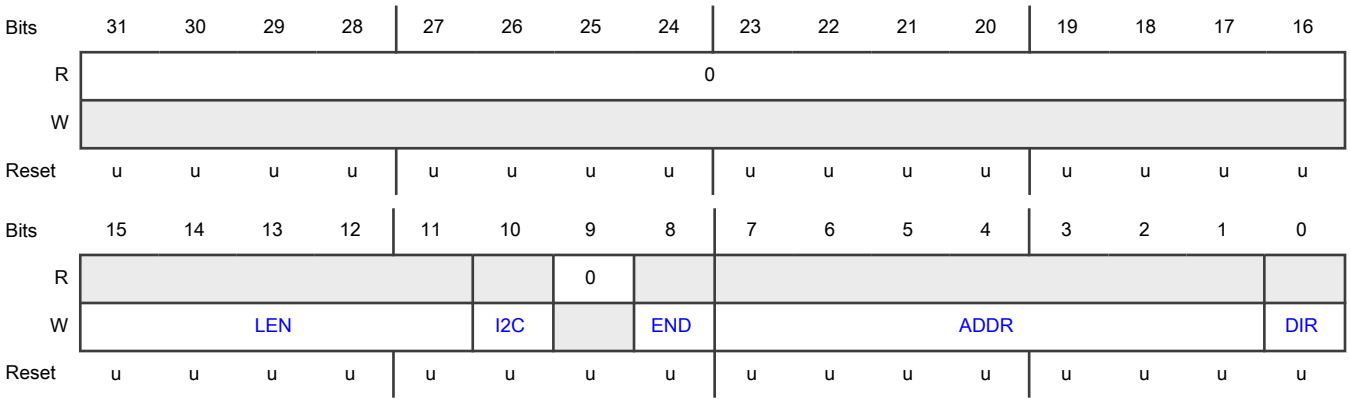
When not in the middle of a message, the MWMSG_SDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_SDR_DATA register is used until the length (see the LEN field) counts down, or until data with END = 1 is used.

The MWMSG_SDR_CONTROL register is written with the 16-bit control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 2 (MSGSDR) and MSTATUS[BETWEEN] = 0, the register (at this offset address) functions as the MWMSG_SDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_SDR_CONTRL register, as long as MSTATUS[STATE] is not in another mode.

The control word contains the byte length (6-bit), address, direction, and how it ends (stop, ready for next, continuation with more length). If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt occurs. In that case, the message is restarted.

The MWMSG_SDR_CONTROL and MWMSG_SDR_DATA registers are only targeted for DMA operations, but the processor can also write to the MWMSG_SDR_CONTROL register (instead of using MCTRL and MxDATAB).

Diagram



Fields

Field	Function
31-16 —	Reserved
15-11 LEN	Length Indicates the byte length of the message. If LEN = 0, then only END is used (and it does not use the address if stopped). LEN = 1 must not be used. The minimal LEN size is 2 bytes (LEN = 2).
10 I2C	I2C Specifies whether the message is I2C or I3C. 0b - I3C message 1b - I2C message
9 —	Reserved
8 END	End of SDR Message Indicates the end of SDR message. MSTATUS[COMPLETE] = 1 when done. The end can happen: • Before LEN bytes are read in I3C mode, if a target ends sooner. • Before LEN bytes are written in I2C mode, if a target NACKs. If this field is 0, the SDR message ends waiting for a new SDR message (issues a Repeated START for a new message). If this field is 1, the SDR message ends at the STOP.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - Not the end 1b - End
7-1 ADDR	Address Contains the address to be written.
0 DIR	Direction 0b - Write 1b - Read

69.7.48 Controller Write Message Data in SDR mode (MWMSG_SDR_DATA)

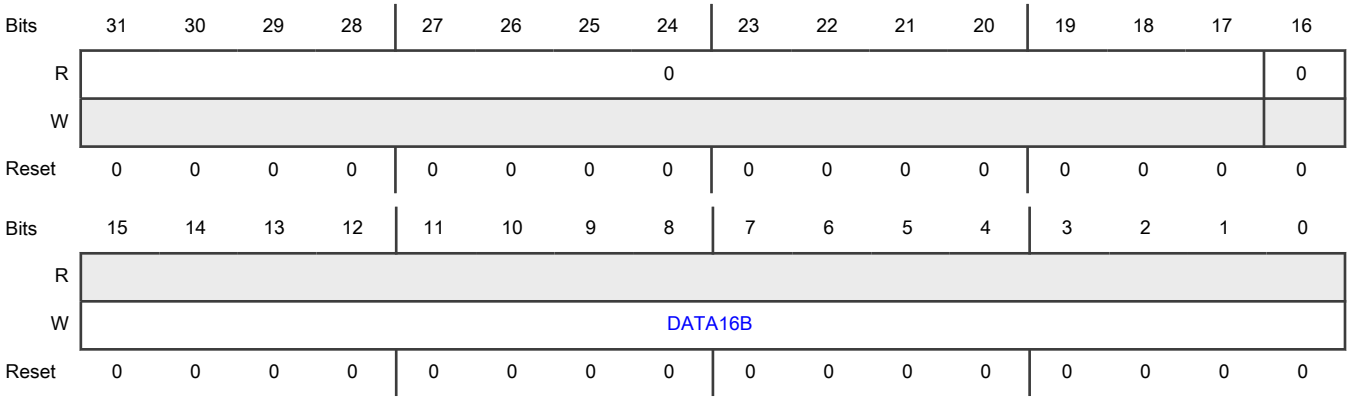
Offset

Register	Offset
MWMSG_SDR_DATA	D0h

Function

Contains the 16-bit word to be written in Single Data Rate (SDR) mode. This register functions in a way similar to MWMSG_SDR_CONTROL.

Diagram



Fields

Field	Function
31-17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
16 —	Reserved
15-0 DATA16B	Data

69.7.49 Controller Read Message in SDR mode (MRMSG_SDR)

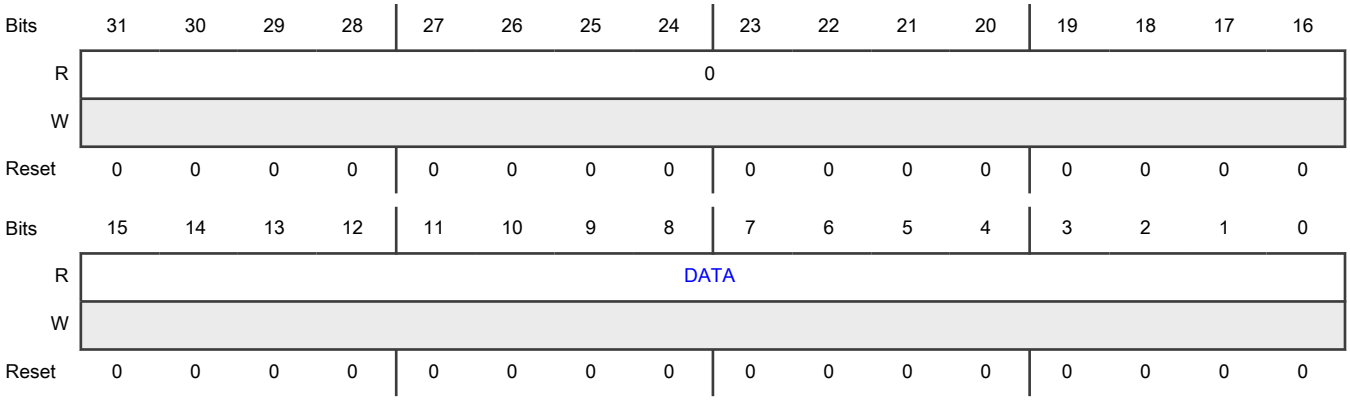
Offset

Register	Offset
MRMSG_SDR	D4h

Function

Allows reading 16-bit words from a target in SDR Message mode. The MRMSG_SDR register is used to read 16-bit words from an active message started with MWMSG_SDR. These words are intended to be read by DMA.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Data Contains the 16-bit word read from the target: • If the length (LEN) is an odd number, the upper byte is 0.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	• If the target ends before LEN is finished, the module treats the read as completed. • If the target is not done before LEN is finished and END is not a continuation, the read is terminated.

69.7.50 Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL)

Offset

Register	Offset
MWMSG_DDR_CONTROL	D8h

Function

Allows setting up and writing 16-bit words in DDR mode. The register has three modes:

- The first write to set up a new message functions as the MWSMSG_DDR_CONTROL register.
- The second write contains the functions as shown in the MWMSG_DDR_CONTROL2 register.
- For subsequent writes, this register functions in a way similar to Controller Write Message Data in DDR mode (MWMSG_DDR_DATA).
- The control information is not pipelined. You must write to control information registers for a start and do not write to them until data is sent.

When not in the middle of a message, the MWMSG_DDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_DDR_DATA register is used until the length (see LEN field) counts down or until data with END = 1 is used.

The MWMSG_DDR_CONTROL register is written with the 16-bit control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 4 (MSGDDR) and MSTATUS[BETWEEN] = 0, then the register (at this offset address) functions as the MWMSG_DDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_DDR_CONTROL register, as long as MSTATUS[STATE] is not in another mode.

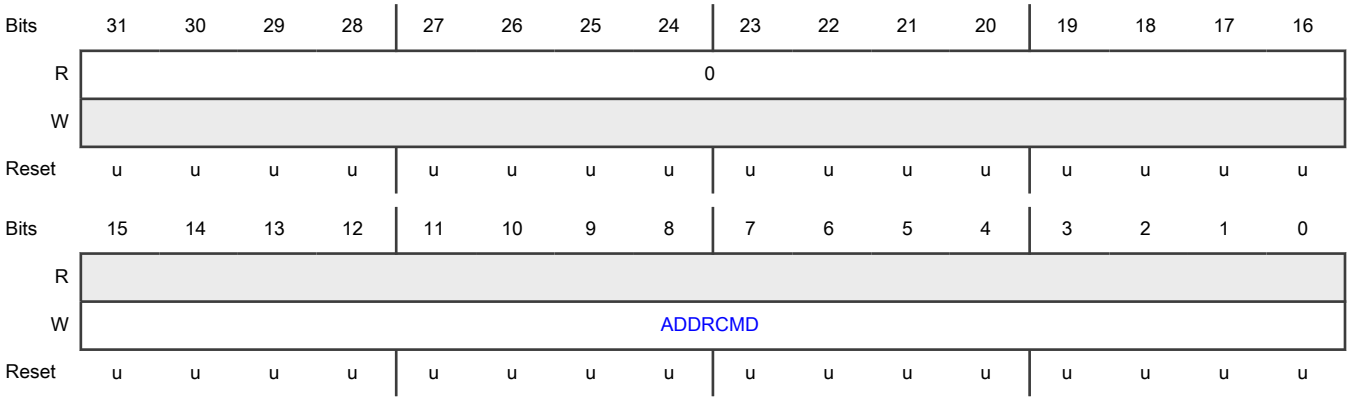
The main control word contains the 16-bit word length and how it ends (stop, ready for next, continuation with more length). Then the command word contains the command and address for read or write. If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt occurs. In that case, the message is restarted.

The MWMSG_DDR_CONTROL, MWMSG_DDR_CONTROL2, and MWMSG_DDR_DATA registers are only targeted for DMA operations, but the processor can also write to the MWMSG_DDR_CONTROL register (instead of using MCTRL and MxDATAB).

NOTE

The module handles preamble, parity, and CRC.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 ADDRCMD	Address Command Indicates the first data write after control write, with LEN ≠ 0. This is formatted as: • Bits 15:9: target address to read or write • Bit 8: reserved, must be 0 • Bit 7: 1 if read, 0 if write • Bits 6:0: CMD as 7-bit value, always for controller

69.7.51 Controller Write Message in DDR Mode Control 2 (MWMSG_DDR_CONTROL2)

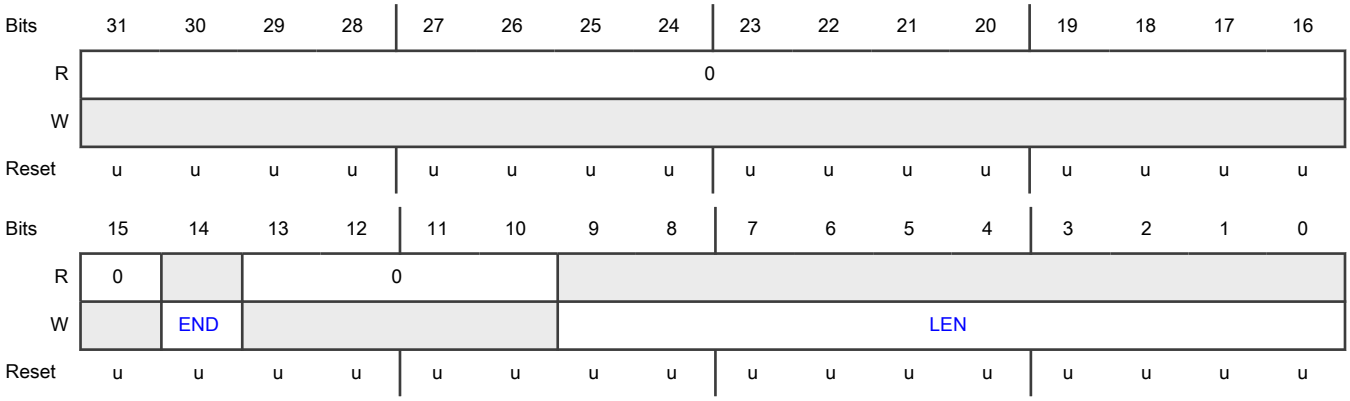
Offset

Register	Offset
MWMSG_DDR_CONTROL2	D8h

Function

Contains the second control word instructions with the length of message and end.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 —	Reserved
14 END	End of Message Indicates the end of DDR message. MSTATUS[COMPLETE] becomes 1 when done. The end can happen before LEN bytes are read if the target ends sooner. If this field is 0, DDR message ends waiting for a new DDR message (issues an HDR restart for the new message). If this field is 1, DDR message ends on HDR exit. 0b - Not the end 1b - End
13-10 —	Reserved
9-0 LEN	Length of Message Contains the length of the message (including the command) in halfwords, up to 2046 bytes. If LEN = 0, then only END is applied: • For reads, + 1 for the CRC. For example, to read 4 bytes (2 halfwords), use 1 + 2 + 1 for CMD + 2 halfwords + CRC. • For writes, LEN is the number of halves of data bytes + 1 (for command).

69.7.52 Controller Write Message Data in DDR mode (MWMSG_DDR_DATA)

Offset

Register	Offset
MWMSG_DDR_DATA	D8h

Function

Contains the 16-bit word to be written in DDR mode. This register functions in a way similar to Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA16B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15-0 DATA16B	Data

69.7.53 Controller Read Message in DDR mode (MRMSG_DDR)

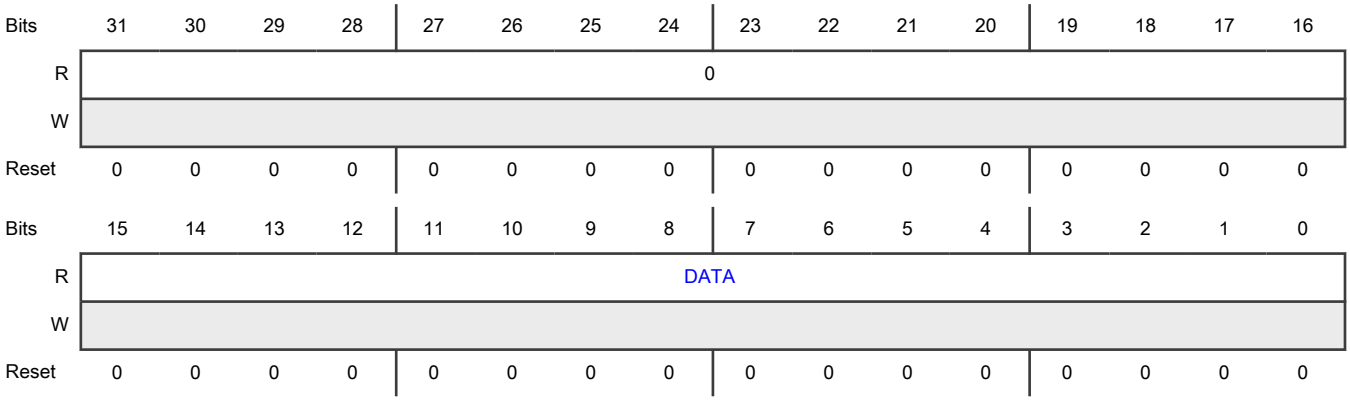
Offset

Register	Offset
MRMSG_DDR	DCh

Function

Allows reading 16-bit words from a target in DDR Message mode from an active message started with MWMSG_DDR. These words are intended to be read by DMA.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Data Contains the 16-bit word read from a target. The first byte is the LSB, and is in DATA[7:0]. The second byte is the MSB, and is in DATA[15:8]: • If the target ends before the entire length of the message (MWMSG_DDR[LEN]) is read, the module considers the DATA read as completed. In I3C mode, the target can indicate the end of message (the last byte). Otherwise, the controller terminates the message if the message is more than the controller can accept. • If the target has not yet finished sending DATA before the entire length of the message (MWMSG_DDR[LEN]) is read and END is not a continuation, the DATA read is terminated.

69.7.54 Controller Dynamic Address (MDYNADDR)

Offset

Register	Offset
MDYNADDR	E4h

Function

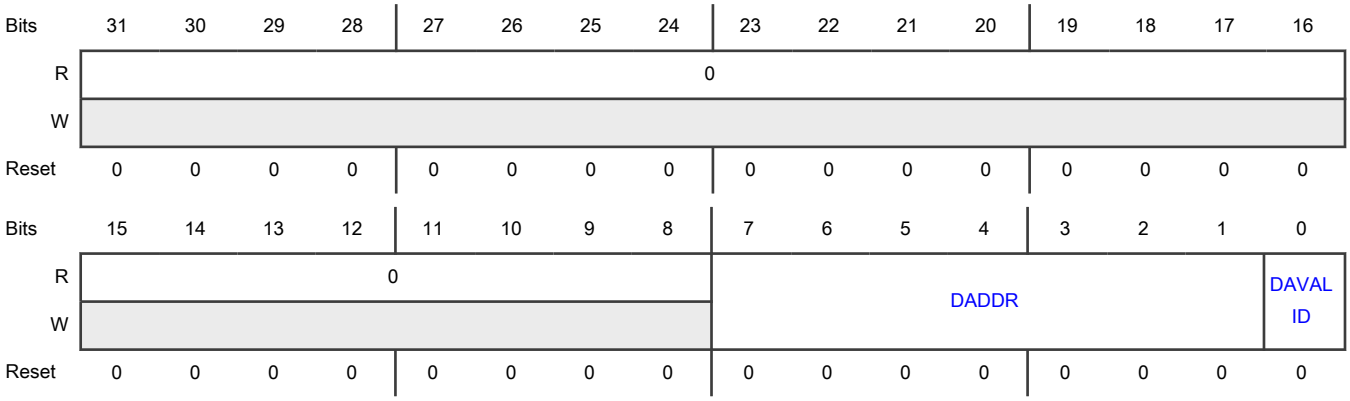
Allows the I3C module to write its own dynamic address (DA) when the module changes from Controller mode to Target mode.

If this device is the main controller (the controller during bus initialization), then the device may use this mechanism to assign itself its DA. When the device hands off control to a secondary controller, it becomes a target itself. This DA must be written before switching to Target mode and must not be changed once in Target mode (it is not clock-safe to do so). It must be written with a valid address value in DADDR if DAVALID = 1.

NOTE

The main controller also uses DEFSLVS CCC to define the target addresses, including itself. This mechanism is how secondary controllers know this address. If the controller is not the main controller, then this mechanism must not be used.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-1 DADDR	Dynamic Address Contains the assigned dynamic address when DAVALID = 1.
0 DAVALID	Dynamic Address Valid 0b - No valid DA assigned 1b - Valid DA assigned

69.7.55 Map Feature Control 0 (SMAPCTRL0)

Offset

Register	Offset
SMAPCTRL0	11Ch

Function

Provides map feature control.

The SMAPCTRL*n* registers are named SMAPCTRL0, SMAPCTRL1, and so on based on the number of mapped addresses. MAPCTRL0 represents the primary DA or SA with SMAPCTRL1 onwards being the mapped addresses.

The features of the SMAPCTRL*n* registers depend on configuration. SMAPCTRL0 acts differently, as described in this chapter.

In general, this mechanism is intended to replace the DYNADDR register for all MAP-related uses.

When using the Auto-MAP and DASA or AASA, the slot is changed from SA to DA. If the controller then issues RSTDAA, the application must rewrite the static addresses and enable them because they are marked disabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CAUSE			DA							ENA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15-11 —	Reserved
10-8 CAUSE	Cause Indicates the cause of the most recent DA assignment, which lead to SSTATUS[DACHG] interrupt. This field has MAP enabled. If this field is 100b (auto MAP change happened last), the change may have changed this DA as well (for example, ENTDAA and SETAASA), but at least one MAP entry automatically changed after that. 000b - No information (this value occurs when not configured to write DA) 001b - Set using ENTDAA 010b - Set using SETDASA, SETAASA, or SETNEWDA 011b - Cleared using RSTDAA 100b - Auto MAP change happened last All other values are reserved.
7-1 DA	Dynamic Address Contains primary DA when ENA = 1. When ENA = 0, static address is used (but not described here). This field has MAP enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Function
0 ENA	Enable Primary Dynamic Address Indicates whether the MAP is enabled. 0b - Disabled 1b - Enabled

69.7.56 Extended IBI Data 1 (IBIEXT1)

Offset

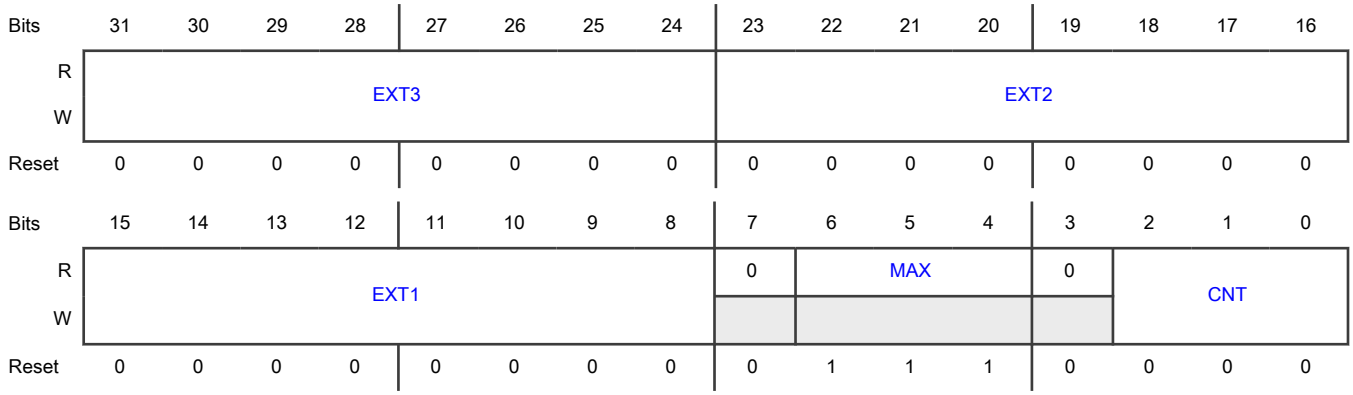
Register	Offset
IBIEXT1	140h

Function

Contains extended IBI data.

The CTRL register is used to submit IBI, CR, and Hot-Join when enabled to do so. If allowed, an IBI may have additional bytes following the mandatory data byte (MDB). Extended IBI data is allowed when SCTRL[EXTDATA] = 1. If allowed, the extra bytes are indicated using these registers.

Diagram



Fields

Field	Function
31-24 EXT3	Extra Byte 3 Contains the third extra byte.
23-16 EXT2	Extra Byte 2 Contains the second extra byte.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15-8 EXT1	Extra Byte 1 Contains the first extra byte.
7 —	Reserved
6-4 MAX	Maximum Indicates the maximum number of extra bytes allowed by configuration. This field is 0 if there are no extra bytes.
3 —	Reserved
2-0 CNT	Count Contains the number of extra bytes beyond the MDB to be used. This field is 0 if there are no extra bytes.

69.7.57 Extended IBI Data 2 (IBIEXT2)

Offset

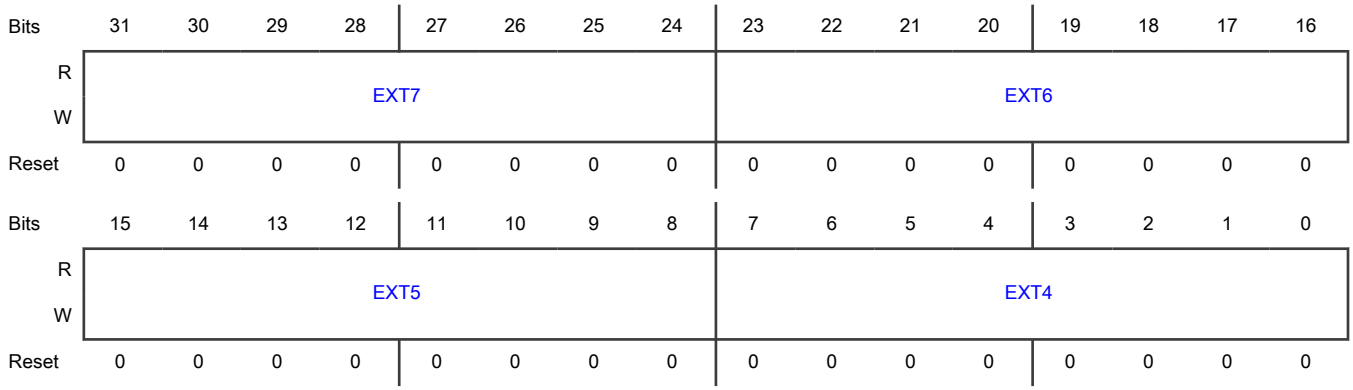
Register	Offset
IBIEXT2	144h

Function

Contains extended IBI data.

The CTRL register is used to submit IBI, CR, and Hot-Join when enabled to do so. If allowed, an IBI may have additional bytes following the mandatory data byte (MDB). Extended IBI data is allowed when SCTRL[EXTDATA] = 1. If allowed, the extra bytes are indicated using these registers.

Diagram



Fields

Field	Function
31-24 EXT7	Extra Byte 7 Contains the seventh extra byte.
23-16 EXT6	Extra Byte 6 Contains the sixth extra byte.
15-8 EXT5	Extra Byte 5 Contains the fifth extra byte.
7-0 EXT4	Extra Byte 4 Contains the forth extra byte.

69.7.58 Target Module ID (SID)

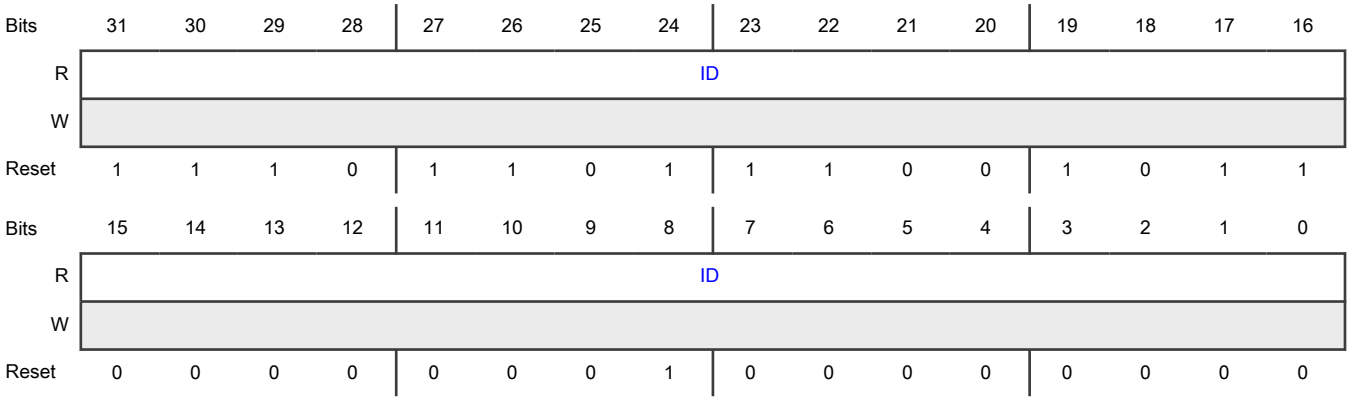
Offset

Register	Offset
SID	FFCh

Function

Allows software to detect the module and its version information, if BlockID is enabled.

Diagram



Fields

Field	Function
31-0 ID	ID Indicates the ID. The ID meaning is specific to each use of the I3C module. ID = EDCB0100h.

Chapter 70
Input Multiplexing (INPUTMUX)

70.1 Chip-specific INPUTMUX information

Table 682. Reference links to related information

Topic	Related module	Reference
Full description	INPUTMUX	INPUTMUX
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Signal multiplexing	IOPCTL	External signals and pin multiplexing IO PAD controller

70.1.1 Module instances

This device contains two instances of the INPUTMUX module.

Table 683. INPUTMUX instances

	INPUTMUX0	INPUTMUX1
Located	VDD2_COMP	VDD1_SENSE
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3

70.2 Overview

INPUTMUX provides signal routing options for internal peripherals. Some peripheral inputs are multiplexed to multiple input sources. The sources can be external pins, interrupts, output signals of other peripherals, or other internal signals.

NOTE

Depending on the package, not all inputs from external pins are available.

70.2.1 Block diagram

Figure 295 shows a generic input multiplexer arrangement with n inputs.

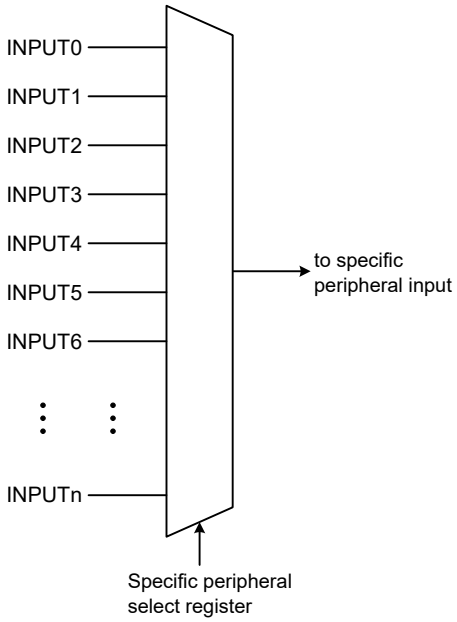


Figure 295. Generic input multiplexing

70.2.2 Features

- Configures the inputs to the SCT module.
- Configures the inputs to the asynchronous CTIMER modules.
- Configures the inputs to the HiFi1 and HiFi4.
- Configures the inputs to the LP_FLEXCOMM and FLEXIO modules.
- Configures the inputs to the EZH-V
- Configures the inputs to the PINT modules.
- Configures the inputs to ADC0 module.
- Configures the inputs to the FREQME module.

70.3 Functional description

The INPUTMUX implements a number of input multiplexers that select one of many inputs to be routed to a specific input signal for a given peripheral. This is used to allow user configuration of data paths between internal modules and/or external pins on the device. For every module input (output from the INPUTMUX), there is a register that selects the input to use, where the register name and description provide details on the module input controlled by each register. The input signal/pin options for each of the multiplex are configurable, and can vary from each other. See register descriptions for the details on the input signal/pin options used for each INPUTMUX output in Memory map and register definition.

70.3.1 Clocking

Table 684. INPUTMUX clock

Clock	Description
Module clock (clk_and_reset_pclk)	Functions the module and controls the access to the registers.

After the INPUTMUX is configured, no clock is needed to function the module, you can disable the clocks by:

- Writing 1 to the CLKCTL0_PSCCTL3_CLR[INPUTMUX0] to disable the clock to the INPUTMUX0.
- Writing 1 to the CLKCTL1_PSCCTL1_CLR[INPUTMUX1] to disable the clock to the INPUTMUX1.

70.3.2 Reset

The following table shows the INPUTMUX registers reset caused by RSTCTL module:

Table 685. INPUTMUX related reset

Write 1 to	Reset
RSTCTL0_PRSTCTL4[INPUTMUX0] or RSTCTL0_PRSTCTL4_SET[INPUTMUX0]	• INPUTMUX0_HiFi4_INTERRUPT0-INPUTMUX0_HiFi4_INTERRUPT26 • INPUTMUX0_LP_FLEXCOMM0_TRIG-INPUTMUX0_FLEXCOMM13_TRIG • INPUTMUX0_CTIMER0_CAP0-INPUTMUX0_CTIMER4_CAP3 • INPUTMUX0_CTIMER0_TRIG-INPUTMUX0_CTIMER4_TRIG • INPUTMUX0_FREQME_TRIG0-INPUTMUX0_FREQME_TRIG1 • INPUTMUX0_EZHV_TRIG0-INPUTMUX0_EZHV_TRIG7 • INPUTMUX0_FLEXIO_TRIG0-INPUTMUX0_FLEXIO_TRIG3
RSTCTL3_PRSTCTL1[INPUTMUX1] or RSTCTL3_PRSTCTL1_SET[INPUTMUX1]	• INPUTMUX1_HiFi1_INTERRUPT0-INPUTMUX1_HiFi1_INTERRUPT26 • INPUTMUX1_LP_FLEXCOMM17_TRIG-INPUTMUX1_LP_FLEXCOMM20_TRIG • INPUTMUX1_ADC0_TRIG0-INPUTMUX1_ADC0_TRIG1 • INPUTMUX1_CTIMER5_CAP0-INPUTMUX1_CTIMER7_CAP3 • INPUTMUX1_CTIMER5_TRIG-INPUTMUX1_CTIMER7_TRIG
RSTCTL0_PRSTCTL2[SCT0] or RSTCTL0_PRSTCTL2_SET[SCT0]	• SCT0 module • INPUTMUX0_SCT0_TRIG0-INPUTMUX0_SCT0_TRIG6
RSTCTL0_PRSTCTL4[PINT0] or RSTCTL0_PRSTCTL4_SET[PINT0]	• PINT0 module • INPUTMUX0_PINT0_TRIG0-INPUTMUX0_PINT0_TRIG7
RSTCTL1_PRSTCTL0[PINT1] or RSTCTL1_PRSTCTL0_SET[PINT1]	• PINT1 module • INPUTMUX1_PINT1_TRIG0-INPUTMUX1_PINT1_TRIG3

70.3.3 Interrupts

This module has no interrupts.

70.4 External signals

The INPUTMUX has no dedicated pins. Multiplexer inputs from external pins work independently of any other function assigned to the pin as long as no analog function is enabled.

Table 686. INPUTMUX pin description

Pins	Peripheral	Section
Any existing pin on PIO0, PIO1, PIO8 and PIO10	Pin interrupts 0 to 7	See INPUTMUX
PIO0_28	FREQME	See INPUTMUX
SCT_GPIN[0:7] pin functions selected from IOPCTL register	SCT	See SCT
CTIMER_C_INP [0:15] pin functions selected from IOPCTL register	CTIMER/FLEXIO/ LP_FLEXCOMM	See CTIMER
CTIMER_S_INP [0:9] pin functions selected from IOPCTL register	CTIMER/FLEXIO/ LP_FLEXCOMM	See CTIMER

70.5 Initialization

This module has no initialization considerations.

70.6 Memory map and register definition

70.6.1 INPUTMUX0 register descriptions

70.6.1.1 INPUTMUX0 memory map

INPUTMUX0 base address: 4002_6000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 18h	SCT0 Input Connections (SCT0_TRIG0 - SCT0_TRIG6)	32	RW	0000_001Fh
100h - 11Ch	PINT0 Input Connections (PINT0_TRIG0 - PINT0_TRIG7)	32	RW	0000_003Fh
140h - 1A8h	HiFi4 Interrupt Connections (HIFI4_INTERRUPT0 - HIFI4_INTERRUPT26)	32	RW	0000_007Fh
200h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM0_TRIG)	32	RW	0000_000Fh
220h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM1_TRIG)	32	RW	0000_000Fh
240h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM2_TRIG)	32	RW	0000_000Fh
260h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM3_TRIG)	32	RW	0000_000Fh
280h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM4_TRIG)	32	RW	0000_000Fh
2A0h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM5_TRIG)	32	RW	0000_000Fh
2C0h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM6_TRIG)	32	RW	0000_000Fh
2E0h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM7_TRIG)	32	RW	0000_000Fh
300h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM8_TRIG)	32	RW	0000_000Fh
320h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM9_TRIG)	32	RW	0000_000Fh

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
340h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM10_TRIG)	32	RW	0000_000Fh
360h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM11_TRIG)	32	RW	0000_000Fh
380h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM12_TRIG)	32	RW	0000_000Fh
3A0h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM13_TRIG)	32	RW	0000_000Fh
600h	CTIMER Input Connections (CTIMER0_CAP0)	32	RW	0000_001Fh
604h	CTIMER Input Connections (CTIMER0_CAP1)	32	RW	0000_001Fh
608h	CTIMER Input Connections (CTIMER0_CAP2)	32	RW	0000_001Fh
60Ch	CTIMER Input Connections (CTIMER0_CAP3)	32	RW	0000_001Fh
610h	CTIMER Input Connections (CTIMER0_TRIG)	32	RW	0000_001Fh
620h	CTIMER Input Connections (CTIMER1_CAP0)	32	RW	0000_001Fh
624h	CTIMER Input Connections (CTIMER1_CAP1)	32	RW	0000_001Fh
628h	CTIMER Input Connections (CTIMER1_CAP2)	32	RW	0000_001Fh
62Ch	CTIMER Input Connections (CTIMER1_CAP3)	32	RW	0000_001Fh
630h	CTIMER Input Connections (CTIMER1_TRIG)	32	RW	0000_001Fh
640h	CTIMER Input Connections (CTIMER2_CAP0)	32	RW	0000_001Fh
644h	CTIMER Input Connections (CTIMER2_CAP1)	32	RW	0000_001Fh
648h	CTIMER Input Connections (CTIMER2_CAP2)	32	RW	0000_001Fh
64Ch	CTIMER Input Connections (CTIMER2_CAP3)	32	RW	0000_001Fh
650h	CTIMER Input Connections (CTIMER2_TRIG)	32	RW	0000_001Fh
660h	CTIMER Input Connections (CTIMER3_CAP0)	32	RW	0000_001Fh
664h	CTIMER Input Connections (CTIMER3_CAP1)	32	RW	0000_001Fh
668h	CTIMER Input Connections (CTIMER3_CAP2)	32	RW	0000_001Fh
66Ch	CTIMER Input Connections (CTIMER3_CAP3)	32	RW	0000_001Fh
670h	CTIMER Input Connections (CTIMER3_TRIG)	32	RW	0000_001Fh
680h	CTIMER Input Connections (CTIMER4_CAP0)	32	RW	0000_001Fh
684h	CTIMER Input Connections (CTIMER4_CAP1)	32	RW	0000_001Fh
688h	CTIMER Input Connections (CTIMER4_CAP2)	32	RW	0000_001Fh
68Ch	CTIMER Input Connections (CTIMER4_CAP3)	32	RW	0000_001Fh
690h	CTIMER Input Connections (CTIMER4_TRIG)	32	RW	0000_001Fh
700h - 704h	FREQME Clock Selection (FREQME_TRIG0 - FREQME_TRIG1)	32	RW	0000_001Fh
720h - 73Ch	EZH-V Input Connections (EZHV_TRIG0 - EZHV_TRIG7)	32	RW	0000_007Fh

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
760h - 76Ch	FLEXIO Input Connections (FLEXIO_TRIG0 - FLEXIO_TRIG3)	32	RW	0000_003Fh

70.6.1.2 SCT0 Input Connections (SCT0_TRIG0 - SCT0_TRIG6)

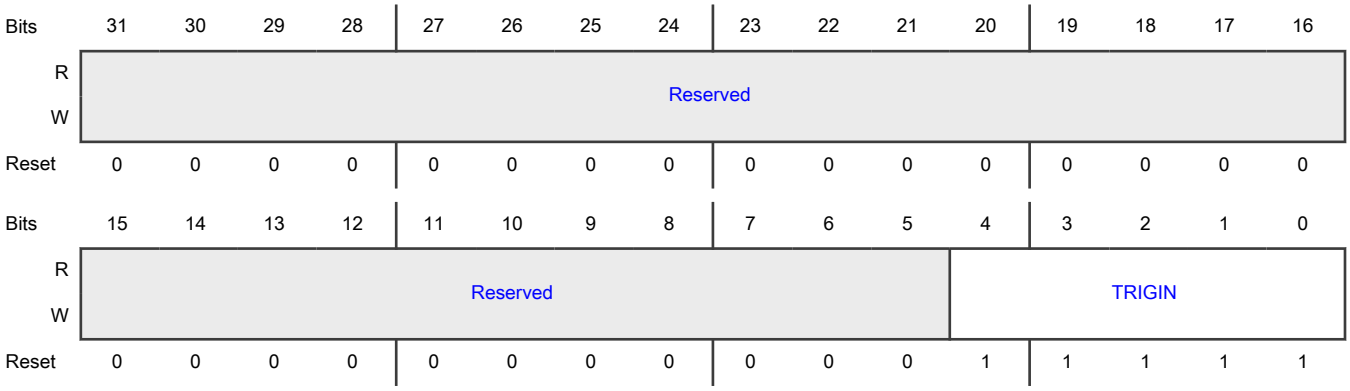
Offset

Register	Offset
SCT0_TRIG0	0h
SCT0_TRIG1	4h
SCT0_TRIG2	8h
SCT0_TRIG3	Ch
SCT0_TRIG4	10h
SCT0_TRIG5	14h
SCT0_TRIG6	18h

Function

Selects the SCT0 trigger inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
4-0 TRIGIN	SCT0 Input Trigger 0_0000b - Selects SCT0_GPIN0 0_0001b - Selects SCT0_GPIN1 0_0010b - Selects SCT0_GPIN2 0_0011b - Selects SCT0_GPIN3 0_0100b - Selects SCT0_GPIN4 0_0101b - Selects SCT0_GPIN5 0_0110b - Selects SCT0_GPIN6 0_0111b - Selects SCT0_GPIN7 0_1000b - Selects CTIMER0_MAT0 0_1001b - Selects CTIMER1_MAT0 0_1010b - Selects CTIMER2_MAT0 0_1011b - Selects CTIMER3_MAT0 0_1100b - Selects CTIMER4_MAT0 0_1101b - Selects USB1 start of frame 0_1110b - Selects PINT0 boolean pattern-match output 0_1111b - Selects USB0 start of frame 1_0000b - Selects ACMP0 CMPO 1_0001b - Selects SAI0_TX_BCLK 1_0010b - Selects SAI0_RX_BCLK 1_0011b - Selects SAI1_TX_BCLK 1_0100b - Selects SAI1_RX_BCLK 1_0101b - Selects SAI2_TX_BCLK 1_0110b - Selects SAI2_RX_BCLK 1_0111b - Selects MCLK 1_1000b - Selects CPU0 TXEV 1_1001b - Selects CPU0 Halted 1_1010b - Selects ADC0 trigger completion flag 0 1_1011b - Selects ADC0 trigger completion flag 1 All other values are reserved.

70.6.1.3 PINT0 Input Connections (PINT0_TRIG0 - PINT0_TRIG7)

Offset

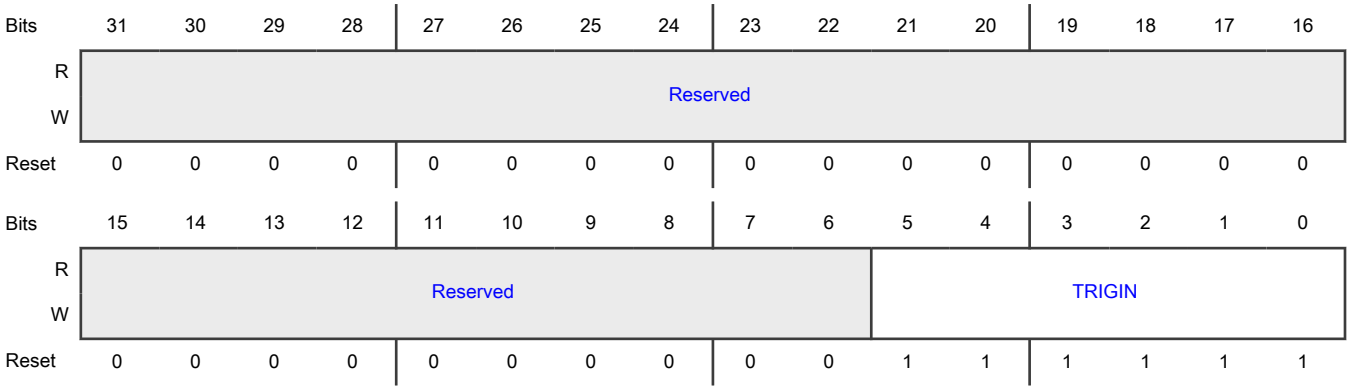
For a = 0 to 7:

Register	Offset
PINT0_TRIGa	100h + (a × 4h)

Function

Selects the PINT0 trigger inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	PINT0 Input Trigger 00_0000b - Selects PIO0_0 00_0001b - Selects PIO0_1 00_0010b - Selects PIO0_2 00_0011b - Selects PIO0_3 00_0100b - Selects PIO0_4 00_0101b - Selects PIO0_5 00_0110b - Selects PIO0_6 00_0111b - Selects PIO0_7 00_1000b - Selects PIO0_8 00_1001b - Selects PIO0_9 00_1010b - Selects PIO0_10 00_1011b - Selects PIO0_11 00_1100b - Selects PIO0_12 00_1101b - Selects PIO0_13

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00_1110b - Selects PIO0_14 00_1111b - Selects PIO0_15 01_0000b - Selects PIO0_16 01_0001b - Selects PIO0_17 01_0010b - Selects PIO0_18 01_0011b - Selects PIO0_19 01_0100b - Selects PIO0_20 01_0101b - Selects PIO0_21 01_0110b - Selects PIO0_22 01_0111b - Selects PIO0_23 01_1000b - Selects PIO0_24 01_1001b - Selects PIO0_25 01_1010b - Selects PIO0_26 01_1011b - Selects PIO0_27 01_1100b - Selects PIO0_28 01_1101b - Selects PIO0_29 01_1110b - Selects PIO0_30 01_1111b - Selects PIO0_31 10_0000b - Selects PIO1_0 10_0001b - Selects PIO1_1 10_0010b - Selects PIO1_2 10_0011b - Selects PIO1_3 10_0100b - Selects PIO1_4 10_0101b - Selects PIO1_5 10_0110b - Selects PIO1_6 10_0111b - Selects PIO1_7 10_1000b - Selects PIO1_8 10_1001b - Selects PIO1_9 10_1010b - Selects PIO1_10 10_1011b - Selects PIO1_11 10_1100b - Selects PIO1_12 10_1101b - Selects PIO1_13 10_1110b - Selects PIO1_14

Table continues on the next page...

Table continued from the previous page...

Field	Function
	10_1111b - Selects PIO1_15 11_0000b - Selects PIO1_16 11_0001b - Selects PIO1_17 11_0010b - Selects PIO1_18 11_0011b - Selects PIO1_19 All other values are reserved.

70.6.1.4 HiFi4 Interrupt Connections (HIFI4_INTERRUPT0 - HIFI4_INTERRUPT26)

Offset

For a = 0 to 26:

Register	Offset
HIFI4_INTERRUPTa	140h + (a × 4h)

Function

Selects the HiFi4 interrupt inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TRIGIN							
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Fields

Field	Function
31-7 —	Reserved
6-0 TRIGIN	HiFi4 Input Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	000_0000b - Selects LP_FLEXCOMM0 interrupt 000_0001b - Selects LP_FLEXCOMM1 interrupt 000_0010b - Selects LP_FLEXCOMM2 interrupt 000_0011b - Selects LP_FLEXCOMM3 interrupt 000_0100b - Selects LP_FLEXCOMM4 interrupt 000_0101b - Selects LP_FLEXCOMM5 interrupt 000_0110b - Selects LP_FLEXCOMM6 interrupt 000_0111b - Selects LP_FLEXCOMM7 interrupt 000_1000b - Selects LP_FLEXCOMM8 interrupt 000_1001b - Selects LP_FLEXCOMM9 interrupt 000_1010b - Selects LP_FLEXCOMM10 interrupt 000_1011b - Selects LP_FLEXCOMM11 interrupt 000_1100b - Selects LP_FLEXCOMM12 interrupt 000_1101b - Selects LP_FLEXCOMM13 interrupt 000_1110b - Selects MU2_MUA interrupt for HiFi4 to CPU1 000_1111b - Selects MU4_MUB interrupt for HiFi4 to CPU0 001_0000b - Selects WWDT0 interrupt 001_0001b - Selects WWDT1 interrupt 001_0010b - Selects UTICK0 interrupt 001_0011b - Selects LPSPI14 interrupt 001_0100b - Selects MRT0 Ored interrupt 001_0101b - Selects OSTIMER event timer for HiFi4 wakeup/interrupt 001_0110b - Selects CTIMER0 interrupt 001_0111b - Selects CTIMER1 interrupt 001_1000b - Selects CTIMER2 interrupt 001_1001b - Selects CTIMER3 interrupt 001_1010b - Selects CTIMER4 interrupt 001_1011b - Selects RTC0 alarm interrupt to computer domain 001_1100b - Selects RTC_SS0 wakeup interrupt to compute domain 001_1101b - Selects I3C0 interrupt 001_1110b - Selects I3C1 interrupt 001_1111b - Selects MICFIL interrupt for read data or error 010_0000b - Selects MICFIL hardware voice activity detector interrupt or error interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010_0001b - Selects LCDIF interrupt 010_0010b - Selects VGPU interrupt from graphics core (domain HCLK) 010_0011b - Selects EZH-V interrupt 010_0100b - Selects FLEXIO interrupt 010_0101b - Selects LPSPI16 interrupt 010_0110b - Selects eDMA0 channel 0 interrupt 010_0111b - Selects eDMA0 channel 1 interrupt 010_1000b - Selects eDMA0 channel 2 interrupt 010_1001b - Selects eDMA0 channel 3 interrupt 010_1010b - Selects eDMA0 channel 4 interrupt 010_1011b - Selects eDMA0 channel 5 interrupt 010_1100b - Selects eDMA0 channel 6 interrupt 010_1101b - Selects eDMA0 channel 7 interrupt 010_1110b - Selects eDMA0 channel 8 interrupt 010_1111b - Selects eDMA0 channel 9 interrupt 011_0000b - Selects eDMA0 channel 10 interrupt 011_0001b - Selects eDMA0 channel 11 interrupt 011_0010b - Selects eDMA0 channel 12 interrupt 011_0011b - Selects eDMA0 channel 13 interrupt 011_0100b - Selects eDMA0 channel 14 interrupt 011_0101b - Selects eDMA0 channel 15 interrupt 011_0110b - Selects eDMA1 channel 0 interrupt 011_0111b - Selects eDMA1 channel 1 interrupt 011_1000b - Selects eDMA1 channel 2 interrupt 011_1001b - Selects eDMA1 channel 3 interrupt 011_1010b - Selects eDMA1 channel 4 interrupt 011_1011b - Selects eDMA1 channel 5 interrupt 011_1100b - Selects eDMA1 channel 6 interrupt 011_1101b - Selects eDMA1 channel 7 interrupt 011_1110b - Selects eDMA1 channel 8 interrupt 011_1111b - Selects eDMA1 channel 9 interrupt 100_0000b - Selects eDMA1 channel 10 interrupt 100_0001b - Selects eDMA1 channel 11 interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	100_0010b - Selects eDMA1 channel 12 interrupt 100_0011b - Selects eDMA1 channel 13 interrupt 100_0100b - Selects eDMA1 channel 14 interrupt 100_0101b - Selects eDMA1 channel 15 interrupt 100_0110b - Selects GPIO0 channel 0 interrupt 100_0111b - Selects GPIO0 channel 1 interrupt 100_1000b - Selects GPIO1 channel 0 interrupt 100_1001b - Selects GPIO1 channel 1 interrupt 100_1010b - Selects GPIO2 channel 0 interrupt 100_1011b - Selects GPIO2 channel 1 interrupt 100_1100b - Selects GPIO3 channel 0 interrupt 100_1101b - Selects GPIO3 channel 1 interrupt 100_1110b - Selects GPIO4 channel 0 interrupt 100_1111b - Selects GPIO4 channel 1 interrupt 101_0000b - Selects GPIO5 channel 0 interrupt 101_0001b - Selects GPIO5 channel 1 interrupt 101_0010b - Selects GPIO6 channel 0 interrupt 101_0011b - Selects GPIO6 channel 1 interrupt 101_0100b - Selects GPIO7 channel 0 interrupt 101_0101b - Selects GPIO7 channel 1 interrupt 101_0110b - Selects PINT0 interrupt 0 101_0111b - Selects PINT0 interrupt 1 101_1000b - Selects PINT0 interrupt 2 101_1001b - Selects PINT0 interrupt 3 101_1010b - Selects PINT0 interrupt 4 101_1011b - Selects PINT0 interrupt 5 101_1100b - Selects PINT0 interrupt 6 101_1101b - Selects PINT0 interrupt 7 101_1110b - Selects SAI0 TX/RX interrupt 101_1111b - Selects SAI1 TX/RX interrupt 110_0000b - Selects SAI2 TX/RX interrupt 110_0010b - Selects XSPI0 Ored interrupt and flash error interrupt. 110_0011b - Selects XSPI1 Ored interrupt and flash error interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	110_0100b - Selects XSPI2 Ored interrupt and flash error interrupt. 110_0101b - Selects JPGDECWRP Ored context 0-3 interrupt 110_0110b - Selects PNGDEC interrupt 110_0111b - Selects NPU interrupt 110_1000b - Selects PMC0 interrupt All other values are reserved.

70.6.1.5 LP_FLEXCOMM Input Connections (LP_FLEXCOMM0_TRIG - LP_FLEXCOMM13_TRIG)

Offset

For a = 0 to 13:

Register	Offset
LP_FLEXCOMMa_TRIG	200h + (a × 20h)

Function

Selects the LP_FLEXCOMM trigger inputs.

NOTE

When TRIGIN field is configured as 1100b to select PINT0 boolean pattern-match output as input trigger and this LP_FLEXCOMM is used as LPUART, the related CFG field of PINT0.PMCFG register must not be configured as 111b (Event).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRIGIN			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Function
31-4 —	Reserved
3-0 TRIGIN	LP_FLEXCOMM Input Trigger 0000b - Selects CTIMER_C_INP0 0001b - Selects CTIMER_C_INP1 0010b - Selects CTIMER_C_INP2 0011b - Selects CTIMER_C_INP3 0100b - Selects SCT0_OUT3 0101b - Selects SCT0_OUT6 0110b - Selects SCT0_OUT9 0111b - Selects CTIMER0_MAT3 1000b - Selects CTIMER1_MAT3 1001b - Selects CTIMER2_MAT3 1010b - Selects CTIMER3_MAT3 1011b - Selects CTIMER4_MAT3 1100b - Selects PINT0 boolean pattern-match output 1101b - Selects ACMP0 CMPO 1110b - Selects RTC_SS0 interrupt All other values are reserved.

70.6.1.6 CTIMER Input Connections (CTIMER0_CAP0 - CTIMER4_CAP3)

Offset

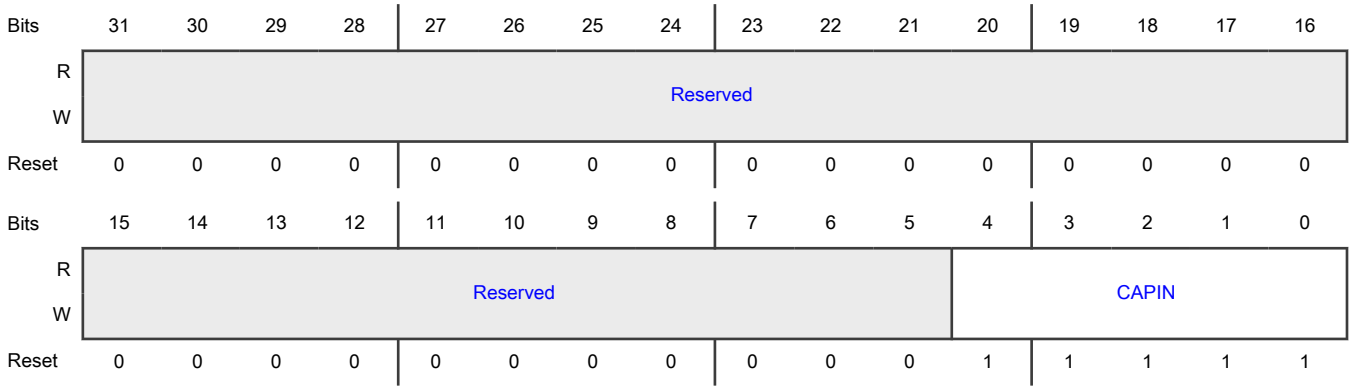
For a = 0 to 4; b = 0 to 3:

Register	Offset
CTIMERa_CAPb	600h + (a × 20h) + (b × 4h)

Function

Selects the CTIMER capture inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 CAPIN	CTIMER input capture 0_0000b - Selects CTIMER_C_INP0 0_0001b - Selects CTIMER_C_INP1 0_0010b - Selects CTIMER_C_INP2 0_0011b - Selects CTIMER_C_INP3 0_0100b - Selects CTIMER_C_INP4 0_0101b - Selects CTIMER_C_INP5 0_0110b - Selects CTIMER_C_INP6 0_0111b - Selects CTIMER_C_INP7 0_1000b - Selects CTIMER_C_INP8 0_1001b - Selects CTIMER_C_INP9 0_1010b - Selects CTIMER_C_INP10 0_1011b - Selects CTIMER_C_INP11 0_1100b - Selects CTIMER_C_INP12 0_1101b - Selects CTIMER_C_INP13 0_1110b - Selects CTIMER_C_INP14 0_1111b - Selects CTIMER_C_INP15 1_0000b - Selects SAI0_TX_SYNC 1_0001b - Selects SAI0_RX_SYNC 1_0010b - Selects SAI1_TX_SYNC 1_0011b - Selects SAI1_RX_SYNC

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1_0100b - Selects SAI2_TX_SYNC 1_0101b - Selects SAI2_RX_SYNC 1_0110b - Selects USB0 start of frame 1_0111b - Selects USB1 start of frame 1_1000b - Selects ACMP0 CMPO 1_1001b - Selects ADC0 trigger completion flag 0 1_1010b - Selects ADC0 trigger completion flag 1 1_1011b - Selects ITRC TMPR_OUT[0] 1_1100b - Selects ITRC TMPR_OUT[1] All other values are reserved.

70.6.1.7 CTIMER Input Connections (CTIMER0_TRIG - CTIMER4_TRIG)

Offset

Register	Offset
CTIMER0_TRIG	610h
CTIMER1_TRIG	630h
CTIMER2_TRIG	650h
CTIMER3_TRIG	670h
CTIMER4_TRIG	690h

Function
Selects the CTIMER trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRIGIN			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Function
31-5 —	Reserved
4-0 TRIGIN	CTIMER Input Trigger 0_0000b - Selects CTIMER_C_INP0 0_0001b - Selects CTIMER_C_INP1 0_0010b - Selects CTIMER_C_INP2 0_0011b - Selects CTIMER_C_INP3 0_0100b - Selects CTIMER_C_INP4 0_0101b - Selects CTIMER_C_INP5 0_0110b - Selects CTIMER_C_INP6 0_0111b - Selects CTIMER_C_INP7 0_1000b - Selects CTIMER_C_INP8 0_1001b - Selects CTIMER_C_INP9 0_1010b - Selects CTIMER_C_INP10 0_1011b - Selects CTIMER_C_INP11 0_1100b - Selects CTIMER_C_INP12 0_1101b - Selects CTIMER_C_INP13 0_1110b - Selects CTIMER_C_INP14 0_1111b - Selects CTIMER_C_INP15 1_0000b - Selects SAI0_TX_SYNC 1_0001b - Selects SAI0_RX_SYNC 1_0010b - Selects SAI1_TX_SYNC 1_0011b - Selects SAI1_RX_SYNC 1_0100b - Selects SAI2_TX_SYNC 1_0101b - Selects SAI2_RX_SYNC 1_0110b - Selects USB0 start of frame 1_0111b - Selects USB1 start of frame 1_1000b - Selects ACMP0 CMPO 1_1001b - Selects ADC0 trigger completion flag 0 1_1010b - Selects ADC0 trigger completion flag 1 1_1011b - Selects ITRC TMPR_OUT[0] 1_1100b - Selects ITRC TMPR_OUT[1] All other values are reserved.

70.6.1.8 FREQME Clock Selection (FREQME_TRIG0 - FREQME_TRIG1)

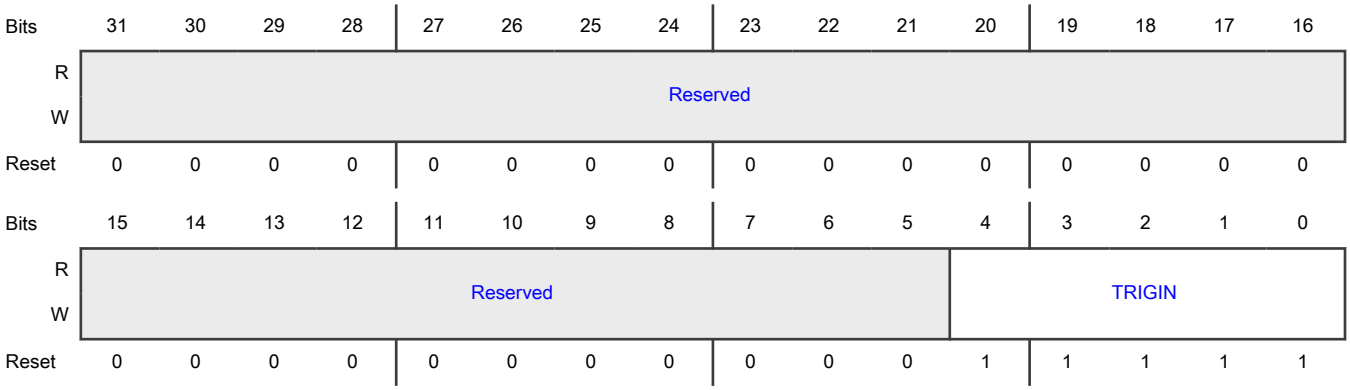
Offset

Register	Offset
FREQME_TRIG0	700h
FREQME_TRIG1	704h

Function

FREQME_TRIG0 selects FREQME reference clock and FREQME_TRIG1 selects FREQME target clock.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 TRIGIN	FREQME Input Clock 0_0000b - Selects main crystal oscillator 0_0001b - Selects FRO1_DIV8 0_0010b - Selects FRO1 0_0011b - Selects LPOSC 0_0100b - Selects OSC32KNP osc32k_out 0_0110b - Selects FRO0 0_0111b - Selects FRO2 0_1000b - Selects FREQME_GPIO_A_CLK 0_1001b - Selects FREQME_GPIO_B_CLK All other values are reserved.

70.6.1.9 EZH-V Input Connections (EZHV_TRIG0 - EZHV_TRIG7)

Offset

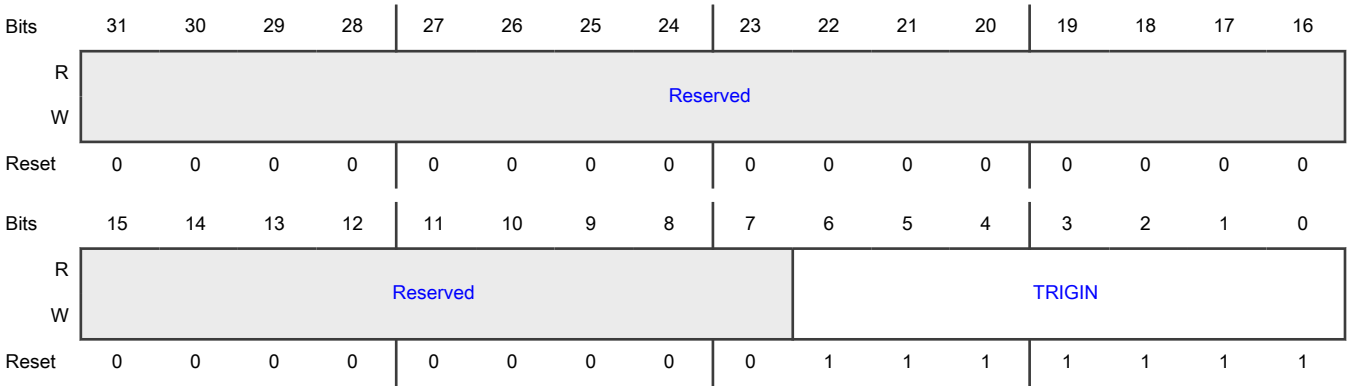
For a = 0 to 7:

Register	Offset
EZHV_TRIGa	720h + (a × 4h)

Function

Selects the EZH-V trigger inputs.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 TRIGIN	EZH-V Input Trigger 000_0000b - Selects GPIO0 peripheral output channel 0 trigger 000_0001b - Selects GPIO0 peripheral output channel 1 trigger 000_0010b - Selects GPIO1 peripheral output channel 0 trigger 000_0011b - Selects GPIO1 peripheral output channel 1 trigger 000_0100b - Selects GPIO2 peripheral output channel 0 trigger 000_0101b - Selects GPIO2 peripheral output channel 1 trigger 000_0110b - Selects GPIO3 peripheral output channel 0 trigger 000_0111b - Selects GPIO3 peripheral output channel 1 trigger 000_1000b - Selects GPIO4 peripheral output channel 0 trigger 000_1001b - Selects GPIO4 peripheral output channel 1 trigger

Table continues on the next page...

Table continued from the previous page...

Field	Function
	000_1010b - Selects GPIO5 peripheral output channel 0 trigger 000_1011b - Selects GPIO5 peripheral output channel 1 trigger 000_1100b - Selects GPIO6 peripheral output channel 0 trigger 000_1101b - Selects GPIO6 peripheral output channel 1 trigger 000_1110b - Selects GPIO7 peripheral output channel 0 trigger 000_1111b - Selects GPIO7 peripheral output channel 1 trigger 001_0000b - Selects LP_FLEXCOMM0 interrupt 001_0001b - Selects LP_FLEXCOMM1 interrupt 001_0010b - Selects LP_FLEXCOMM2 interrupt 001_0011b - Selects LP_FLEXCOMM3 interrupt 001_0100b - Selects LP_FLEXCOMM4 interrupt 001_0101b - Selects LP_FLEXCOMM5 interrupt 001_0110b - Selects LP_FLEXCOMM6 interrupt 001_0111b - Selects LP_FLEXCOMM7 interrupt 001_1000b - Selects LPSPI14 interrupt 001_1001b - Selects LPSPI16 interrupt 001_1010b - Selects I3C0 interrupt 001_1011b - Selects I3C1 interrupt 001_1100b - Selects I3C2 interrupt 001_1101b - Selects I3C3 interrupt 001_1110b - Selects FLEXIO interrupt 001_1111b - Selects GPIO0 interrupt 010_0000b - Selects GPIO1 interrupt 010_0001b - Selects GPIO2 interrupt 010_0010b - Selects GPIO3 interrupt 010_0011b - Selects GPIO4 interrupt 010_0100b - Selects GPIO5 interrupt 010_0101b - Selects GPIO6 interrupt 010_0110b - Selects GPIO7 interrupt 010_0111b - Selects GPIO8 interrupt 010_1000b - Selects GPIO9 interrupt 010_1001b - Selects GPIO10 interrupt 010_1010b - Selects SCT0 interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010_1011b - Selects CTIMER0 interrupt 010_1100b - Selects CTIMER1 interrupt 010_1101b - Selects CTIMER2 interrupt 010_1110b - Selects CTIMER3 interrupt 010_1111b - Selects CTIMER4 interrupt 011_0000b - Selects CTIMER5 interrupt 011_0001b - Selects CTIMER6 interrupt 011_0010b - Selects CTIMER7 interrupt 011_0011b - Selects UTICK0 interrupt 011_0100b - Selects MRT0 interrupt 011_0101b - Selects RTC0 interrupt 011_0110b - Selects OSTIMER event timer CPU0 wakeup/interrupt 011_0111b - Selects WWDT0 interrupt 011_1000b - Selects WWDT1 interrupt 011_1001b - Selects ADC0 interrupt 011_1010b - Selects ACMP0 interrupt 011_1011b - Selects MICFIL interrupt 011_1100b - Selects MICFIL HWVAD 011_1101b - Selects uSDHC0 interrupt 011_1110b - Selects uSDHC1 interrupt 011_1111b - Selects USB0 interrupt 100_0000b - Selects USB1 interrupt 100_0001b - Selects LCDIF interrupt 100_0010b - Selects VGPU interrupt 100_0011b - Selects eDMA0 channel 0 interrupt 100_0100b - Selects eDMA0 channel 1 interrupt 100_0101b - Selects eDMA0 channel 2 interrupt 100_0110b - Selects eDMA0 channel 3 interrupt 100_0111b - Selects eDMA1 channel 0 interrupt 100_1000b - Selects eDMA1 channel 1 interrupt 100_1001b - Selects eDMA1 channel 2 interrupt 100_1010b - Selects eDMA1 channel 3 interrupt 100_1011b - Selects eDMA2 channel 0 interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	100_1100b - Selects eDMA2 channel 1 interrupt 100_1101b - Selects eDMA2 channel 2 interrupt 100_1110b - Selects eDMA2 channel 3 interrupt 100_1111b - Selects eDMA3 channel 0 interrupt 101_0000b - Selects eDMA3 channel 1 interrupt 101_0001b - Selects eDMA3 channel 2 interrupt 101_0010b - Selects eDMA3 channel 3 interrupt 101_0011b - Selects HiFi1 TIE_EXPSTATE_1 101_0100b - Selects HiFi4 TIE_EXPSTATE_1 101_0101b - Selects SCT0_OUT8 101_0110b - Selects SCT0_OUT9 101_0111b - Selects CTIMER4_MAT3 101_1000b - Selects CTIMER4_MAT2 101_1001b - Selects CTIMER3_MAT3 101_1010b - Selects CTIMER3_MAT2 101_1011b - Selects CPU0 TXEV 101_1100b - Selects PINT0 boolean pattern-match output 101_1101b - Selects MIPI_DSI_HOST interrupt 101_1110b - Selects USB0 start of frame 101_1111b - Selects USB1 start of frame 110_0000b - Selects ITRC TMPR_OUT[0] 110_0001b - Selects ITRC TMPR_OUT[1] All other values are reserved.

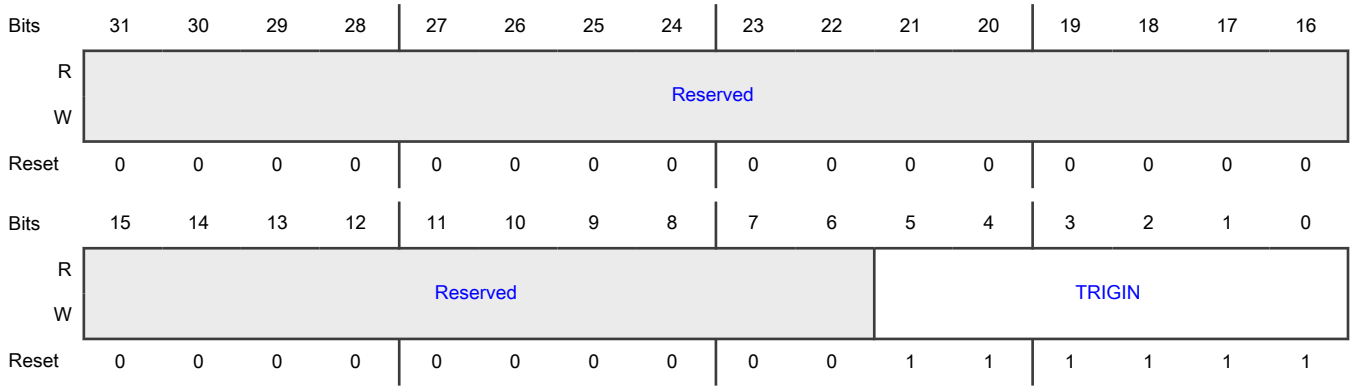
70.6.1.10 FLEXIO Input Connections (FLEXIO_TRIG0 - FLEXIO_TRIG3)

Offset

Register	Offset
FLEXIO_TRIG0	760h
FLEXIO_TRIG1	764h
FLEXIO_TRIG2	768h
FLEXIO_TRIG3	76Ch

Function
Selects the FLEXIO trigger inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	FLEXIO Input Trigger 00_0000b - Selects CTIMER_C_INP0 00_0001b - Selects CTIMER_C_INP1 00_0010b - Selects CTIMER_C_INP2 00_0011b - Selects CTIMER_C_INP3 00_0100b - Selects SCT0_OUT6 00_0101b - Selects SCT0_OUT7 00_0110b - Selects SCT0_OUT8 00_0111b - Selects CTIMER0_MAT3 00_1000b - Selects CTIMER1_MAT3 00_1001b - Selects CTIMER2_MAT3 00_1010b - Selects CTIMER3_MAT3 00_1011b - Selects CTIMER4_MAT3 00_1100b - Selects PINT0 boolean pattern-match output 00_1101b - Selects ACMP0 CMPO 00_1111b - Selects RTC_SS0 interrupt 01_0000b - Selects GPIO0 peripheral output channel 0 trigger 01_0001b - Selects GPIO0 peripheral output channel 1 trigger 01_0010b - Selects GPIO1 peripheral output channel 0 trigger 01_0011b - Selects GPIO1 peripheral output channel 1 trigger 01_0100b - Selects GPIO2 peripheral output channel 0 trigger

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01_0101b - Selects GPIO2 peripheral output channel 1 trigger 01_0110b - Selects GPIO3 peripheral output channel 0 trigger 01_0111b - Selects GPIO3 peripheral output channel 1 trigger 01_1000b - Selects GPIO4 peripheral output channel 0 trigger 01_1001b - Selects GPIO4 peripheral output channel 1 trigger 01_1010b - Selects GPIO5 peripheral output channel 0 trigger 01_1011b - Selects GPIO5 peripheral output channel 1 trigger 01_1100b - Selects GPIO6 peripheral output channel 0 trigger 01_1101b - Selects GPIO6 peripheral output channel 1 trigger 01_1110b - Selects GPIO7 peripheral output channel 0 trigger 01_1111b - Selects GPIO7 peripheral output channel 1 trigger 10_0000b - Selects GPIO8 peripheral output channel 0 trigger 10_0001b - Selects GPIO8 peripheral output channel 1 trigger 10_0010b - Selects GPIO9 peripheral output channel 0 trigger 10_0011b - Selects GPIO9 peripheral output channel 1 trigger 10_0100b - Selects GPIO10 peripheral output channel 0 trigger 10_0101b - Selects GPIO10 peripheral output channel 1 trigger 10_0110b - Selects ITRC TMPR_OUT[0] 10_0111b - Selects ITRC TMPR_OUT[1] All other values are reserved.

70.6.2 INPUTMUX1 register descriptions

70.6.2.1 INPUTMUX1 memory map

INPUTMUX1 base address: 4004_6000h

Offset	Register	Width (In bits)	Access	Reset value
100h - 10Ch	PINT1 Input Connections (PINT1_TRIG0 - PINT1_TRIG3)	32	RW	0000_003Fh
140h - 1A8h	HiFi1 Interrupt Connections (HIFI1_INTERRUPT0 - HIFI1_INTERRUPT26)	32	RW	0000_003Fh
200h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM17_TRIG)	32	RW	0000_000Fh
220h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM18_TRIG)	32	RW	0000_000Fh
240h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM19_TRIG)	32	RW	0000_000Fh

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
260h	LP_FLEXCOMM Input Connections (LP_FLEXCOMM20_TRIG)	32	RW	0000_000Fh
400h - 404h	ADC0 Input Connections (ADC0_TRIG0 - ADC0_TRIG1)	32	RW	0000_001Fh
600h	CTIMER Input Connections (CTIMER5_CAP0)	32	RW	0000_001Fh
604h	CTIMER Input Connections (CTIMER5_CAP1)	32	RW	0000_001Fh
608h	CTIMER Input Connections (CTIMER5_CAP2)	32	RW	0000_001Fh
60Ch	CTIMER Input Connections (CTIMER5_CAP3)	32	RW	0000_001Fh
610h	CTIMER Input Connections (CTIMER5_TRIG)	32	RW	0000_001Fh
620h	CTIMER Input Connections (CTIMER6_CAP0)	32	RW	0000_001Fh
624h	CTIMER Input Connections (CTIMER6_CAP1)	32	RW	0000_001Fh
628h	CTIMER Input Connections (CTIMER6_CAP2)	32	RW	0000_001Fh
62Ch	CTIMER Input Connections (CTIMER6_CAP3)	32	RW	0000_001Fh
630h	CTIMER Input Connections (CTIMER6_TRIG)	32	RW	0000_001Fh
640h	CTIMER Input Connections (CTIMER7_CAP0)	32	RW	0000_001Fh
644h	CTIMER Input Connections (CTIMER7_CAP1)	32	RW	0000_001Fh
648h	CTIMER Input Connections (CTIMER7_CAP2)	32	RW	0000_001Fh
64Ch	CTIMER Input Connections (CTIMER7_CAP3)	32	RW	0000_001Fh
650h	CTIMER Input Connections (CTIMER7_TRIG)	32	RW	0000_001Fh

70.6.2.2 PINT1 Input Connections (PINT1_TRIG0 - PINT1_TRIG3)

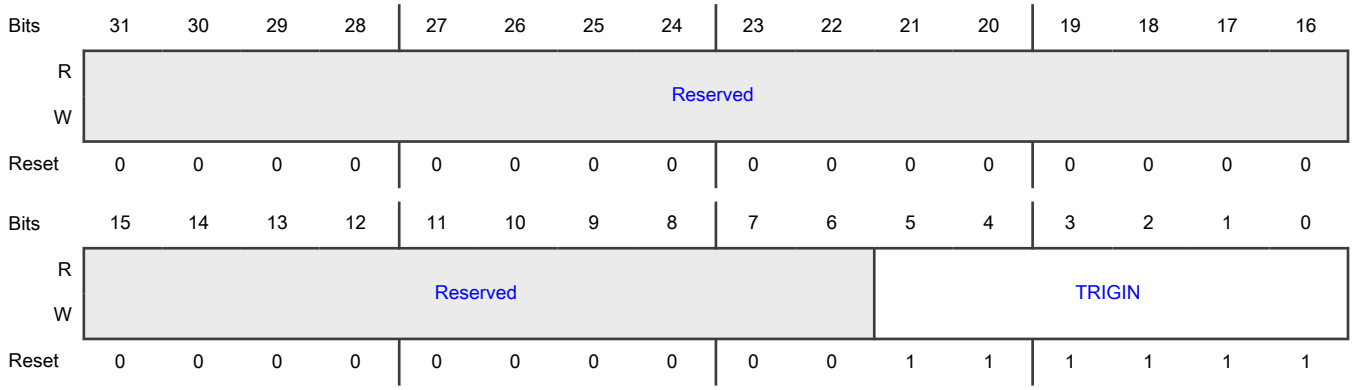
Offset

Register	Offset
PINT1_TRIG0	100h
PINT1_TRIG1	104h
PINT1_TRIG2	108h
PINT1_TRIG3	10Ch

Function

Selects the PINT1 trigger inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	PINT1 Input Trigger 00_0000b - Selects PIO8_0 00_0001b - Selects PIO8_1 00_0010b - Selects PIO8_2 00_0011b - Selects PIO8_3 00_0100b - Selects PIO8_4 00_0101b - Selects PIO8_5 00_0110b - Selects PIO8_6 00_0111b - Selects PIO8_7 00_1000b - Selects PIO8_8 00_1001b - Selects PIO8_9 00_1010b - Selects PIO8_10 00_1011b - Selects PIO8_11 00_1100b - Selects PIO8_12 00_1101b - Selects PIO8_13 00_1110b - Selects PIO8_14 00_1111b - Selects PIO8_15 01_0000b - Selects PIO8_16 01_0001b - Selects PIO8_17 01_0010b - Selects PIO8_18 01_0011b - Selects PIO8_19

Table continues on the next page...

Table continued from the previous page...

Field	Function
	01_0100b - Selects PIO8_20 01_0101b - Selects PIO8_21 01_0110b - Selects PIO8_22 01_0111b - Selects PIO8_23 01_1000b - Selects PIO8_24 01_1001b - Selects PIO8_25 01_1010b - Selects PIO8_26 01_1011b - Selects PIO8_27 01_1100b - Selects PIO8_28 01_1101b - Selects PIO8_29 01_1110b - Selects PIO8_30 01_1111b - Selects PIO8_31 10_0000b - Selects PIO10_0 10_0001b - Selects PIO10_1 10_0010b - Selects PIO10_2 10_0011b - Selects PIO10_3 10_0100b - Selects PIO10_4 10_0101b - Selects PIO10_5 10_0110b - Selects PIO10_6 10_0111b - Selects PIO10_7 10_1000b - Selects PIO10_8 10_1001b - Selects PIO10_9 10_1010b - Selects PIO10_10 10_1011b - Selects PIO10_11 10_1100b - Selects PIO10_12 10_1101b - Selects PIO10_13 10_1110b - Selects PIO10_14 10_1111b - Selects PIO10_15 11_0000b - Selects PIO10_16 11_0001b - Selects PIO10_17 All other values are reserved.

70.6.2.3 HiFi1 Interrupt Connections (HIFI1_INTERRUPT0 - HIFI1_INTERRUPT26)

Offset

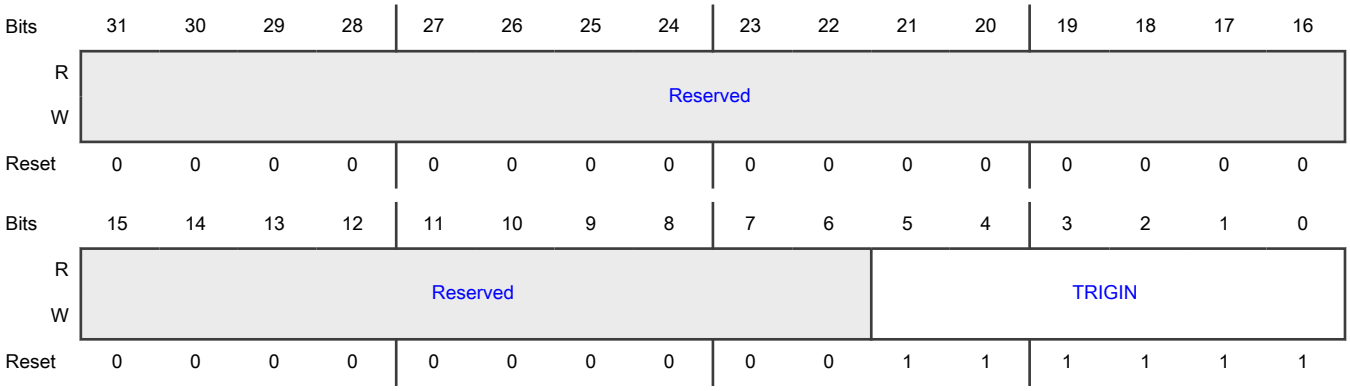
For a = 0 to 26:

Register	Offset
HIFI1_INTERRUPTa	140h + (a × 4h)

Function

Selects the HiFi1 interrupt inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	HiFi1 input interrupt 00_0000b - Selects LP_FLEXCOMM17 interrupt 00_0001b - Selects LP_FLEXCOMM18 interrupt 00_0010b - Selects LP_FLEXCOMM19 interrupt 00_0011b - Selects LP_FLEXCOMM20 interrupt 00_0100b - Selects PMC1 Interrupt 00_0111b - Selects GPIO8 channel 0 interrupt 00_1000b - Selects GPIO8 channel 1 interrupt 00_1001b - Selects GPIO9 channel 0 interrupt 00_1010b - Selects GPIO9 channel 1 interrupt 00_1011b - Selects GPIO10 channel 0 interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	00_1100b - Selects GPIO10 channel 1 interrupt 00_1101b - Selects WWDT2 interrupt 00_1110b - Selects WWDT3 interrupt 00_1111b - Selects MU0_MUB interrupt for HiFi1 to CPU0 01_0000b - Selects MU3_MUB interrupt for HiFi1 to CPU1 01_0001b - Selects UTICK1 interrupt 01_0010b - Selects MRT1 Ored interrupt 01_0011b - Selects OSTIMER event timer HiFi1 wakeup/interrupt 01_0100b - Selects CTIMER5 interrupt 01_0101b - Selects CTIMER6 interrupt 01_0110b - Selects CTIMER7 interrupt 01_0111b - Selects RTC1 alarm interrupt to Sense domain 01_1000b - Selects RTC_SS1 wakeup interrupt to Sense domain 01_1001b - Selects I3C2 interrupt 01_1010b - Selects I3C3 interrupt 01_1011b - Selects MICFIL interrupt for read data or error 01_1100b - Selects MICFIL hardware voice activity detector interrupt or error interrupt 10_0010b - Selects eDMA2 channel 0 interrupt 10_0011b - Selects eDMA2 channel 1 interrupt 10_0100b - Selects eDMA2 channel 2 interrupt 10_0101b - Selects eDMA2 channel 3 interrupt 10_0110b - Selects eDMA2 channel 4 interrupt 10_0111b - Selects eDMA2 channel 5 interrupt 10_1000b - Selects eDMA2 channel 6 interrupt 10_1001b - Selects eDMA2 channel 7 interrupt 10_1010b - Selects eDMA3 channel 0 interrupt 10_1011b - Selects eDMA3 channel 1 interrupt 10_1100b - Selects eDMA3 channel 2 interrupt 10_1101b - Selects eDMA3 channel 3 interrupt 10_1110b - Selects eDMA3 channel 4 interrupt 10_1111b - Selects eDMA3 channel 5 interrupt 11_0000b - Selects eDMA3 channel 6 interrupt 11_0001b - Selects eDMA3 channel 7 interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	11_0010b - Selects PINT1 interrupt 0 11_0011b - Selects PINT1 interrupt 1 11_0100b - Selects PINT1 interrupt 2 11_0101b - Selects PINT1 interrupt 3 11_0110b - Selects SAI3 TX/RX interrupt All other values are reserved.

70.6.2.4 LP_FLEXCOMM Input Connections (LP_FLEXCOMM17_TRIG - LP_FLEXCOMM20_TRIG)

Offset

Register	Offset
LP_FLEXCOMM17_TRIG	200h
LP_FLEXCOMM18_TRIG	220h
LP_FLEXCOMM19_TRIG	240h
LP_FLEXCOMM20_TRIG	260h

Function

Selects the LP_FLEXCOMM trigger inputs.

NOTE

When TRIGIN field is configured as 1100b to select PINT1 boolean pattern-match output as input trigger and this LP_FLEXCOMM is used as LPUART, the related CFG field of PINT1.PMCFG register must not be configured as 111b (Event).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRIGIN			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Function
31-4 —	Reserved
3-0 TRIGIN	LP_FLEXCOMM Input Trigger 0000b - Selects CTIMER_S_INP0 0001b - Selects CTIMER_S_INP1 0010b - Selects CTIMER_S_INP2 0011b - Selects CTIMER_S_INP3 0111b - Selects CTIMER5_MAT3 1000b - Selects CTIMER6_MAT3 1001b - Selects CTIMER7_MAT3 1100b - Selects PINT1 boolean pattern-match output 1101b - Selects ACMP0 CMPO 1110b - Selects RTC_SS1 Interrupt All other values are reserved.

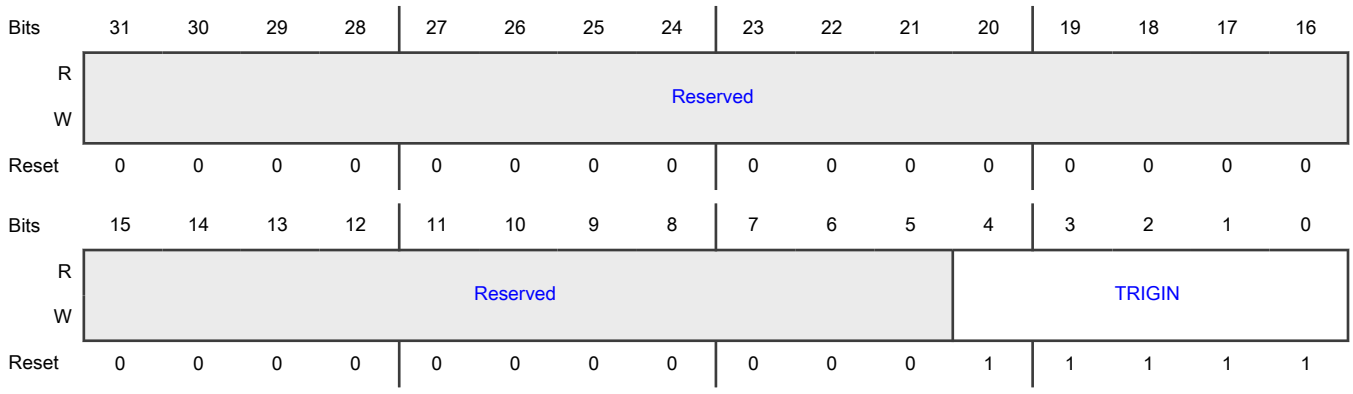
70.6.2.5 ADC0 Input Connections (ADC0_TRIG0 - ADC0_TRIG1)

Offset

Register	Offset
ADC0_TRIG0	400h
ADC0_TRIG1	404h

Function
Selects the ADC0 trigger inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 TRIGIN	ADC0 Input Trigger 0_0000b - Selects GPIO0 peripheral output channel 0 trigger 0_0001b - Selects GPIO1 peripheral output channel 0 trigger 0_0010b - Selects GPIO2 peripheral output channel 0 trigger 0_0011b - Selects GPIO3 peripheral output channel 0 trigger 0_0100b - Selects GPIO4 peripheral output channel 0 trigger 0_0101b - Selects GPIO5 peripheral output channel 0 trigger 0_0110b - Selects GPIO6 peripheral output channel 0 trigger 0_0111b - Selects GPIO7 peripheral output channel 0 trigger 0_1000b - Selects GPIO8 peripheral output channel 0 trigger 0_1001b - Selects GPIO9 peripheral output channel 0 trigger 0_1010b - Selects GPIO10 peripheral output channel 0 trigger 0_1011b - Selects SCT0_OUT4 0_1100b - Selects SCT0_OUT5 0_1101b - Selects SCT0_OUT9 0_1110b - Selects CTIMER0_MAT3 0_1111b - Selects CTIMER1_MAT3 1_0000b - Selects CTIMER2_MAT3 1_0001b - Selects CTIMER3_MAT3 1_0010b - Selects CTIMER4_MAT3 1_0011b - Selects CTIMER5_MAT3 1_0100b - Selects CTIMER6_MAT3 1_0101b - Selects CTIMER7_MAT3 1_0110b - Selects ACMP0 CMPO 1_0111b - Selects CPU0 TXEV 1_1000b - Selects CPU1 TXEV 1_1010b - Selects PINT0 boolean pattern-match output 1_1011b - Selects PINT1 boolean pattern-match output All other values are reserved.

70.6.2.6 CTIMER Input Connections (CTIMER5_CAP0 - CTIMER7_CAP3)

Offset

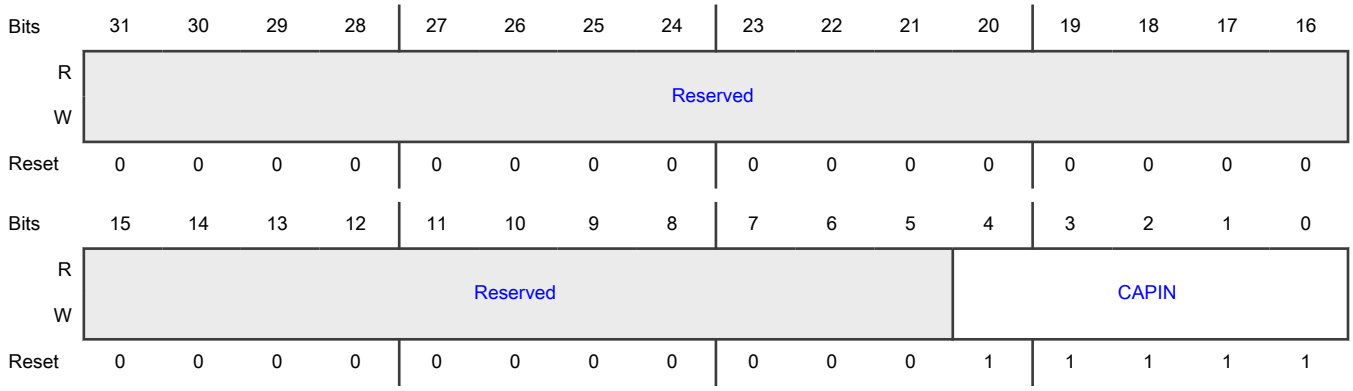
For a = 5 to 7; b = 0 to 3:

Register	Offset
CTIMERa_CAPb	560h + (a × 20h) + (b × 4h)

Function

Selects the CTIMER capture inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 CAPIN	CTIMER Input Capture 0_0000b - Selects CTIMER_S_INP0 0_0001b - Selects CTIMER_S_INP1 0_0010b - Selects CTIMER_S_INP2 0_0011b - Selects CTIMER_S_INP3 0_0100b - Selects CTIMER_S_INP4 0_0101b - Selects CTIMER_S_INP5 0_0110b - Selects CTIMER_S_INP6 0_0111b - Selects CTIMER_S_INP7 0_1000b - Selects CTIMER_S_INP8 0_1001b - Selects CTIMER_S_INP9

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1_0000b - Selects SAI3_TX_SYNC 1_0001b - Selects SAI3_RX_SYNC 1_0010b - Selects ACMP0 CMPO 1_0011b - Selects ADC0 trigger completion flag 0 1_0100b - Selects ADC0 trigger completion flag 1 All other values are reserved.

70.6.2.7 CTIMER Input Connections (CTIMER5_TRIG - CTIMER7_TRIG)

Offset

Register	Offset
CTIMER5_TRIG	610h
CTIMER6_TRIG	630h
CTIMER7_TRIG	650h

Function
Selects the CTIMER trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRIGIN			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Function
31-5 —	Reserved
4-0	CTIMER Input Trigger

Table continues on the next page...

Table continued from the previous page...

Field	Function
TRIGIN	0_0000b - Selects CTIMER_S_INP0 0_0001b - Selects CTIMER_S_INP1 0_0010b - Selects CTIMER_S_INP2 0_0011b - Selects CTIMER_S_INP3 0_0100b - Selects CTIMER_S_INP4 0_0101b - Selects CTIMER_S_INP5 0_0110b - Selects CTIMER_S_INP6 0_0111b - Selects CTIMER_S_INP7 0_1000b - Selects CTIMER_S_INP8 0_1001b - Selects CTIMER_S_INP9 1_0000b - Selects SAI3_TX_SYNC 1_0001b - Selects SAI3_RX_SYNC 1_0010b - Selects ACMP0 CMPO 1_0011b - Selects ADC0 trigger completion flag 0 1_0100b - Selects ADC0 trigger completion flag 1 All other values are reserved.

Chapter 71
Enhanced Direct Memory Access (eDMA)

71.1 Chip-specific eDMA information

Table 687. Reference links to related information

Topic	Related module	Reference
Full description	eDMA	eDMA
Peripheral memory map		Memory map overview
Clocking		Clock overview Clock controller
Interrupts		Interrupts and DMA events assignments
Input multiplexing	INPUTMUX	See CTIMERn_TRIG and CTIMERn_CAPm (n=0-7, m=0-3) registers in INPUTMUX

71.1.1 Module instances

This device has four eDMA controllers:

Table 688. eDMA instances

	eDMA0, eDMA1	eDMA2, eDMA3
Located in	VDD2_COMP	VDD1_SENSE
Core access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1	CPU1, HiFi1, EZH-V, eDMA2, eDMA3
Channel numbers	16	8

71.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

71.1.3 Module DMA requests enable

All the module DMA requests are disabled by default, you can enable them by SYSCON registers.

Table 689. Module DMA requests enable

DMA requests of eDMA instance	Enabled by register
eDMA0	SYSCON0.EDMA0_EN0-SYSCON0.EDMA0_EN3
eDMA1	SYSCON0.EDMA1_EN0-SYSCON0.EDMA1_EN3
eDMA2	SYSCON1.EDMA2_EN0-SYSCON1.EDMA2_EN3
eDMA3	SYSCON1.EDMA3_EN0-SYSCON1.EDMA3_EN3

71.2 Overview

The enhanced direct memory access (eDMA) controller is capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 16 channels

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 690. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

71.2.1 Block diagram

Figure 296 illustrates the components of the eDMA system, including the eDMA module (engine).

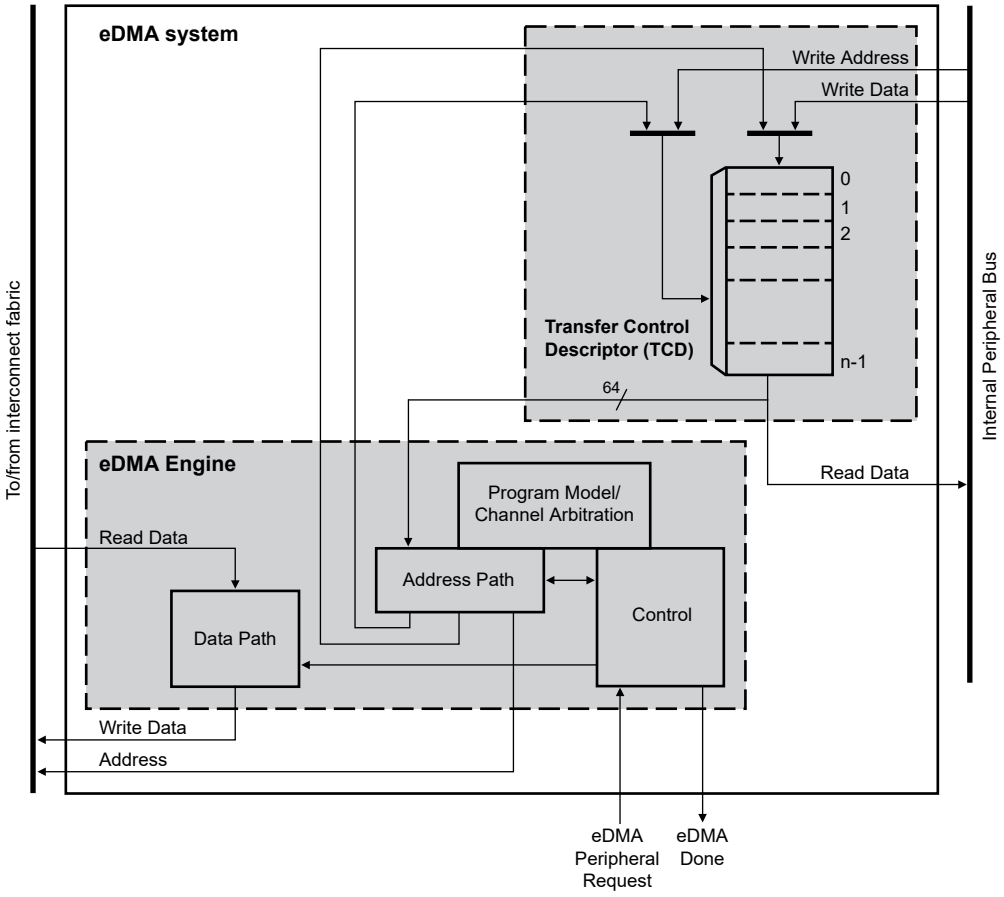


Figure 296. Block diagram

71.2.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 691. eDMA engine submodules

Submodule	Function
Address path	This block: • Implements a primary channel and secondary (preempt) channel • Manages all initiator bus-address calculations All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the primary channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by CH*n*_PRI[ECP]) where a large data transfer can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of a new TCD*n* from memory as part of a scatter/gather operation. See Dynamic scatter/gather for more details.
Data path	This block implements the bus initiator read/write data path. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the first stage of the bus pipeline (address phase), and the data path module implements the second stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has been moved from the source to the destination. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, the eDMA performs two reads, then one 32-bit write.

The transfer control descriptor local memory is further partitioned into:

Table 692. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, and manages accesses from the eDMA engine as well as references from the internal peripheral bus. In simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

71.2.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and is not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for complex address calculations
- 16-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage for all transfers
  - Connections to the crossbar switch for bus driving the data movement
- TCD organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
  - One interrupt per channel, which can be asserted at completion of major iteration count
  - Programmable error terminations per channel that are logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

71.3 Functional description

The operation of eDMA is described in the following subsections.

71.3.1 Modes of operation

The eDMA operates in the following modes:

Table 693. Modes of operation

Mode	Description
Normal	In Normal mode, eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the TCD. The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	eDMA operation is configurable in Debug mode via the control register: • If CSR[EDBG] is cleared to 0, eDMA continues to operate. • If CSR[EDBG] is set to 1, eDMA stops transferring data. If Debug mode is entered when a channel is active, eDMA continues operation until the channel retires.

71.3.2 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

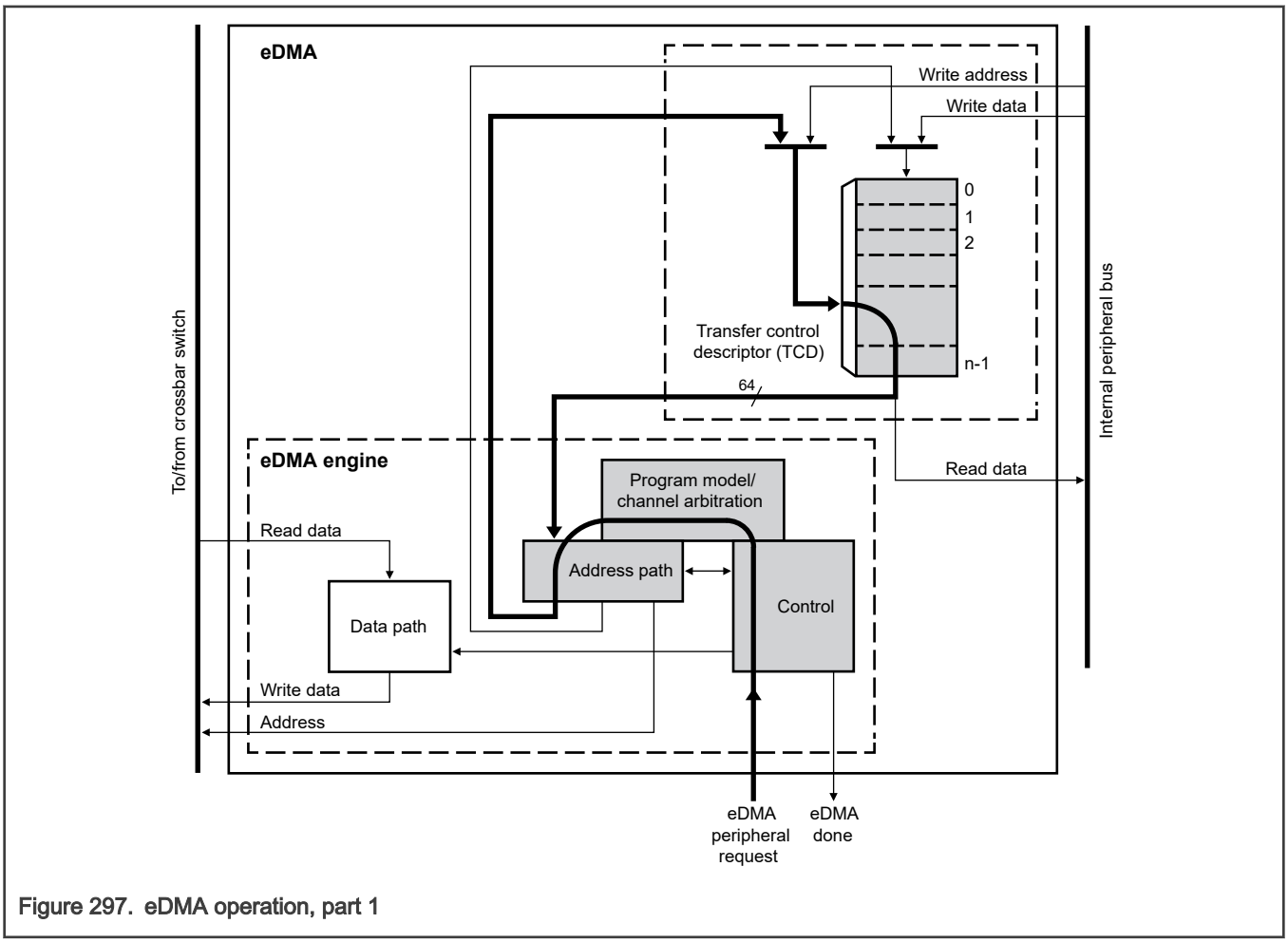


Figure 297. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] field follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration.

In the next cycle, the channel arbitration begins using fixed-priority plus the optional round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor is read from the local memory and then loaded into the eDMA engine address path's primary or secondary channel execution registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path registers.

The following diagram illustrates the second part of the basic data flow:

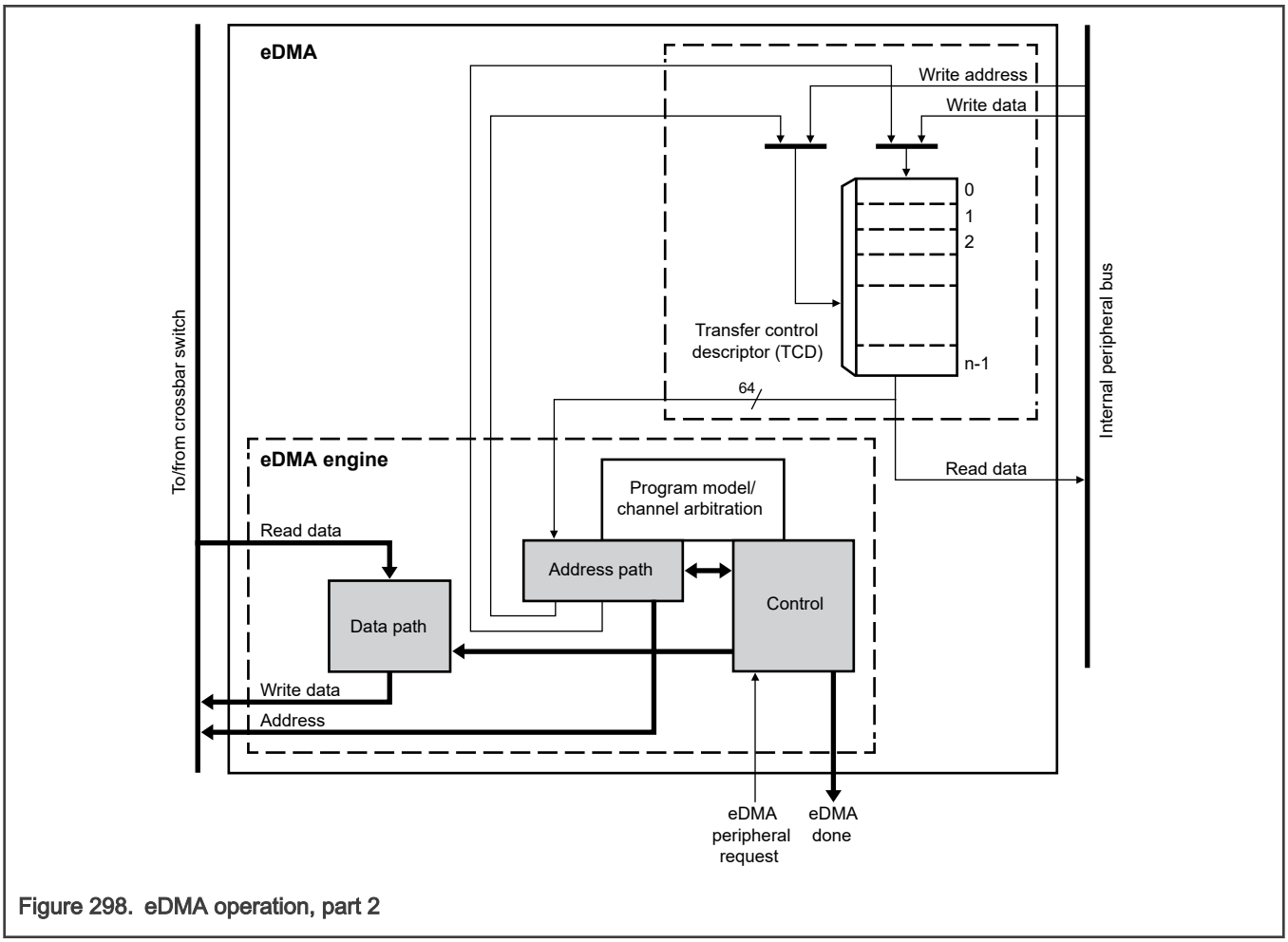


Figure 298. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) go through the required sequence of source reads and destination writes to perform the actual data movement. The source reads are initiated, and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the byte count, NBYTES, has been transferred.

After NBYTES of data has been moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD (for example, SADDR, DADDR, CITER). If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER field. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

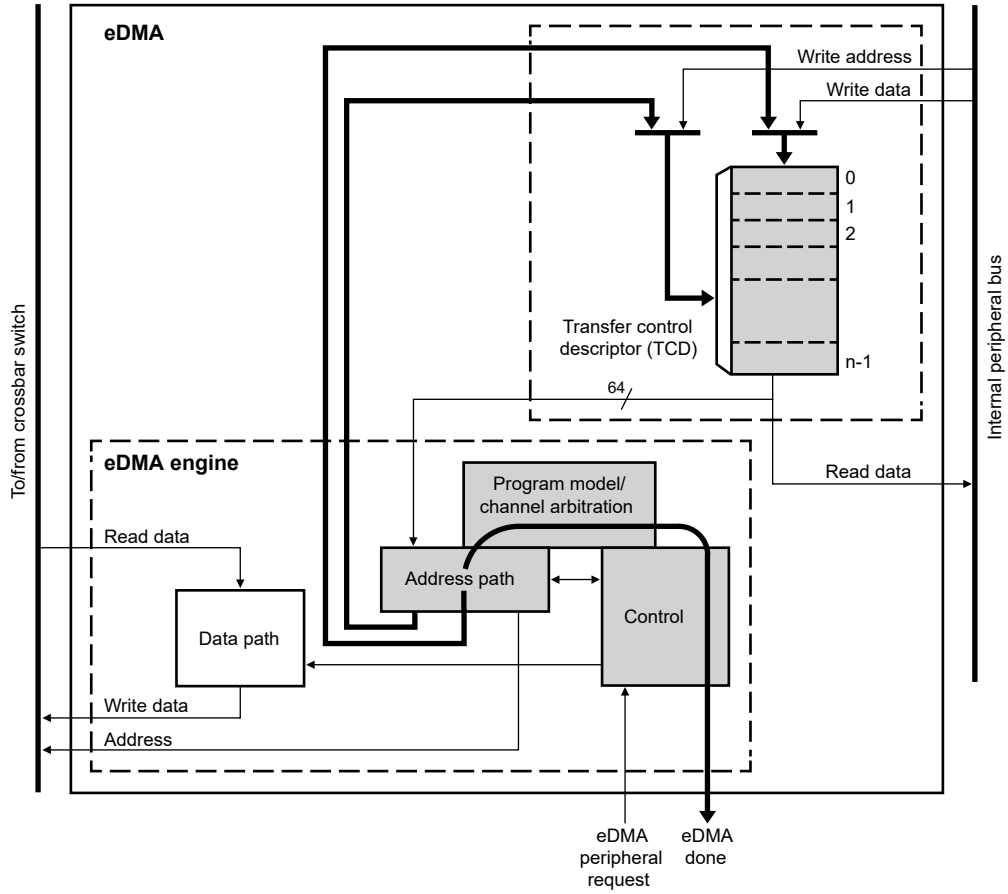


Figure 299. eDMA operation, part 3

71.3.3 Fault reporting and handling

Channel errors are reported in the Error Status register (CHn_ES) and can be caused by any of the following:

- A configuration error, which is an illegal setting in the transfer control descriptor
- An active channel canceled via a "cancel transfer with error" hardware or software request
- An error termination to a bus initiator read or write cycle

A configuration error is reported when an inconsistent state is represented by one of these factors:

- Starting source or destination address
- Source or destination offsets
- Minor loop byte count
- Transfer size

Each of these possible causes is detailed below:

- The addresses and offsets must be aligned on zero-modulo-transfer-sized boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size.

NOTE

To aid in debugging, set the Halt After Error field in the DMA's Control Status register, CSR[HAE]. Upon any error condition, the DMA is halted after the error is recorded. The DMA remains halted and does not process any channel service requests. After the error is fixed, the DMA may be enabled again by clearing the Halt field, CSR[HALT].

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (TCDn_DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[ELINK] field does not equal the TCDn_BITER[ELINK] field.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, are reported as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion if properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel field in the eDMA error register is set to 1. At the same time, the details of the error condition are loaded into the Error Status register (CHn_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag, and the possible assertion of an interrupt request are not affected when an error is detected.

After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

The error status fields are read-only. These error indicators are sticky and cannot be cleared. They show the last recorded error until the DMA is reset. CHn_ES[ERR] is used to determine if a new error condition exists. This field is the logical OR of each channel's error interrupt field (ERR).

After the software has resolved all errors and cleared all of the error interrupt fields, the MP_ES[VLD] is cleared to 0 but the cause of the last error is still indicated.

71.3.4 Channel preemption

The eDMA uses a priority vector value to determine the highest priority channel requesting service.

The priority vector is a combination of:

1. the channel's group priority, CHn_GRPRI
2. the channel's priority level, CHn_PRI[APL]

It can be considered a number composed of these concatenated priority levels: CHn_GRPRI : CHn_PRI[APL]

Priority vector = ((CHn_GRPRI << 8) + (CHn_PRI[APL] << 5) + CH_NUM)

A channel requesting service with the highest priority vector value will receive the next execution slot.

An execution slot is available:

1. immediately if the eDMA is idle
2. when an active channel retires
3. when valid preemption conditions exist

NOTE

Preemption is strictly priority based. Preemption is not bound by a specific group number as defined by CHn_GRPRI.

Channel preemption is enabled on a per-channel basis by setting the CHn_PRI[ECP] field. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher-priority channel. After the preempting channel has completed all of its minor loop data transfers, the preempted channel is restored and resumes execution.

After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended, and the higher-priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted.

A channel's ability to preempt another channel can be disabled by setting CHn_PRI[DPA] to 1. When a channel's preempt ability is disabled, that channel cannot suspend a lower-priority channel's data transfer, regardless of the lower-priority channel's ECP setting. This allows for a pool of low-priority, large-data-moving channels to be defined.

You can configure these low-priority channels to not preempt each other, thus preventing a low-priority channel from consuming the preempt slot normally available to a true high-priority channel. When you enable round-robin channel arbitration mode (CSR[ERCA] is set to 1), any channel with a priority level equal to 0 (CHn_PRI[APL] = 0) has preemption disabled and cannot preempt another channel.

71.3.5 Clocking

The following table describes the clock sources:

Table 694. Clocks

Clock	Description
hclk	Clock used for the DMA engine
ipg_clk	Clock used for the DMA programming model
dma_ram_clk	Clock used for the TCD RAMs

71.3.6 Interrupts

Software can enable the interrupt for each channel for the following events:

1. The major loop is half complete (INTHALF)
2. The major loop is complete (INTMAJOR)
3. A configuration error occurs (EEI)

71.4 External signals

This module has no external signals.

71.5 Initialization

The following sections discuss initialization of the eDMA and programming considerations.

71.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the MP_CSR if a configuration other than the default is wanted.
2. Write the channel priority levels to the CHn_PRI registers and group priority levels to the CHn_GRPRI registers if a configuration other than the default is wanted.

3. Enable error interrupts in the CHn_CSR[EEI] registers if they are wanted.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the CHn_CSR[ERQ] registers.
6. Request channel service via either:
   - Software: setting TCDn_CSR[START]
   - Hardware: target device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 695, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, defined by TCDn_SADDR, to the destination, defined by TCD_DADDR, continue until the number of bytes specified by TCDn_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCDn_SADDR, TCDn_DADDR, and TCDn_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, then eDMA executes further post-processing, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 695. TCD control and status (TCDn_CSR) fields

TCDn_CSR field name	Description
START	Control field to start the channel explicitly when using a software-initiated DMA service (automatically cleared by hardware)
EEOP	Control field to enable end-of-packet processing
ESDA	Control field to enable storing of the destination address to system memory after the major loop completes
DREQ	Control field to disable hardware-initiated DMA service requests after major loop completion
BWC	Control field for throttling the bandwidth control of a channel
ESG	Control field to enable the scatter-gather feature
INTHALF	Control field to enable interrupt when major loop is half-complete
INTMAJOR	Control field to enable interrupt when major loop completes

Table 696. Channel control and status (CHn_CSR) fields

CHn_CSR field name	Description
ACTIVE	Status field indicating the channel is currently in execution
DONE	Status field indicating major loop completion (cleared by software when a channel begins execution)
EEI	Control field to enable error interrupts
EARQ	Control field to enable external, asynchronous wakeup event in conjunction with the ERQ field
ERQ	Control field to enable hardware service requests

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

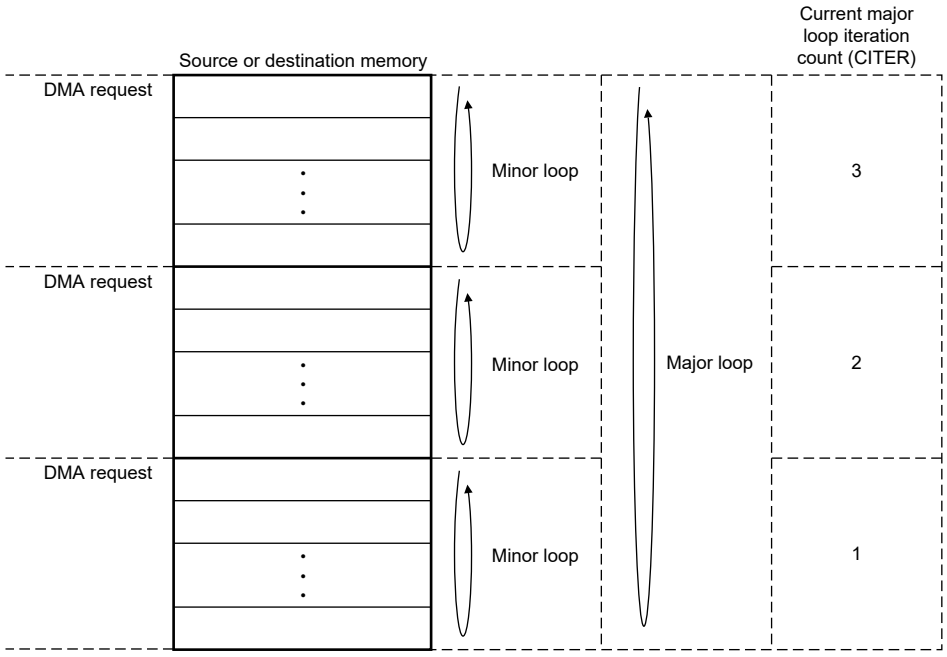


Figure 300. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings are related.

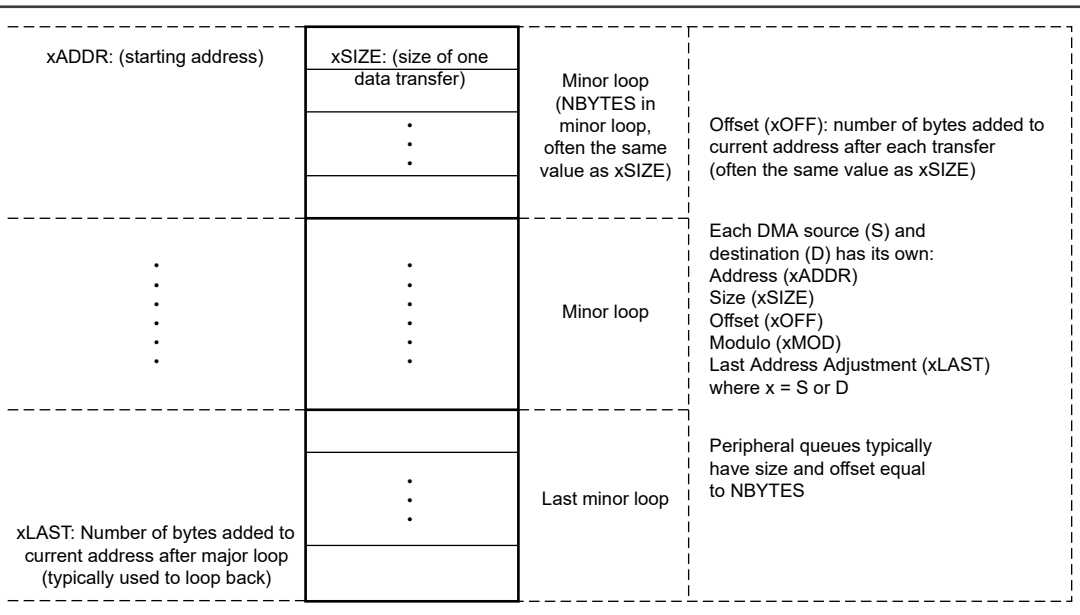


Figure 301. Memory array terms

71.5.2 eDMA arbitration

The eDMA uses a layered arbitration scheme composed of multiple priority levels. The eDMA uses a fixed-priority arbitration scheme with optional round-robin arbitration under specific conditions. The priorities are evaluated in the following order:

Table 697. eDMA arbitration priorities

Priority	Scheme	Description
1 (Highest)	Arbitration group priority	Each channel is assigned an arbitration group via the CHn_GRPRI registers. Priority is given to the highest value (31 being the highest possible value) down to the lowest value (zero, the default).
2	Channel priority	Each channel is assigned a channel priority level via the CHn_PRI registers. The channel priority is a relative priority level within an arbitration group. Priority is given to the highest value (seven being the highest possible value) down to the lowest value (zero, the default). Channel priorities within each arbitration group need not be unique. If multiple channels have the same channel priority level, the channel number will be used to determine priority as defined in row three.
3	Channel number	When two or more channels have the same arbitration group priority and channel priority, the channel number (CHn_NUM) is used to determine the highest priority. Priority is given to the highest channel number. Lowest priority is channel 0. The channel numbers are static and cannot be changed in the programmer's model.
4 (Lowest)	Round-robin	When round-robin is enabled, any channel configured for round-robin operation has lowest priority within an arbitration group. Round-robin is enabled by setting the MP_CSR[ERCA] field to 1. After being enabled, channels with a channel priority of zero (CHn_PRI=0) will use round-robin arbitration. Round-robin arbitration will rotate the channel selection among the channels requesting service with CHn_PRI=0 within the arbitration group. Any non-zero channel within the arbitration group will continue to use fixed-priority arbitration, and if requesting service will be selected over any round-robin channels.

For fixed arbitration, the overall priority can be considered a number composed of three concatenated priority levels: CHn_GRPRI:CHn_PRI:CH_NUM. The largest number has the highest priority and the lowest number has the lowest priority.

For round-robin arbitration, the priority number is CHn_GRPRI:0:X. The module rotates through the CHn_PRI=0 channels requesting service without regard to priority among these channels. Any channel within the arbitration group for which CHn_PRI is greater than 0 will be serviced before the round-robin channels.

71.5.3 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data.

The channel number causing the error is recorded in the Error Status register (CHn_ES). If the error source is not removed before the next activation of the problematic channel, the error is detected and recorded again. Setting the halt after error field, CSR[HAE], will halt the DMA and prevent recurrence of the error.

71.5.4 Arbitration mode considerations

This section discusses arbitration considerations for eDMA.

71.5.4.1 Fixed group arbitration, fixed channel arbitration

In this mode, eDMA selects for execution the channel service request from the highest-priority channel in the highest-priority group. If eDMA is programmed so that the channels within a high-priority group have a high number of requests or large data transfers, that group may consume all the bandwidth of the eDMA controller. That is, no lower-priority groups are serviced if there is always at least one DMA request pending on a channel in the highest-priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly.

71.5.4.2 Fixed group arbitration, round-robin channel arbitration

The highest-priority group with a request is serviced. Lower-priority groups are serviced if no pending requests exist in the higher-priority groups.

Within each group, channels are serviced starting with the highest non-zero channel priority. For all channels with a channel priority programmed to 0, selection begins with the highest channel number requesting service and then rotates through to the lowest channel number requesting service. The round-robin channel arbitration can provide a fairness mechanism to lower-priority channels.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, fixed channel arbitration, but all the channels in the highest-priority group will be serviced. Service latency is short on the highest-priority group, but could potentially be very much longer as the group priority decreases.

71.5.5 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

71.5.5.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the CH*n*_CSR[DONE] field is set to 1 and an interrupt is generated if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte-wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source, and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
```

```
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJ] = 1
TCDn_CSR[START] = 1 (should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] field requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:
   - CH*n*_CSR[DONE] = 0
   - TCD*n*_CSR[START] = 0
   - CH*n*_CSR[ACTIVE] = 1
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32 bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

71.5.5.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop, transferring 16 bytes per iteration. After the channel's hardware requests are enabled via the CHn_CSR[ERQ] register field, the target device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware (eDMA peripheral) requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.

4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
    h. Write 32 bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0.
8. The channel retires, which concludes one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware (eDMA peripheral) requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: Channel TCD data from local memory to internal register file.
13. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32 bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32 bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32 bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32 bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
16. The channel retires, which concludes with the major loop complete. The eDMA goes idle or services the next channel.

71.5.5.3 Using the modulo feature

The modulo feature of the eDMA allows implementation of a circular data queue in which the size of the queue is a power of 2. xMOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature. Modulo addressing applies to cases where the minor loop offset is enabled; that is, the upper address bits remain the same after the minor loop offset is added to the source or destination address.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value but the 28 upper address bits (0x1234567*x*) retain their original value. In this example, the source address is set to 0x12345670, the offset is set to four bytes, and the MOD field is set to four, which allows for a 2^4 byte (16 byte) queue size.

Table 698. Modulo example

Transfer number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

71.5.6 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

71.5.6.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software-initiated service requests.

1. The first method is to read the TCDn_CITER field and test for a change.
2. The second method, extracted from the sequence shown below, is to test the TCDn_CSR[START] field and the CHn_CSR[ACTIVE] field. The minor-loop-complete condition is indicated by both fields reading 0 after TCDn_CSR[START] is set to 1. Polling the CHn_CSR[ACTIVE] field only may be inconclusive because the active status may be missed if the channel execution is short in duration.

The CHn_CSR and TCDn_CSR status fields execute the following sequence for a software-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	1	0	0	Initiate channel service request via software.
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

The best method to test for minor-loop completion when using hardware-initiated (that is, peripheral-initiated) service requests is to read the TCDn_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status fields execute the following sequence for a hardware-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	0	0	0	Initiate channel service request via hardware (peripheral request asserted).
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

For both activation types, the major-loop-complete status is explicitly indicated via the CHn_CSR[DONE] field.

The TCDn_CSR[START] field is cleared to 0 automatically when the channel begins execution, regardless of how the channel activates.

71.5.6.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCDn_SADDR, TCDn_DADDR, and TCDn_NBYTES values if they are read when a channel executes. The true values of SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file, and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES (which decrements to zero as the transfer progresses), can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

71.5.6.3 Checking channel preemption status

A preemptive situation is one in which a preempt-enabled channel is executing and a higher-priority request becomes active. When round-robin channel arbitration mode is enabled, all channels with their channel priority set to 0 lose their preempt ability. Channel priorities of 0 are treated as equal, that is, they are constantly rotating, when round-robin arbitration mode is enabled.

The CHn_CSR[ACTIVE] field for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended when the preempting channel executes one major loop iteration. If two CHn_CSR[ACTIVE] fields are set simultaneously in the global TCD map, a higher-priority channel is actively preempting a lower-priority channel.

71.5.7 Channel linking

Channel linking (or chaining) is a mechanism in which one channel sets the TCDn_CSR[START] field of another channel (or itself), thus initiating a service request for that channel. When properly enabled, the eDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCDn_CITER[ELINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, using an initial field setting of:

```
MP_CSR[GCLC] = 1
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJORELINK] = 1
TCDn_CSR[MAJORLINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] field
2. Minor loop done → set TCD12_CSR[START] field
3. Minor loop done → set TCD12_CSR[START] field
4. Minor loop done, major loop done→ set TCD7_CSR[START] field

When minor loop linking is enabled (TCDn_CITER[ELINK] = 1), the TCDn_CITER[CITER] field uses a nine-bit vector to form the current iteration count. When minor loop linking is disabled (TCDn_CITER[ELINK] = 0), the TCDn_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCDn_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

NOTE

The TCDn_CITER[ELINK] field and the TCDn_BITER[ELINK] field must be equal — if they are not, a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop halfway done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD, at the end of a loop.

Table 699. Channel linking parameters

Wanted link behavior	TCD control field name	Description
Link at end of minor loop	TCDn_CITER[ELINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	TCDn_CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of major loop	TCDn_CSR[MAJORELINK]	Enable channel-to-channel linking on major loop completion
	TCDn_CSR[MAJORLINKCH]	Link channel number when linking at end of major loop

71.5.8 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

Steps for channel linking:

1. Set GCLC = 1
2. For linking after major loop complete:
   a. Set MAJORLINKCH = target channel number
   b. Set MAJORELINK = 1, enable the major loop link
   c. Upon major loop completion (CITER decrements to 0), the link will be made.
3. For linking on minor loop completion:
   a. Set LINKCH = target channel number
   b. Set ELINK = 1, enable the minor loop link
   c. Set the beginning iteration count to the same values; LINKCH = target channel and ELINK = 1,
   d. Upon minor loop completion (that is, after each service request except when the major loop is complete) the link will be made.

71.5.8.1 Dynamically changing the channel priority

To change group or channel priority levels:

1. Halt the DMA by writing 1 to the CSR[HALT] field.
2. Change the group or channel priorities as wanted.
3. Enable normal DMA operations by writing 0 to the CSR[HALT] field.

71.5.8.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCDn_CSR[MAJORELINK] field during channel execution (see the diagram in TCD structure). This field is read from the TCD local memory at the end of channel execution, thus allowing you to enable the feature during channel execution.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic channel link by enabling the TCDn_CSR[MAJORELINK] field at the same time the eDMA engine is retiring the channel. TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

We recommend that you use the following coherency model when executing a dynamic channel link request.

1. Write 1 to the TCDn_CSR[MAJORELINK] field.
2. Read back the TCDn_CSR[MAJORELINK] field.
3. Test the TCDn_CSR[MAJORELINK] request status:
   - If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
   - If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCDn_CSR[MAJORELINK] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field is set to 1, indicating the major loop is complete.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing to the TCDn_CSR[MAJORELINK] field. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution.

71.5.8.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in the eDMA programmer's model, thus replacing the current descriptor.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] field at the same time the eDMA engine is retiring the channel. The TCDn_CSR[ESG] field would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods are recommended for executing a dynamic scatter/gather request. Whenever the TCDn_CSR is written, the TCD local memory controller forces the TCDn_CSR[ESG] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field has been set to 1, indicating the major loop is complete. If attempting to set the ESG, ensure the DONE field is cleared to 0.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing the TCDn_CSR[ESG] or TCDn_CSR[ESG] fields. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution and is set to 1 upon major loop completion.

71.5.8.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCDn_CSR[MAJORELINK] field is 0, the TCDn_CSR[MAJORLINKCH] field is not used by the eDMA. In this case, the TCDn_CSR[MAJORLINKCH] bits may be used for other purposes. This method uses the TCDn_CSR[MAJORLINKCH] field as a TCDn_CSR identification (ID).

When the descriptors are built, write a unique TCDn_CSR ID in the TCDn_CSR[MAJORLINKCH] field for each TCDn_CSR associated with a channel using dynamic scatter/gather.

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the TCDn_CSR[DREQ] field to 1 will prevent future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the 16-bit TCDn_CSR control/status field.
5. Test the TCDn_CSR[ESG] request status and TCDn_CSR[MAJORLINKCH] value:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0 and the MAJORLINKCH (ID) did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the MAJORLINKCH (ID) changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

71.5.8.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCDn_DLAST_SGA field as a TCD identification (ID).

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the DREQ field to 1 will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the TCDn_CSR[ESG] field.
5. Test the TCDn_CSR[ESG] request status:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0, read the 32-bit TCDn_DLAST_SGA field.
   - If ESG = 0 and the TCDn_DLAST_SGA did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the TCDn_DLAST_SGA changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

71.5.9 Lockstep

This chip contains an eDMA Checker which, when enabled, operates in Lockstep mode with the primary eDMA, executing the exact same transfer profiles. The eDMA Checker checks to ensure the primary eDMA is operating correctly. A block diagram is shown below.

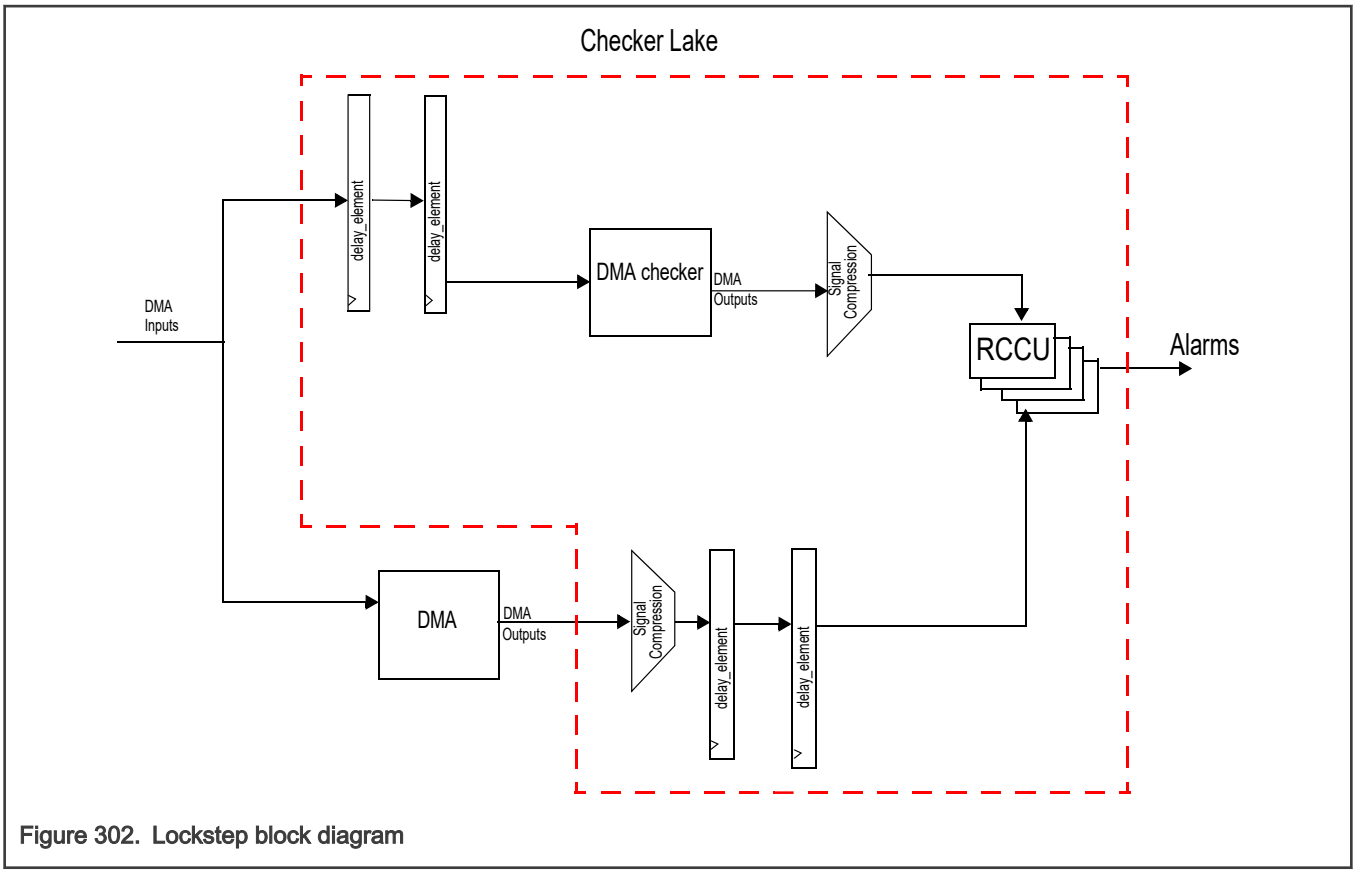


Figure 302. Lockstep block diagram

Initialization

To prevent the eDMA Checker from generating a false comparison error after power-on-reset, the eDMA's data output buffer must be initialized prior to enabling the Checker. To initialize the eDMA data output buffer, the eDMA must perform one or more 32-bit data transfers using any data value.

Errors

To handle source-address errors (SAE), perform the following steps.

1. Fix the addressing error.
2. Clear the ERR bit in the appropriate Channel Error Status (CH0_ES - CH15_ES). Management Page Error Status (MP_ES) indicates the presence, location, and type of the last recorded channel error.
3. Start the channel in a normal fashion.

71.5.10 Suspend/resume a DMA channel with active hardware service requests

The DMA allows you to move data from memory or peripheral registers to another location in memory or to peripheral registers without CPU interaction. After the DMA and peripherals are configured and active, it is rare but supported to suspend a peripheral's service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, you must follow a specific procedure.

71.5.10.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.

2. Check the DMA's Hardware Request Status (MP_HRS) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ field to 0 on the appropriate DMA channel.

For example, assume the SPI is set as a initiator for transmitting data via a DMA service request when the TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If you need to suspend the DMA/SPI transfer loop, perform the following steps:

1. Disable the DMA service request at the SPI.
2. Ensure there is no DMA service request from the SPI by verifying that MP_HRS[HRS] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ field to 0. If a service request is present, wait until the request has been processed and the HRS field reads 0.

71.5.10.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting its ERQ field to 1.
2. Enable the DMA service request at the peripheral.

71.6 Memory map and register definition

The eDMA programming model is partitioned into three parts:

1. The first part defines a number of registers providing overall control functions and is known as the management page.
2. The second part corresponds to the channel (CH) control, status, and configuration.
3. The third part corresponds to the local TCD memory.

TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the data movement operation. Each TCDn definition is presented as 11 registers of 16 or 32 bits. See EDMA_TCD register descriptions for details.

TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

TCD structure

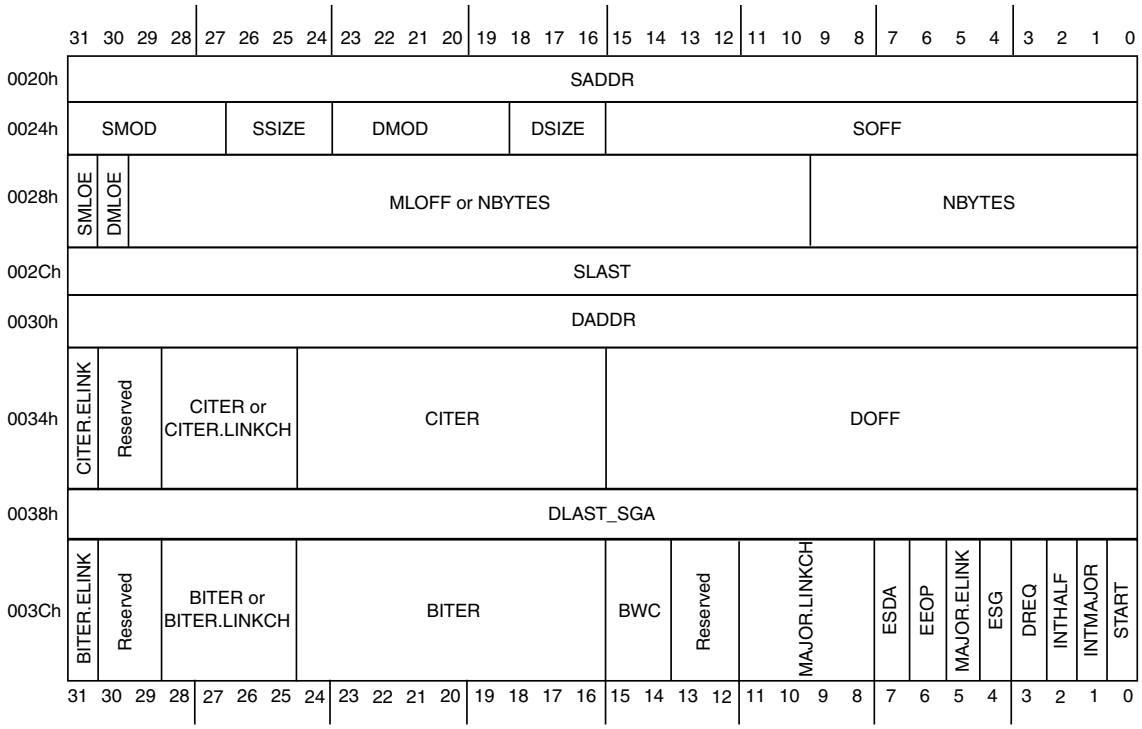


Figure 303. TCD structure

Accesses to reserved memory and fields

- Reading reserved fields in a register returns the value of zero.
- Writes to reserved fields in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

71.6.1 EDMA_MP register descriptions

71.6.1.1 EDMA_MP memory map

DMA0 base address: 4014_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0031_0000h
4h	Management Page Error Status (MP_ES)	32	R	0000_0000h
8h	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000h
100h - 13Ch	Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI)	32	RW	0000_0000h

71.6.1.2 Management Page Control (MP_CSR)

Offset

Register	Offset
MP_CSR	0h

Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ACTIVE	Reserved			ACTIVE_ID				Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CX	ECX	GMRC	GCLC	HALT	HAE	Reserved	ERCA	EDBG	Reserved
W	0												0			0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-28 —	Reserved
27-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication disabled for all channels 1b - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5	Halt DMA Operations

Table continues on the next page...

Table continued from the previous page...

Field	Function
HALT	This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate 1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

71.6.1.3 Management Page Error Status (MP_ES)

Offset

Register	Offset
MP_ES	4h

Function

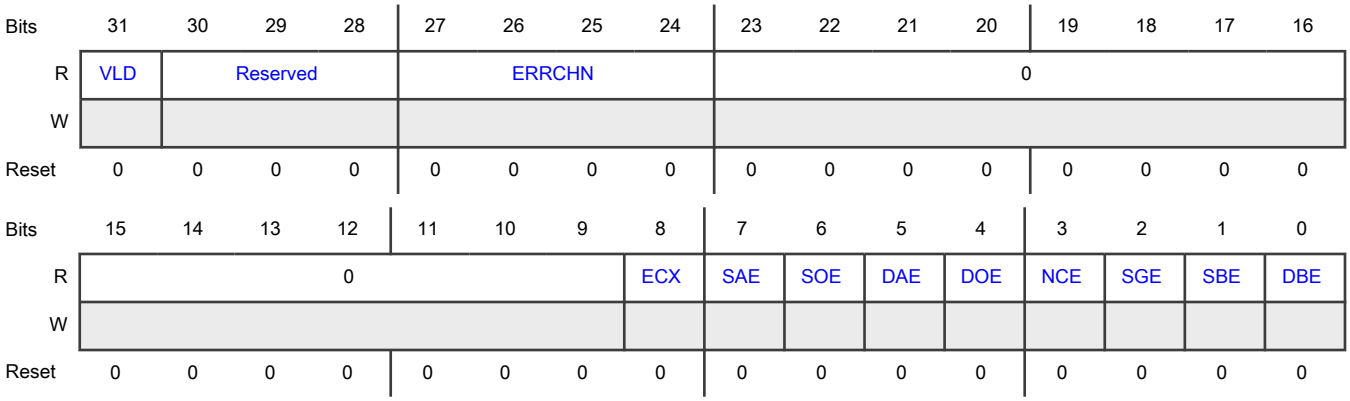
The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM

- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 VLD	Valid Logical OR of all CHn_ES[ERR] status fields. 0b - No CHn_ES[ERR] fields are set to 1 1b - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-28 —	Reserved
27-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0	Destination Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBE	0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

71.6.1.4 Management Page Interrupt Request Status (MP_INT)

Offset

Register	Offset
MP_INT	8h

Function

This register shows the current state of the interrupt service requests for all eDMA channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel’s interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

71.6.1.5 Management Page Hardware Request Status (MP_HRS)

Offset

Register	Offset
MP_HRS	Ch

Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

71.6.1.6 Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI)

Offset

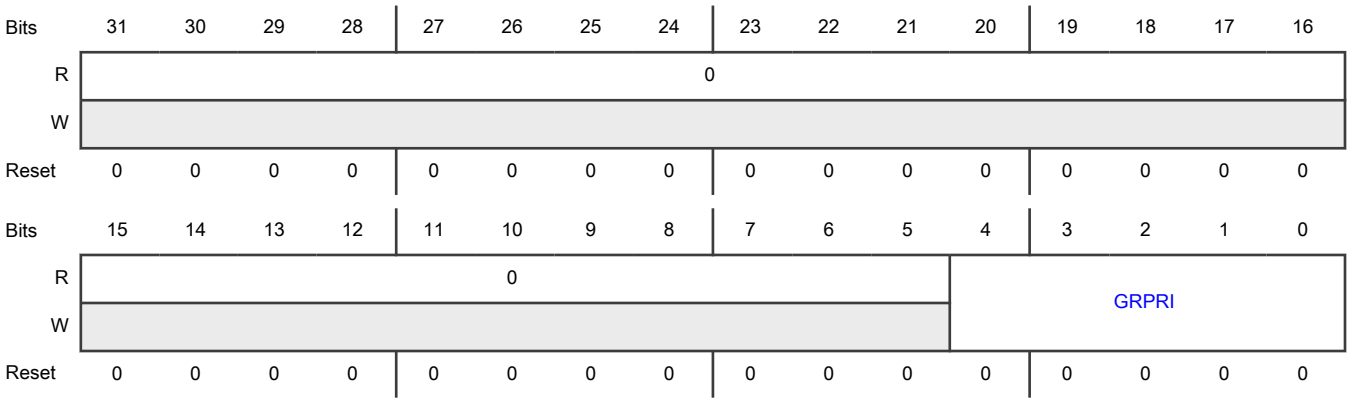
For n = 0 to 15:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

71.6.2 EDMA_MP register descriptions

71.6.2.1 EDMA_MP memory map

DMA1 base address: 4016_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0031_0000h
4h	Management Page Error Status (MP_ES)	32	R	0000_0000h
8h	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000h
100h - 13Ch	Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI)	32	RW	0000_0000h

71.6.2.2 Management Page Control (MP_CSR)

Offset

Register	Offset
MP_CSR	0h

Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ACTIVE	Reserved			ACTIVE_ID				Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CX	ECX	GMRC	GCLC	HALT	HAE	Reserved	ERCA	EDBG	Reserved
W	0												0			0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-28 —	Reserved
27-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication disabled for all channels 1b - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5	Halt DMA Operations

Table continues on the next page...

Table continued from the previous page...

Field	Function
HALT	This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate 1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

71.6.2.3 Management Page Error Status (MP_ES)

Offset

Register	Offset
MP_ES	4h

Function

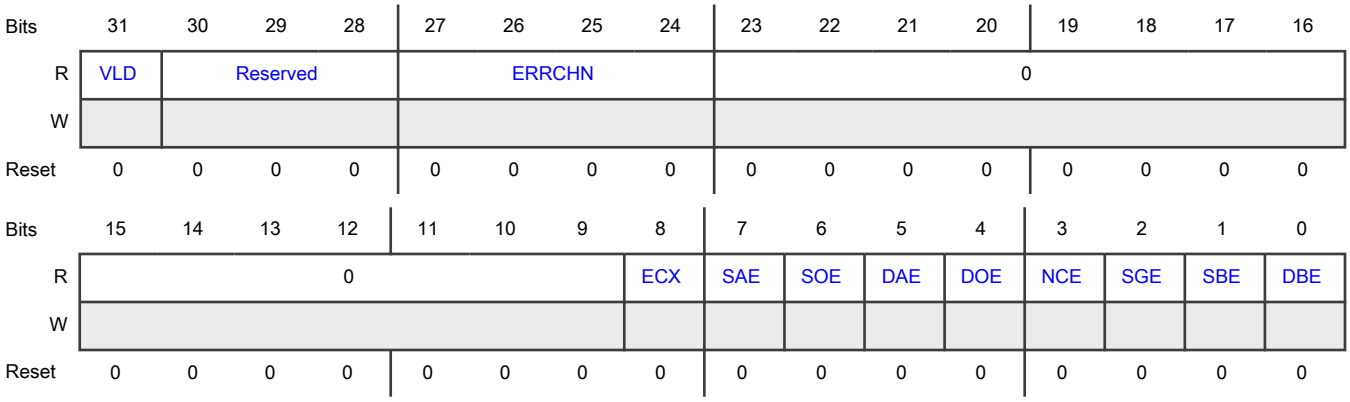
The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM

- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 VLD	Valid Logical OR of all CHn_ES[ERR] status fields. 0b - No CHn_ES[ERR] fields are set to 1 1b - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-28 —	Reserved
27-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0	Destination Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBE	0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

71.6.2.4 Management Page Interrupt Request Status (MP_INT)

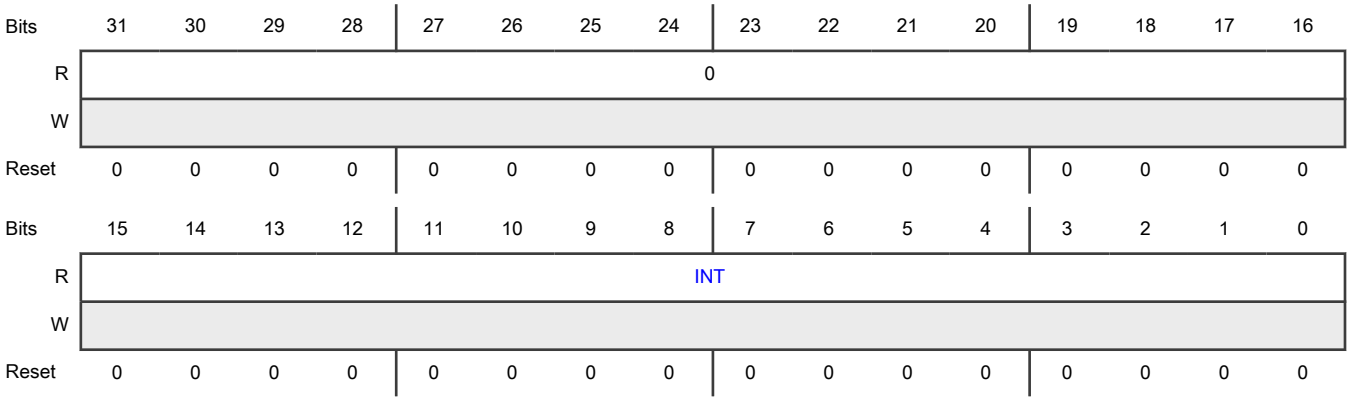
Offset

Register	Offset
MP_INT	8h

Function

This register shows the current state of the interrupt service requests for all eDMA channels.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

71.6.2.5 Management Page Hardware Request Status (MP_HRS)

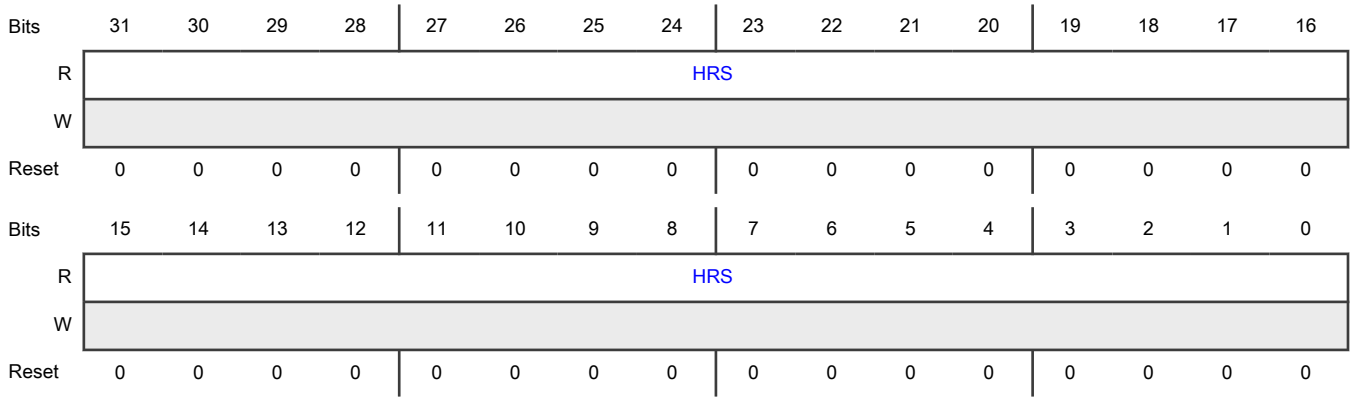
Offset

Register	Offset
MP_HRS	Ch

Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

Diagram



Fields

Field	Function
31-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

71.6.2.6 Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI)

Offset

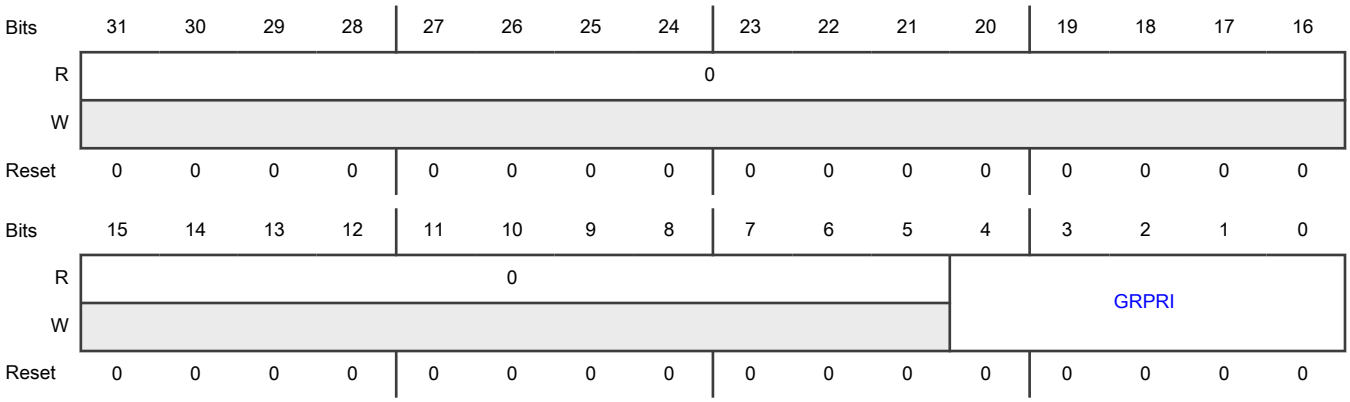
For n = 0 to 15:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

71.6.3 EDMA_MP register descriptions

71.6.3.1 EDMA_MP memory map

DMA2 base address: 4030_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0031_0000h
4h	Management Page Error Status (MP_ES)	32	R	0000_0000h
8h	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000h
100h - 11Ch	Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI)	32	RW	0000_0000h

71.6.3.2 Management Page Control (MP_CSR)

Offset

Register	Offset
MP_CSR	0h

Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI).

NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ACTIVE	Reserved				ACTIVE_ID			Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CX	ECX	GMRC	GCLC	HALT	HAE	Reserved	ERCA	EDBG	Reserved
W	0												0			0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-27 —	Reserved
26-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication disabled for all channels 1b - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5	Halt DMA Operations

Table continues on the next page...

Table continued from the previous page...

Field	Function
HALT	This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate 1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

71.6.3.3 Management Page Error Status (MP_ES)

Offset

Register	Offset
MP_ES	4h

Function

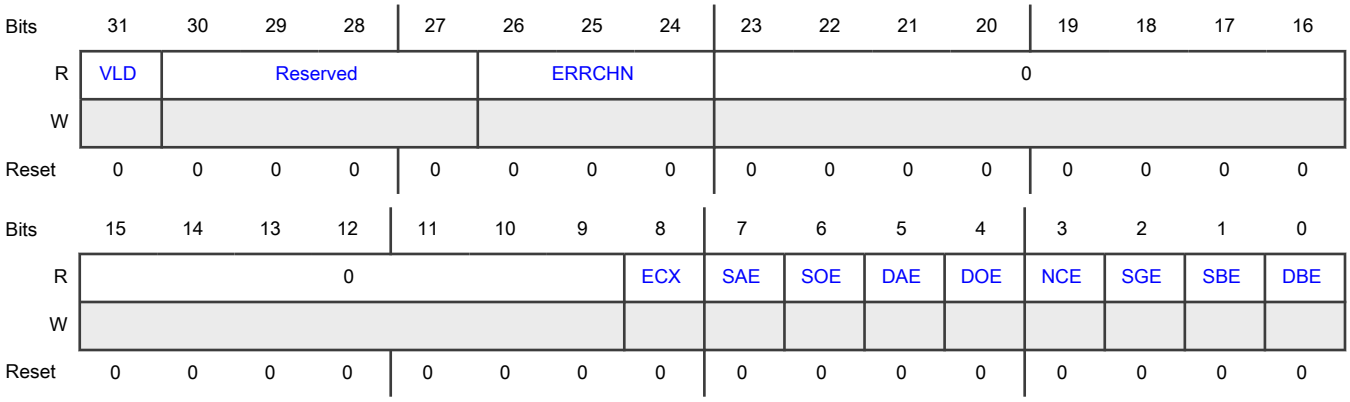
The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM

- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 VLD	Valid Logical OR of all CHn_ES[ERR] status fields. 0b - No CHn_ES[ERR] fields are set to 1 1b - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-27 —	Reserved
26-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0	Destination Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBE	0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

71.6.3.4 Management Page Interrupt Request Status (MP_INT)

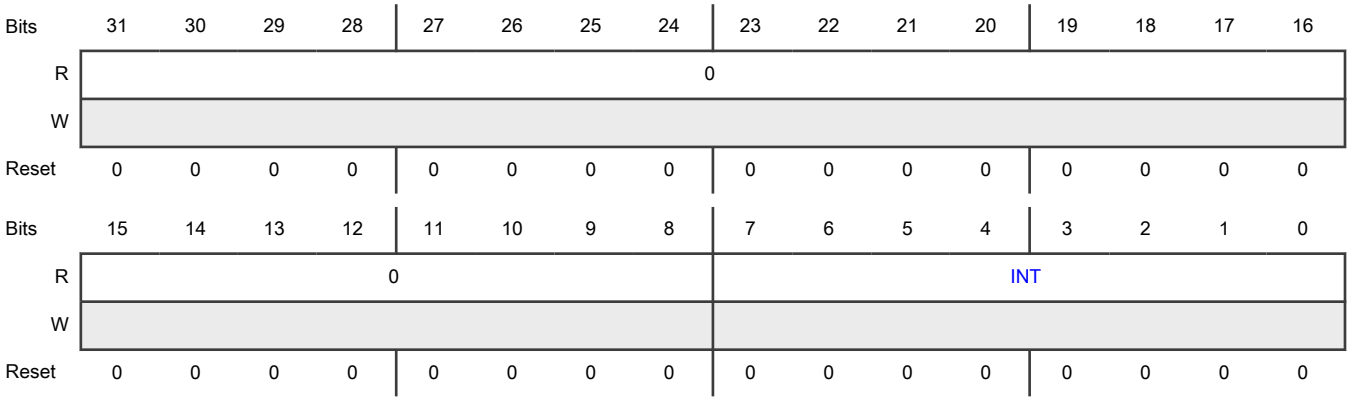
Offset

Register	Offset
MP_INT	8h

Function

This register shows the current state of the interrupt service requests for all eDMA channels.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

71.6.3.5 Management Page Hardware Request Status (MP_HRS)

Offset

Register	Offset
MP_HRS	Ch

Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

71.6.3.6 Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI)

Offset

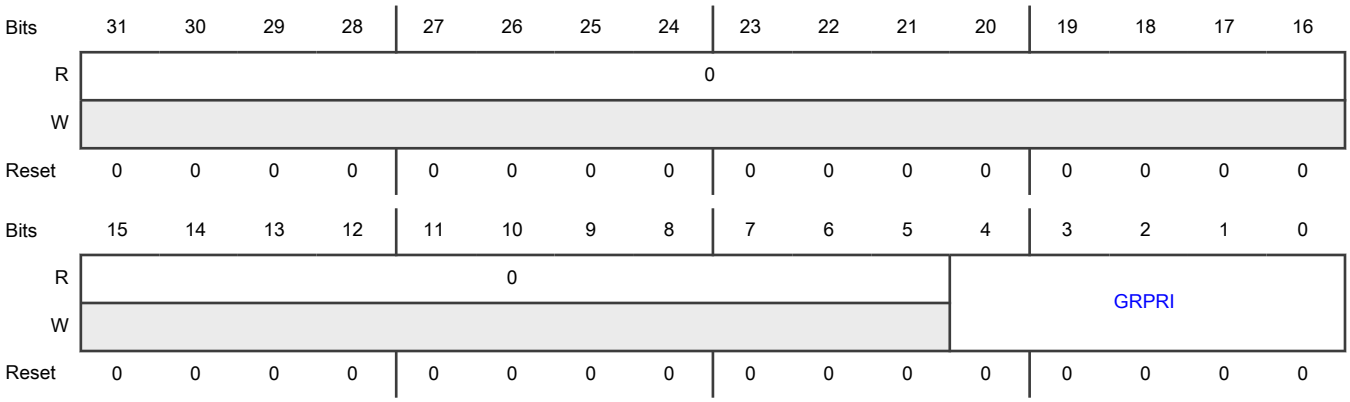
For n = 0 to 7:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

71.6.4 EDMA_MP register descriptions

71.6.4.1 EDMA_MP memory map

DMA3 base address: 4031_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0031_0000h
4h	Management Page Error Status (MP_ES)	32	R	0000_0000h
8h	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000h
100h - 11Ch	Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI)	32	RW	0000_0000h

71.6.4.2 Management Page Control (MP_CSR)

Offset

Register	Offset
MP_CSR	0h

Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI).

NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ACTIVE	Reserved				ACTIVE_ID			Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CX	ECX	GMRC	GCLC	HALT	HAE	Reserved	ERCA	EDBG	Reserved
W	0												0			0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel

Table continues on the next page...

Table continued from the previous page...

Field	Function
30-27 —	Reserved
26-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication disabled for all channels 1b - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5	Halt DMA Operations

Table continues on the next page...

Table continued from the previous page...

Field	Function
HALT	This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate 1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

71.6.4.3 Management Page Error Status (MP_ES)

Offset

Register	Offset
MP_ES	4h

Function

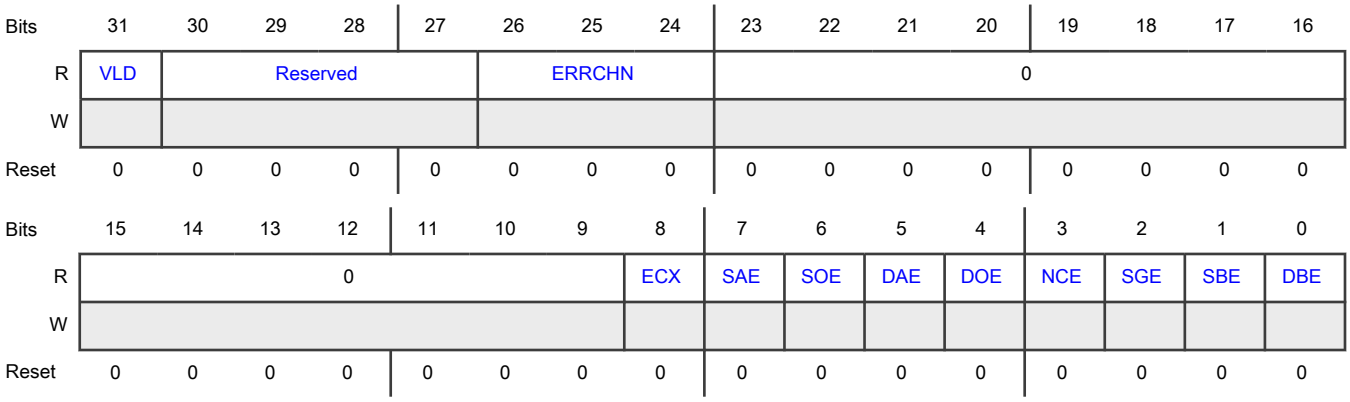
The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM

- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 VLD	Valid Logical OR of all CHn_ES[ERR] status fields. 0b - No CHn_ES[ERR] fields are set to 1 1b - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-27 —	Reserved
26-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0	Destination Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Function
DBE	0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

71.6.4.4 Management Page Interrupt Request Status (MP_INT)

Offset

Register	Offset
MP_INT	8h

Function

This register shows the current state of the interrupt service requests for all eDMA channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel’s interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

71.6.4.5 Management Page Hardware Request Status (MP_HRS)

Offset

Register	Offset
MP_HRS	Ch

Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

71.6.4.6 Channel Arbitration Group (CH0_GRPRI - CH7_GRPRI)

Offset

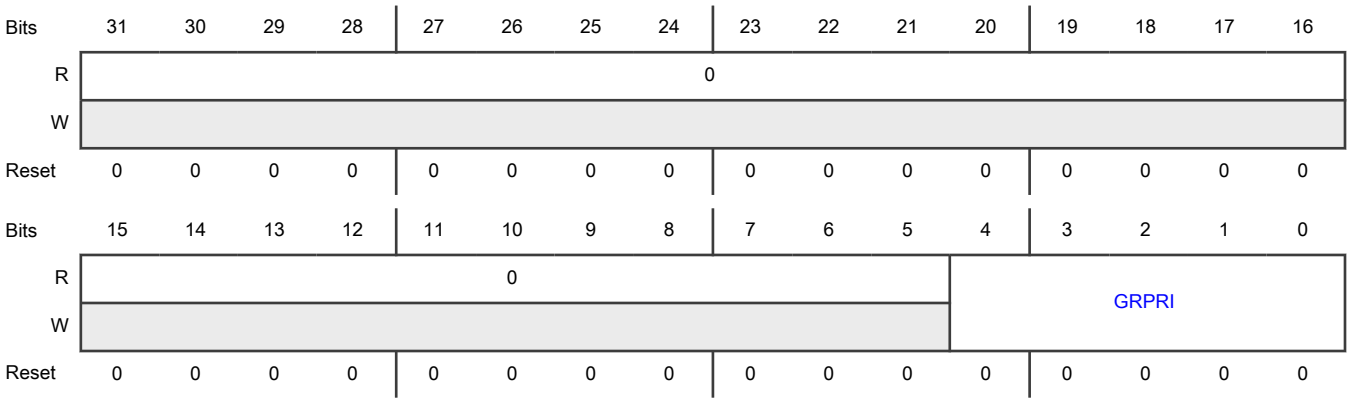
For n = 0 to 7:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

71.6.5 EDMA_TCD register descriptions

71.6.5.1 EDMA_TCD memory map

DMA0_TCD base address: 4014_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - F000h	Channel Control and Status (CH0_CSR - CH15_CSR)	32	RW	0000_0000h
4h - F004h	Channel Error Status (CH0_ES - CH15_ES)	32	RW	0000_0000h
8h - F008h	Channel Interrupt Status (CH0_INT - CH15_INT)	32	RW	0000_0000h
Ch - F00Ch	Channel System Bus (CH0_SBR - CH15_SBR)	32	RW	0000_8001h
10h - F010h	Channel Priority (CH0_PRI - CH15_PRI)	32	RW	0000_0000h
14h - F014h	Channel Multiplexor Configuration (CH0_MUX - CH15_MUX)	32	RW	0000_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
20h - F020h	TCD Source Address (TCD0_SADDR - TCD15_SADDR)	32	RW	See section
24h - F024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD15_SOFF)	16	RW	See section
26h - F026h	TCD Transfer Attributes (TCD0_ATTR - TCD15_ATTR)	16	RW	See section
28h - F028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD15_NBYTES_MLOFFNO)	32	RW	See section
28h - F028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD15_NBYTES_MLOFFYES)	32	RW	See section
2Ch - F02Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD15_SLAST_SDA)	32	RW	See section
30h - F030h	TCD Destination Address (TCD0_DADDR - TCD15_DADDR)	32	RW	See section
34h - F034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD15_DOFF)	16	RW	See section
36h - F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD15_CITER_ELINKNO)	16	RW	See section
36h - F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD15_CITER_ELINKYES)	16	RW	See section
38h - F038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD15_DLAST_SGA)	32	RW	See section
3Ch - F03Ch	TCD Control and Status (TCD0_CSR - TCD15_CSR)	16	RW	See section
3Eh - F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD15_BITER_ELINKNO)	16	RW	See section
3Eh - F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD15_BITER_ELINKYES)	16	RW	See section

71.6.5.2 Channel Control and Status (CH0_CSR - CH15_CSR)

Offset

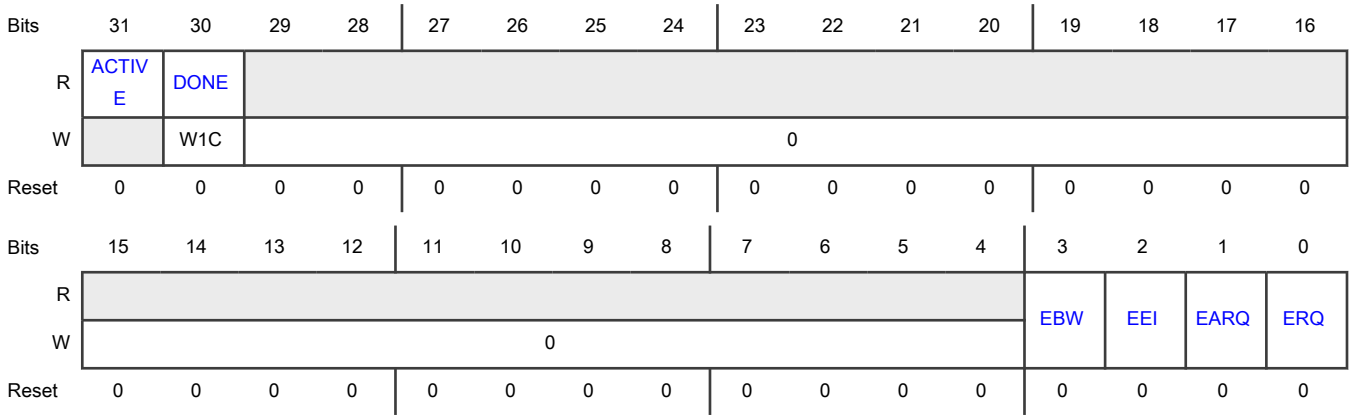
For n = 0 to 15:

Register	Offset
CHn_CSR	0h + (n × 1000h)

Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

Diagram



Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

71.6.5.3 Channel Error Status (CH0_ES - CH15_ES)

Offset

For n = 0 to 15:

Register	Offset
CHn_ES	4h + (n × 1000h)

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

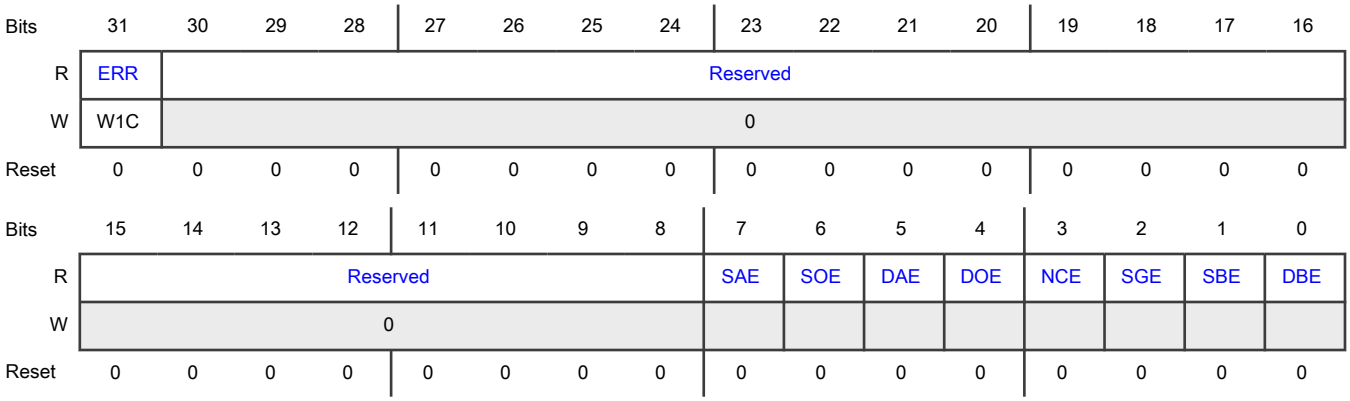
During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
DOE	0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

71.6.5.4 Channel Interrupt Status (CH0_INT - CH15_INT)

Offset

For n = 0 to 15:

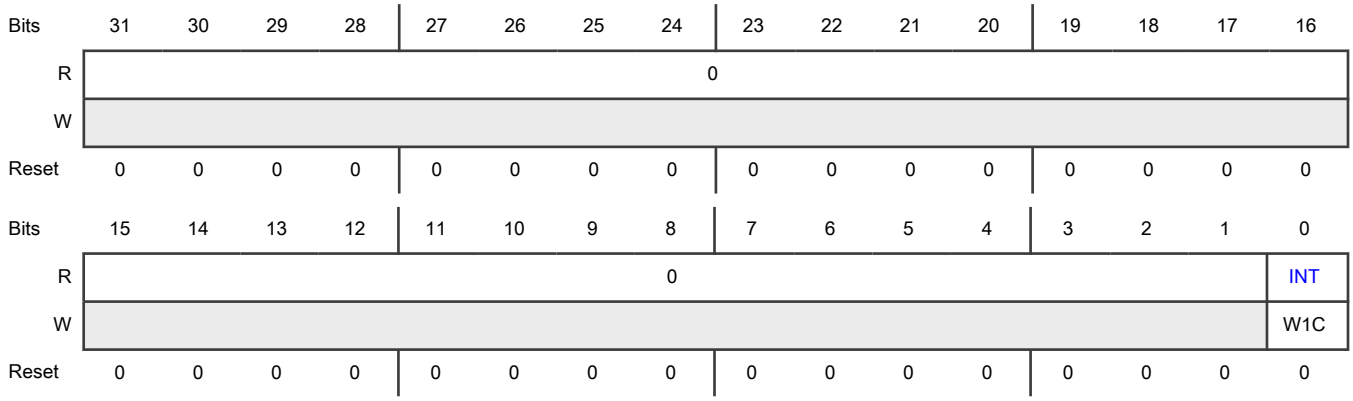
Register	Offset
CHn_INT	8h + (n × 1000h)

Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

71.6.5.5 Channel System Bus (CH0_SBR - CH15_SBR)

Offset

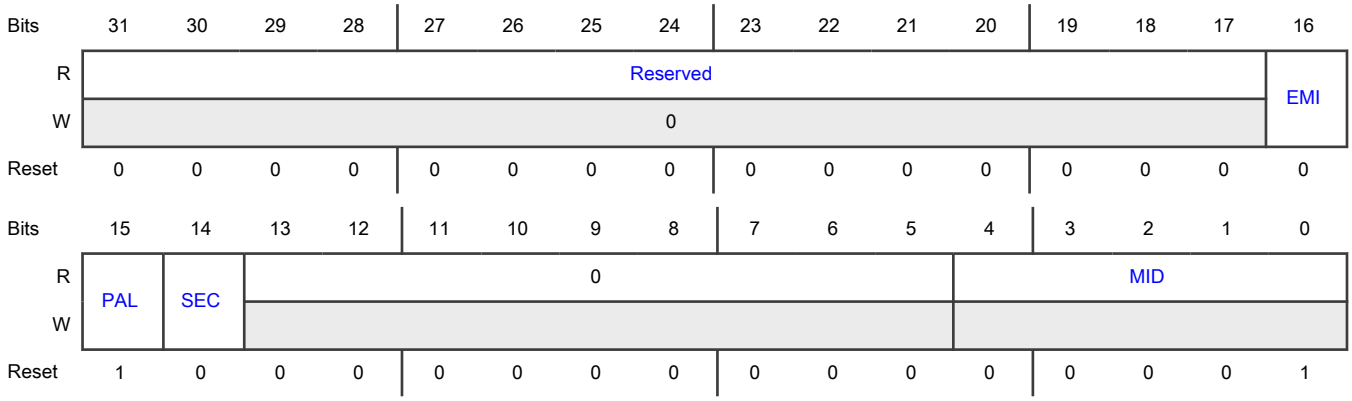
For n = 0 to 15:

Register	Offset
CHn_SBR	Ch + (n × 1000h)

Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication is disabled 1b - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14 SEC	Security Level DMA's security level on the system bus when the channel is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - Nonsecure protection level for DMA transfers 1b - Secure protection level for DMA transfers
13-5 —	Reserved
4-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's security attributes, TCD*n*_SBR[SEC].

71.6.5.6 Channel Priority (CH0_PRI - CH15_PRI)

Offset

For n = 0 to 15:

Register	Offset
CHn_PRI	10h + (n × 1000h)

Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

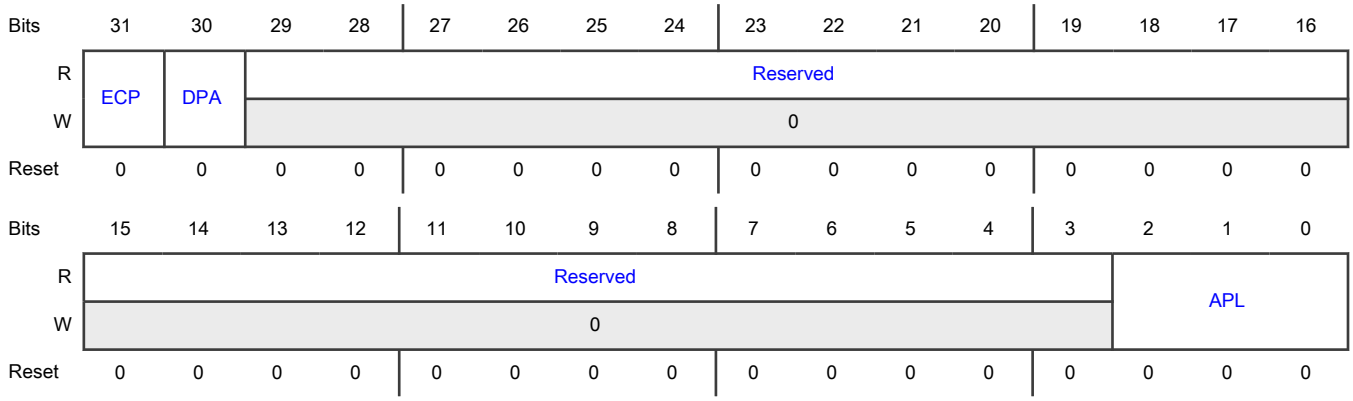
The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Diagram



Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

71.6.5.7 Channel Multiplexor Configuration (CH0_MUX - CH15_MUX)

Offset

For n = 0 to 15:

Register	Offset
CHn_MUX	14h + (n × 1000h)

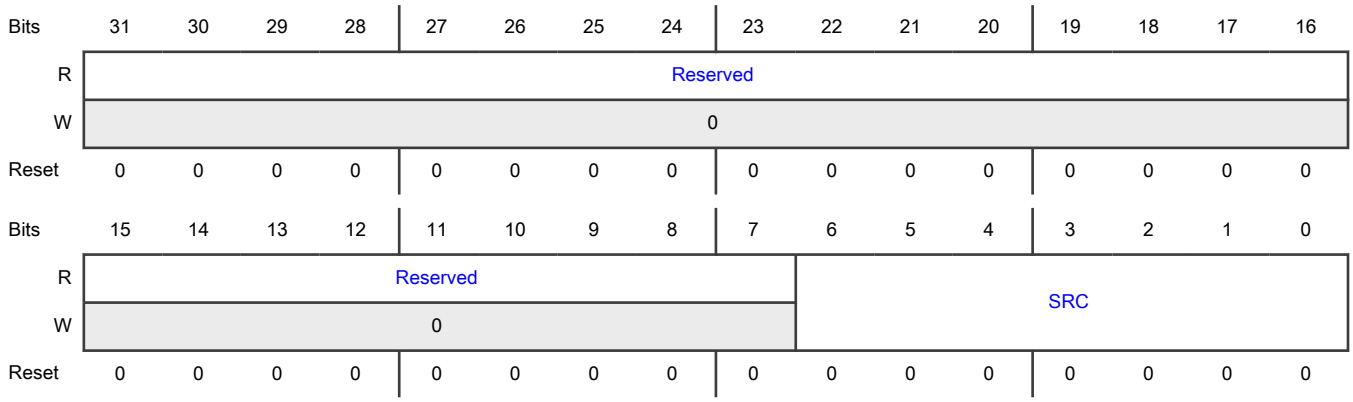
Function

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by any channel, a mux configuration of 0 (SRC = 0) will be written.

All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

71.6.5.8 TCD Source Address (TCD0_SADDR - TCD15_SADDR)

Offset

For n = 0 to 15:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

Function

This register contains the address for the read transactions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

71.6.5.9 TCD Signed Source Address Offset (TCD0_SOFF - TCD15_SOFF)

Offset

For n = 0 to 15:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

Function

This register contains the sign-extended value added to Source Address register after each read transaction.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

71.6.5.10 TCD Transfer Attributes (TCD0_ATTR - TCD15_ATTR)

Offset

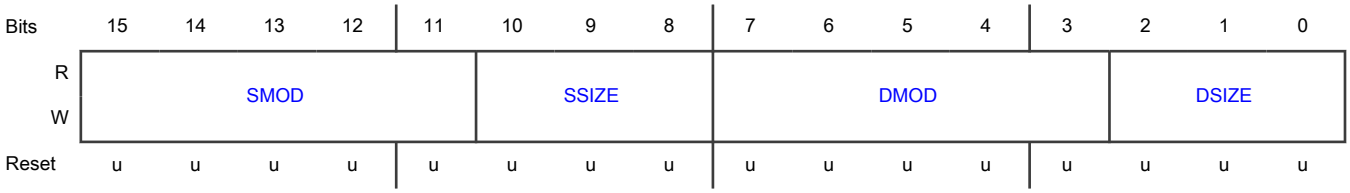
For n = 0 to 15:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

Function

This register contains size and option modulo addressing information for source and destination addresses.

Diagram



Fields

Field	Function
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000b - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - 16-byte 101b - 32-byte 110b - 64-byte 111b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

71.6.5.11 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD15_NBYTES_MLOFFNO)

Offset

For n = 0 to 15:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

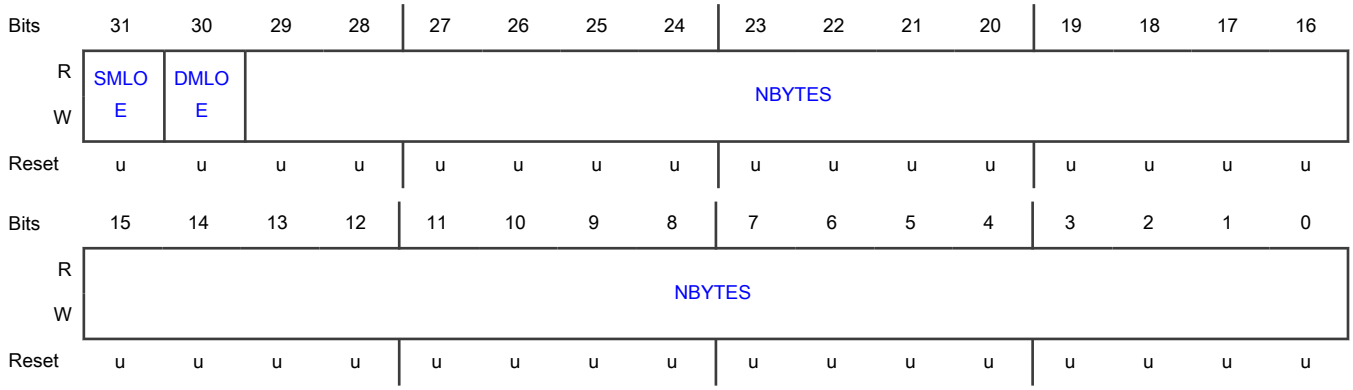
The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.5.12 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD15_NBYTES_MLOFFYES)

Offset

For n = 0 to 15:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

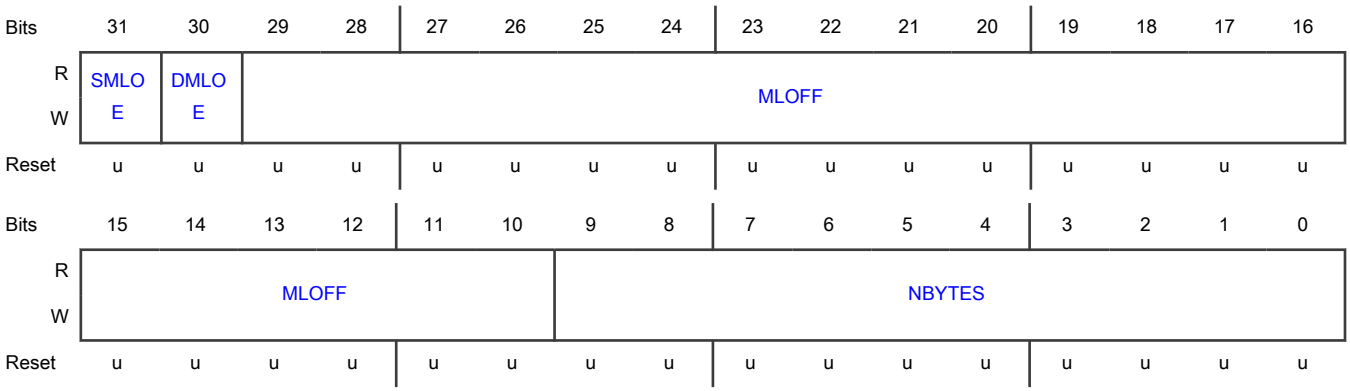
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.5.13 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD15_SLAST_SDA)

Offset

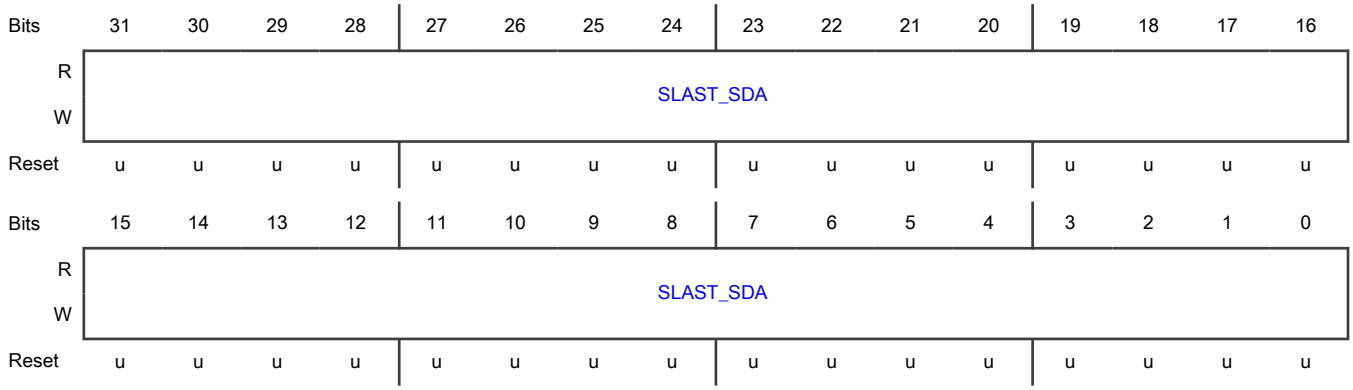
For n = 0 to 15:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

Function

This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Diagram



Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

71.6.5.14 TCD Destination Address (TCD0_DADDR - TCD15_DADDR)

Offset

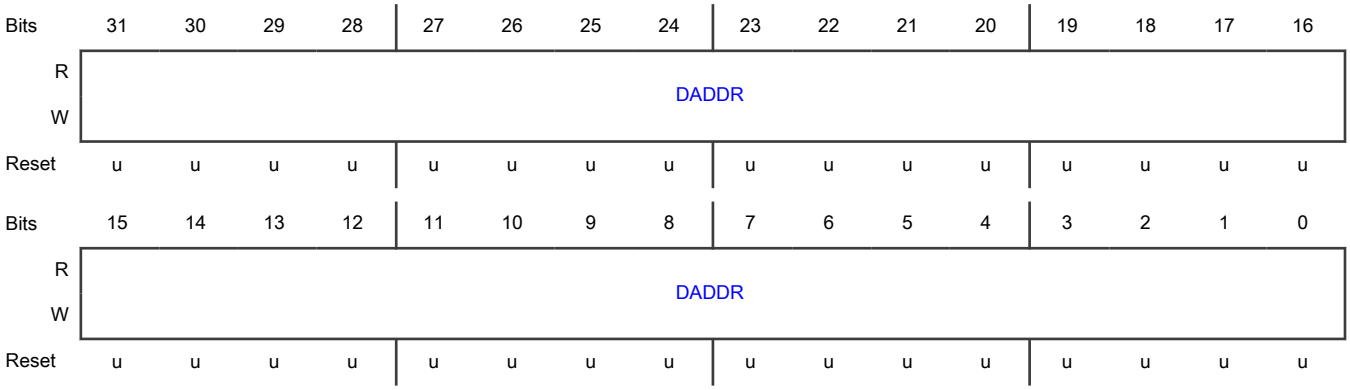
For n = 0 to 15:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

Function

This register contains the address for the write transactions.

Diagram



Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

71.6.5.15 TCD Signed Destination Address Offset (TCD0_DOFF - TCD15_DOFF)

Offset

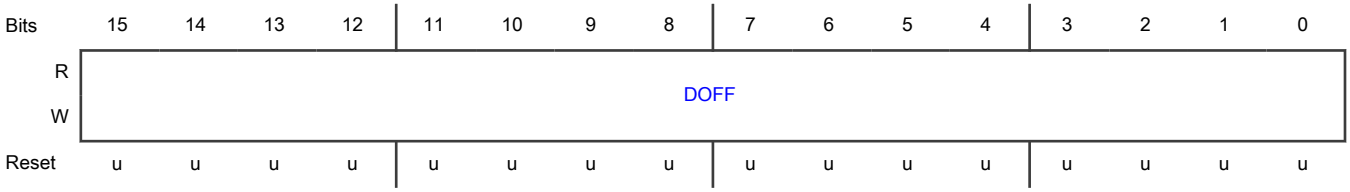
For n = 0 to 15:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

Function

This register contains the sign-extended value added to Destination Address register after each write transaction.

Diagram



Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

71.6.5.16 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD15_CITER_ELINKNO)

Offset

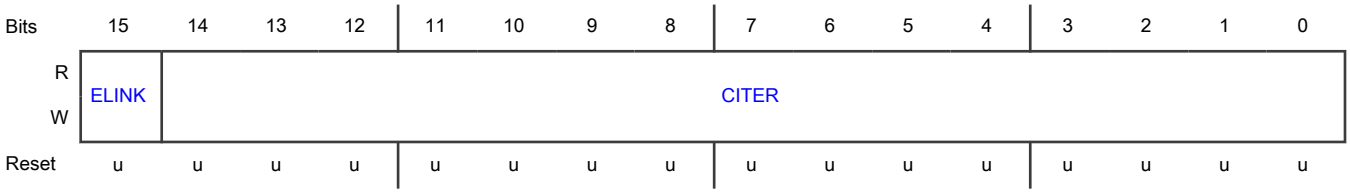
For n = 0 to 15:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.5.17 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD15_CITER_ELINKYES)

Offset

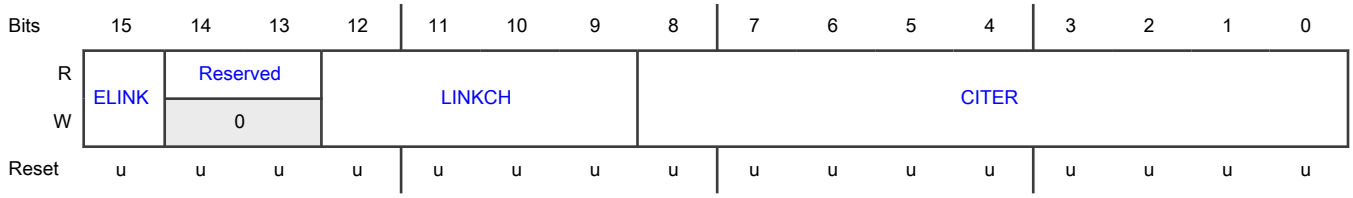
For n = 0 to 15:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-13 —	Reserved
12-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.5.18 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD15_DLAST_SGA)

Offset

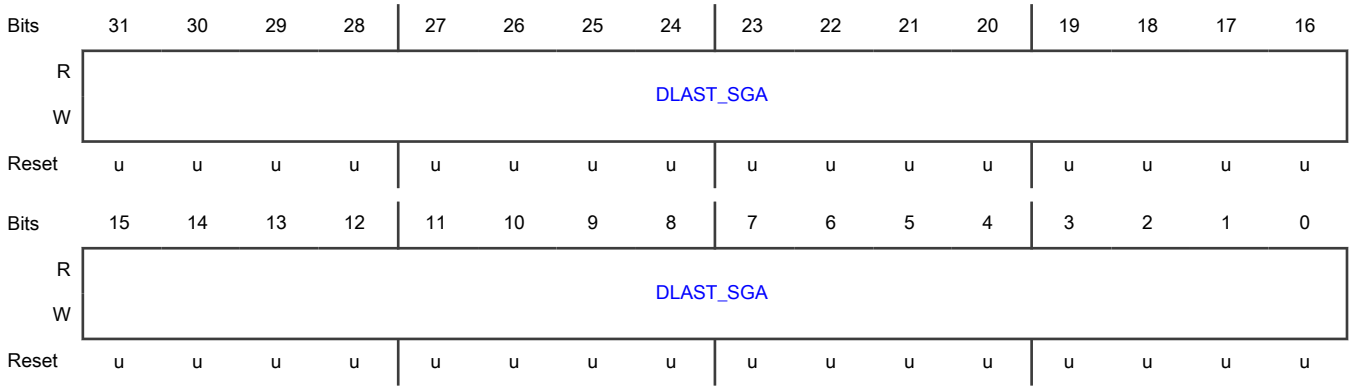
For n = 0 to 15:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

Function

This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Diagram



Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

71.6.5.19 TCD Control and Status (TCD0_CSR - TCD15_CSR)

Offset

For n = 0 to 15:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

Function

This register is used to enable optional features.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BWC				MAJORLINKCH				ESDA	EEOP	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
W			0													
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0

Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13-12 —	Reserved
11-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2	Enable Interrupt If Major Counter Half-complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
INTHALF	If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

71.6.5.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD15_BITER_ELINKNO)

Offset

For n = 0 to 15:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Diagram

Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.5.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD15_BITER_ELINKYES)

Offset

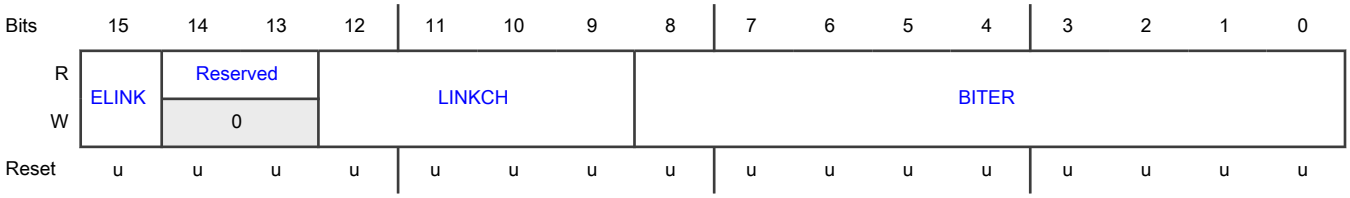
For n = 0 to 15:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-13 —	Reserved
12-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.6 EDMA_TCD register descriptions

71.6.6.1 EDMA_TCD memory map

DMA1_TCD base address: 4016_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - F000h	Channel Control and Status (CH0_CSR - CH15_CSR)	32	RW	0000_0000h
4h - F004h	Channel Error Status (CH0_ES - CH15_ES)	32	RW	0000_0000h
8h - F008h	Channel Interrupt Status (CH0_INT - CH15_INT)	32	RW	0000_0000h
Ch - F00Ch	Channel System Bus (CH0_SBR - CH15_SBR)	32	RW	0000_8002h
10h - F010h	Channel Priority (CH0_PRI - CH15_PRI)	32	RW	0000_0000h
14h - F014h	Channel Multiplexor Configuration (CH0_MUX - CH15_MUX)	32	RW	0000_0000h
20h - F020h	TCD Source Address (TCD0_SADDR - TCD15_SADDR)	32	RW	See section
24h - F024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD15_SOFF)	16	RW	See section
26h - F026h	TCD Transfer Attributes (TCD0_ATTR - TCD15_ATTR)	16	RW	See section
28h - F028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD15_NBYTES_MLOFFNO)	32	RW	See section
28h - F028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD15_NBYTES_MLOFFYES)	32	RW	See section
2Ch - F02Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD15_SLAST_SDA)	32	RW	See section
30h - F030h	TCD Destination Address (TCD0_DADDR - TCD15_DADDR)	32	RW	See section
34h - F034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD15_DOFF)	16	RW	See section
36h - F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD15_CITER_ELINKNO)	16	RW	See section
36h - F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD15_CITER_ELINKYES)	16	RW	See section
38h - F038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD15_DLAST_SGA)	32	RW	See section
3Ch - F03Ch	TCD Control and Status (TCD0_CSR - TCD15_CSR)	16	RW	See section
3Eh - F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD15_BITER_ELINKNO)	16	RW	See section
3Eh - F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD15_BITER_ELINKYES)	16	RW	See section

71.6.6.2 Channel Control and Status (CH0_CSR - CH15_CSR)

Offset

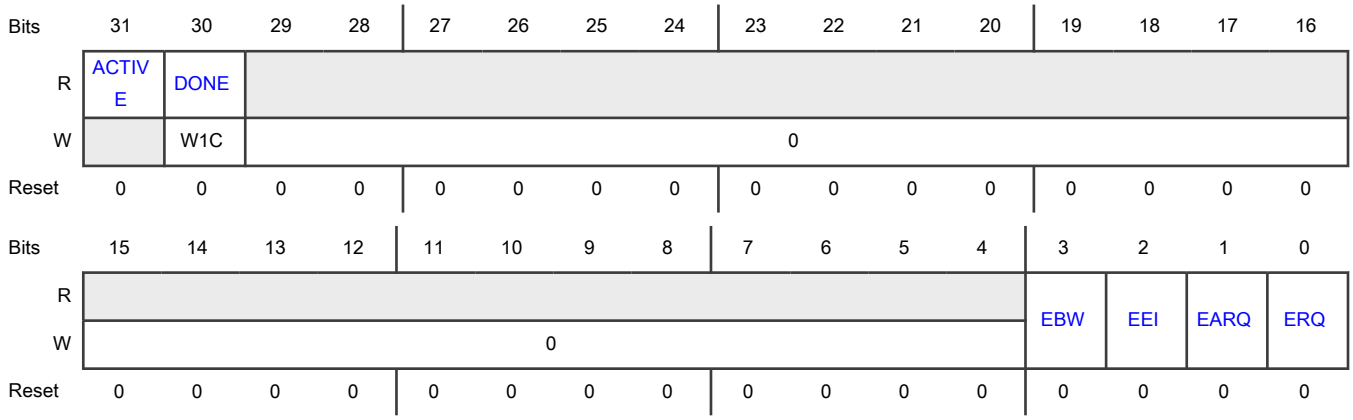
For n = 0 to 15:

Register	Offset
CHn_CSR	0h + (n × 1000h)

Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

Diagram



Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt 1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

71.6.6.3 Channel Error Status (CH0_ES - CH15_ES)

Offset

For n = 0 to 15:

Register	Offset
CHn_ES	4h + (n × 1000h)

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The

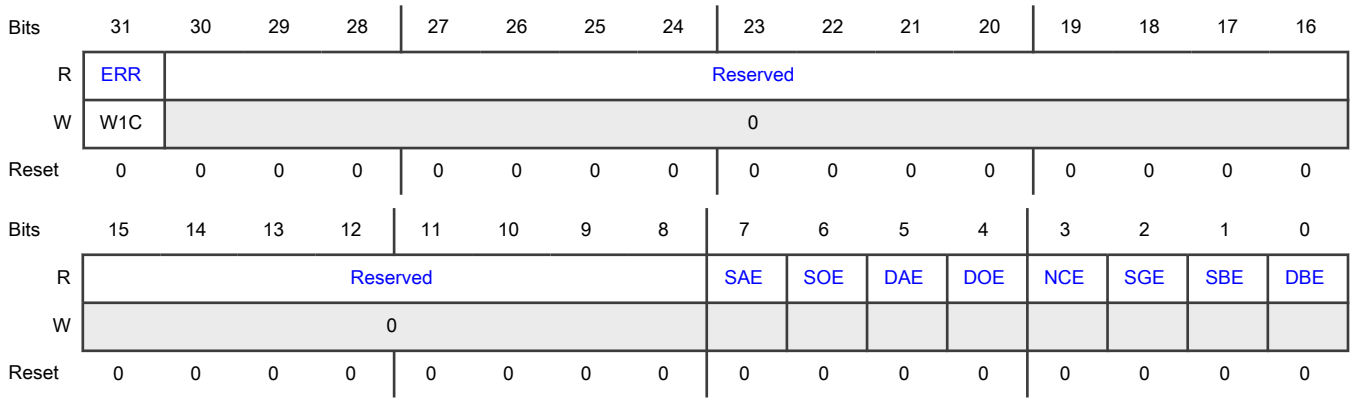
contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

71.6.6.4 Channel Interrupt Status (CH0_INT - CH15_INT)

Offset

For n = 0 to 15:

Register	Offset
CHn_INT	8h + (n × 1000h)

Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

71.6.6.5 Channel System Bus (CH0_SBR - CH15_SBR)

Offset

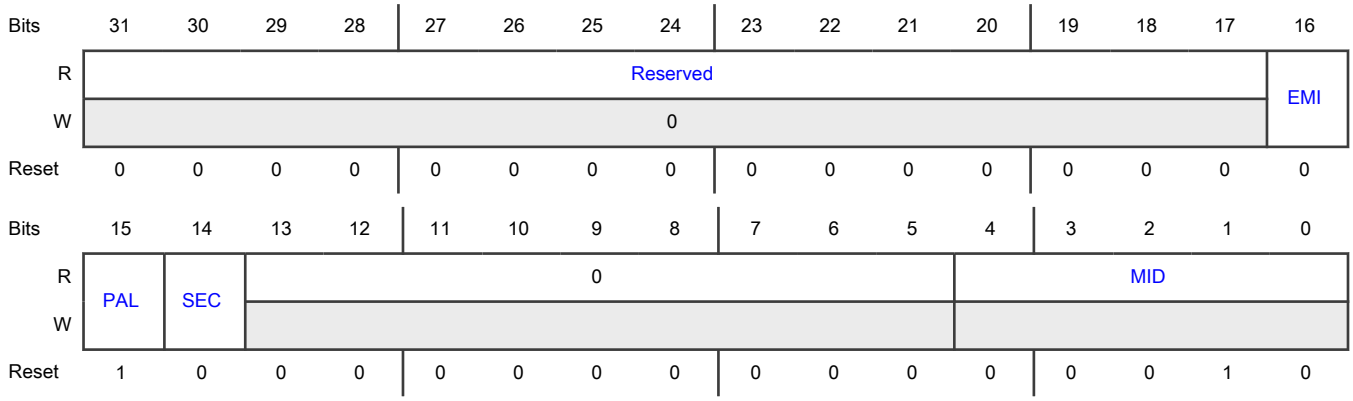
For n = 0 to 15:

Register	Offset
CHn_SBR	Ch + (n × 1000h)

Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication is disabled 1b - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14 SEC	Security Level DMA's security level on the system bus when the channel is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - Nonsecure protection level for DMA transfers 1b - Secure protection level for DMA transfers
13-5 —	Reserved
4-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's security attributes, TCD*n*_SBR[SEC].

71.6.6.6 Channel Priority (CH0_PRI - CH15_PRI)

Offset

For n = 0 to 15:

Register	Offset
CHn_PRI	10h + (n × 1000h)

Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

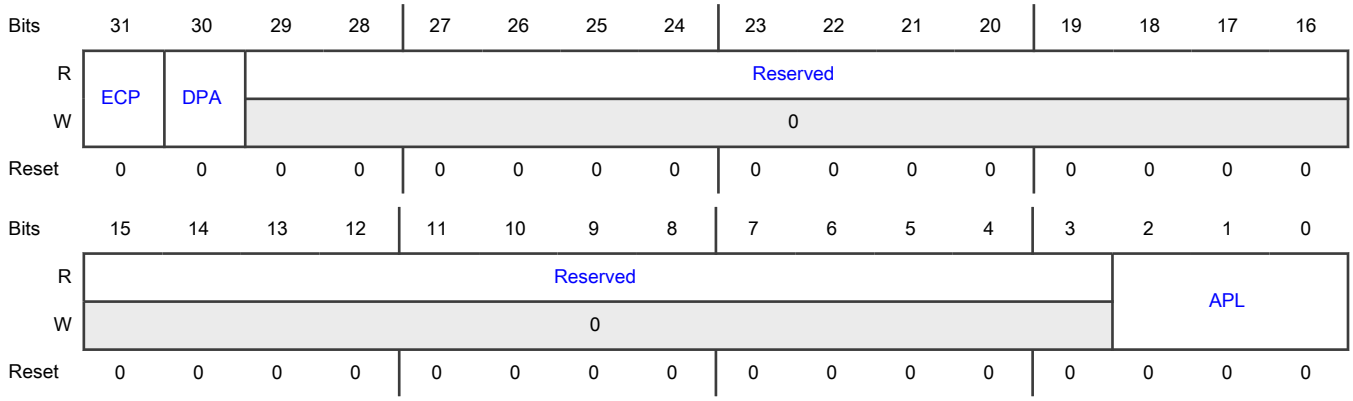
The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Diagram



Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

71.6.6.7 Channel Multiplexor Configuration (CH0_MUX - CH15_MUX)

Offset

For n = 0 to 15:

Register	Offset
CHn_MUX	14h + (n × 1000h)

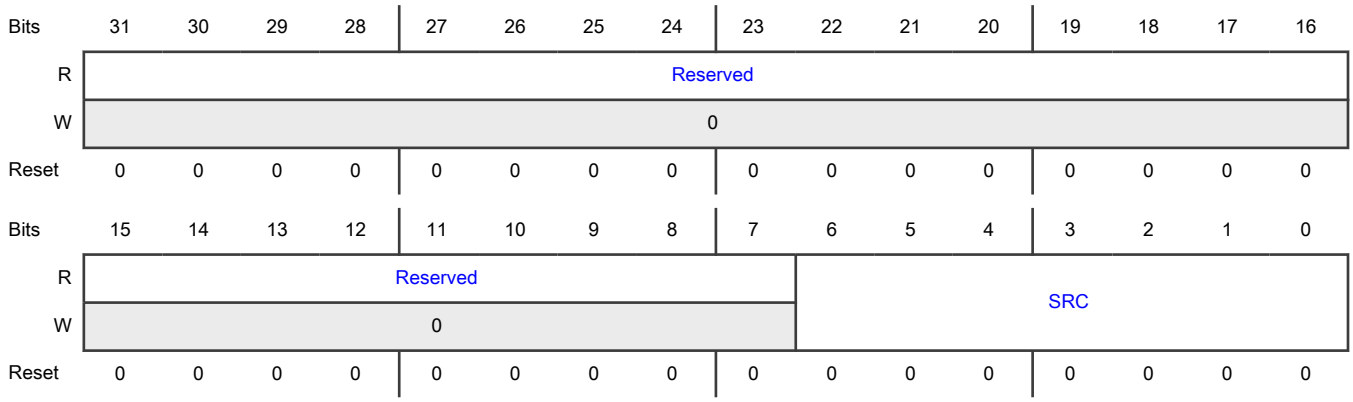
Function

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by any channel, a mux configuration of 0 (SRC = 0) will be written.

All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

71.6.6.8 TCD Source Address (TCD0_SADDR - TCD15_SADDR)

Offset

For n = 0 to 15:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

Function

This register contains the address for the read transactions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SADDR															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SADDR															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

71.6.6.9 TCD Signed Source Address Offset (TCD0_SOFF - TCD15_SOFF)

Offset

For n = 0 to 15:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

Function

This register contains the sign-extended value added to Source Address register after each read transaction.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SOFF															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

71.6.6.10 TCD Transfer Attributes (TCD0_ATTR - TCD15_ATTR)

Offset

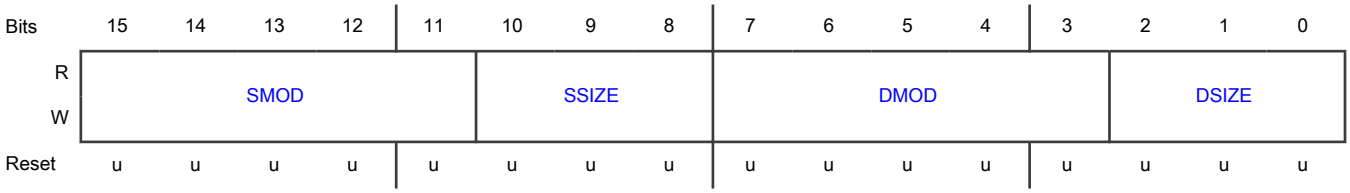
For n = 0 to 15:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

Function

This register contains size and option modulo addressing information for source and destination addresses.

Diagram



Fields

Field	Function
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000b - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - 16-byte 101b - 32-byte 110b - 64-byte 111b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

71.6.6.11 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD15_NBYTES_MLOFFNO)

Offset

For n = 0 to 15:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

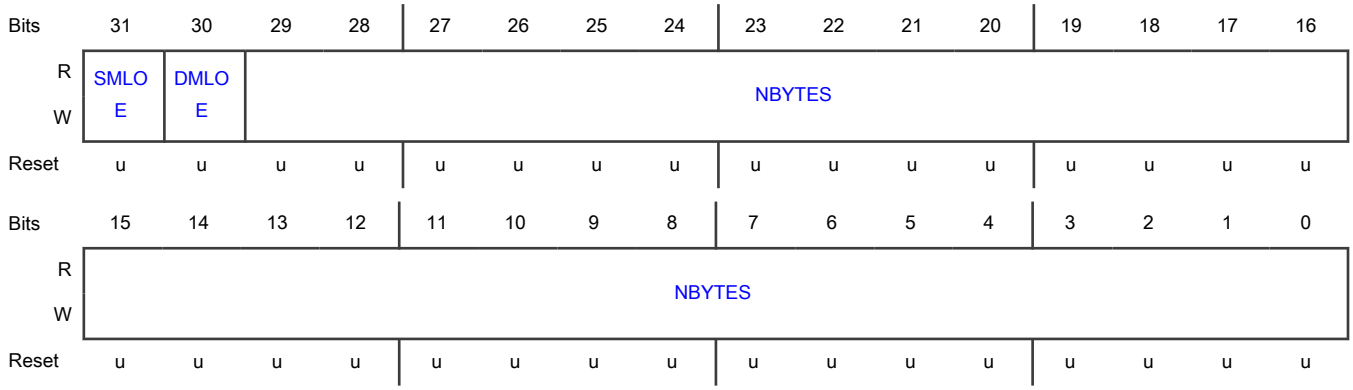
The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.6.12 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD15_NBYTES_MLOFFYES)

Offset

For n = 0 to 15:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

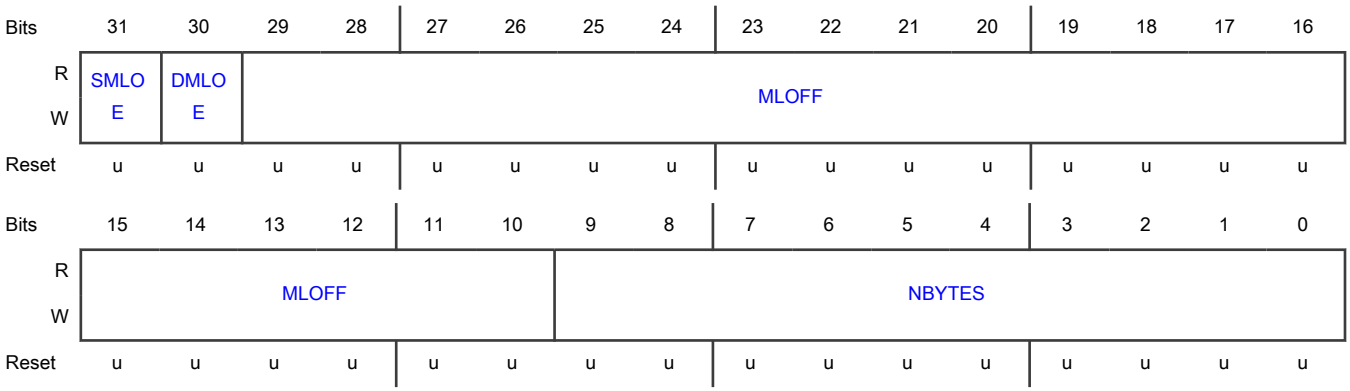
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.6.13 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD15_SLAST_SDA)

Offset

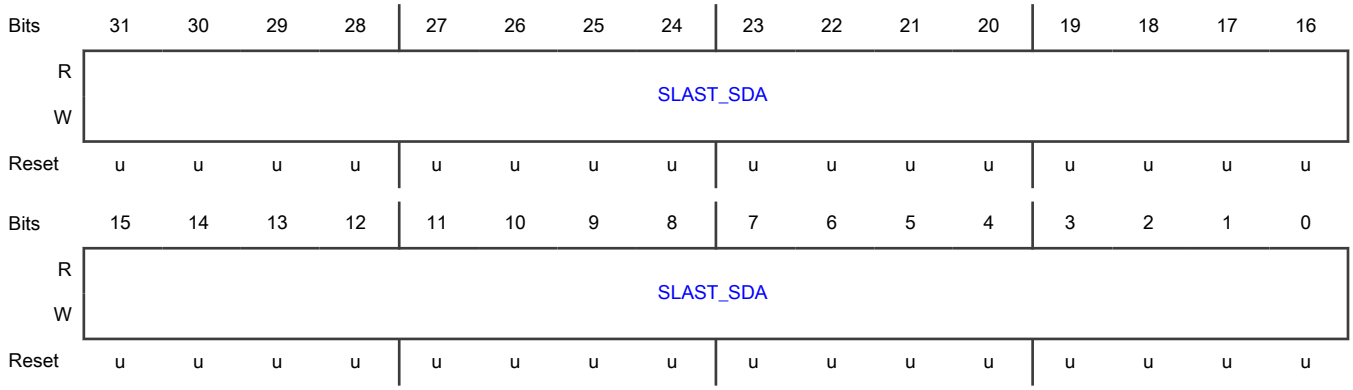
For n = 0 to 15:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

Function

This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Diagram



Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

71.6.6.14 TCD Destination Address (TCD0_DADDR - TCD15_DADDR)

Offset

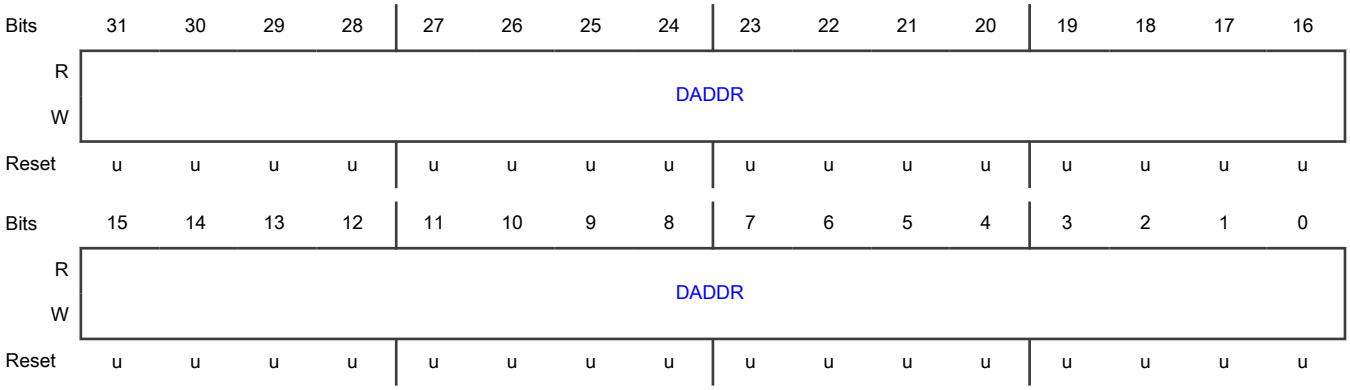
For n = 0 to 15:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

Function

This register contains the address for the write transactions.

Diagram



Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

71.6.6.15 TCD Signed Destination Address Offset (TCD0_DOFF - TCD15_DOFF)

Offset

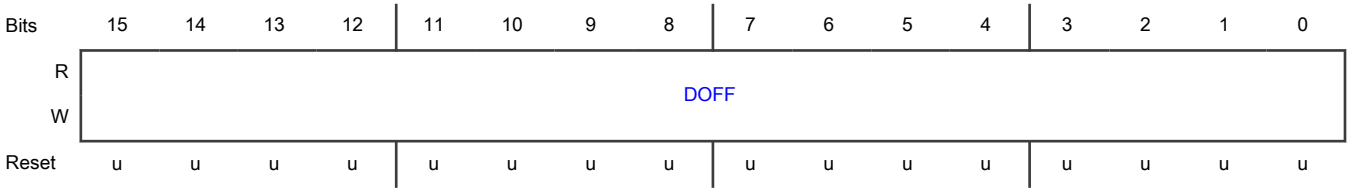
For n = 0 to 15:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

Function

This register contains the sign-extended value added to Destination Address register after each write transaction.

Diagram



Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

71.6.6.16 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD15_CITER_ELINKNO)

Offset

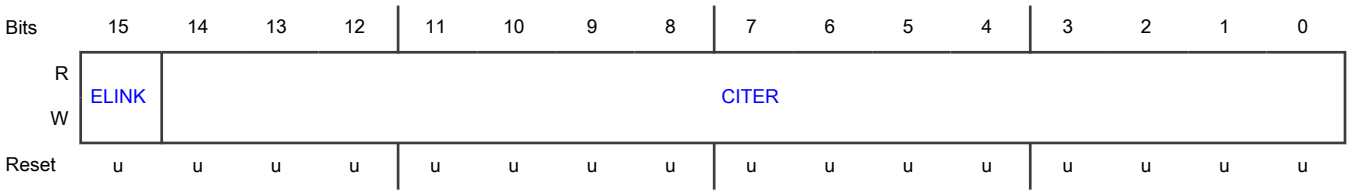
For n = 0 to 15:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.6.17 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD15_CITER_ELINKYES)

Offset

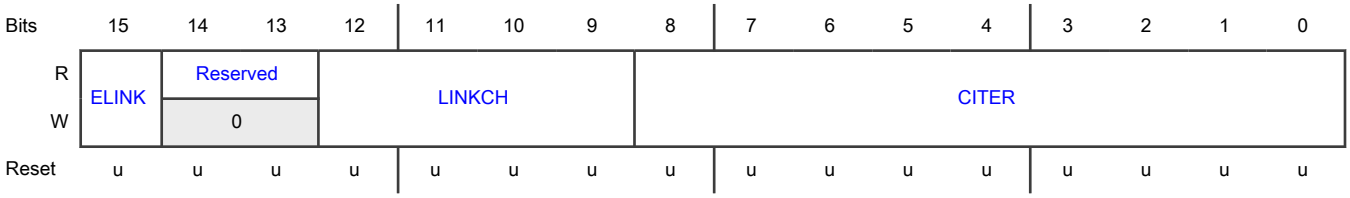
For n = 0 to 15:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-13 —	Reserved
12-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.6.18 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD15_DLAST_SGA)

Offset

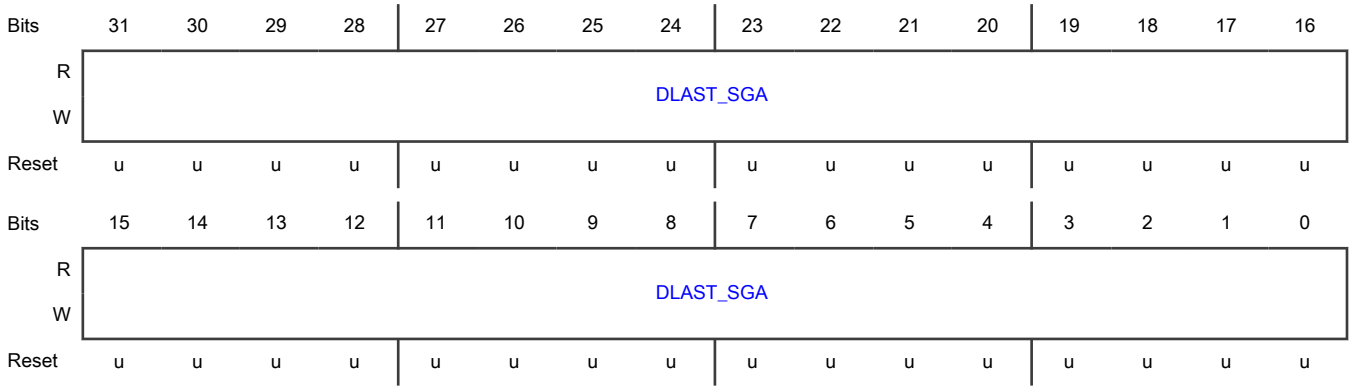
For n = 0 to 15:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

Function

This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Diagram



Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

71.6.6.19 TCD Control and Status (TCD0_CSR - TCD15_CSR)

Offset

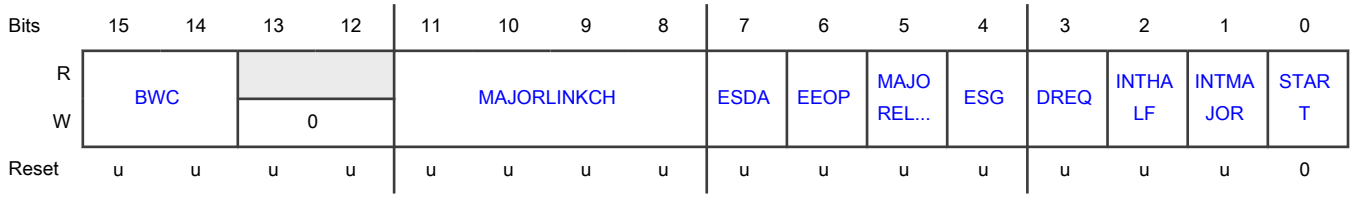
For n = 0 to 15:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

Function

This register is used to enable optional features.

Diagram



Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13-12 —	Reserved
11-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2	Enable Interrupt If Major Counter Half-complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
INTHALF	If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

71.6.6.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD15_BITER_ELINKNO)

Offset

For n = 0 to 15:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Diagram

Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.6.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD15_BITER_ELINKYES)

Offset

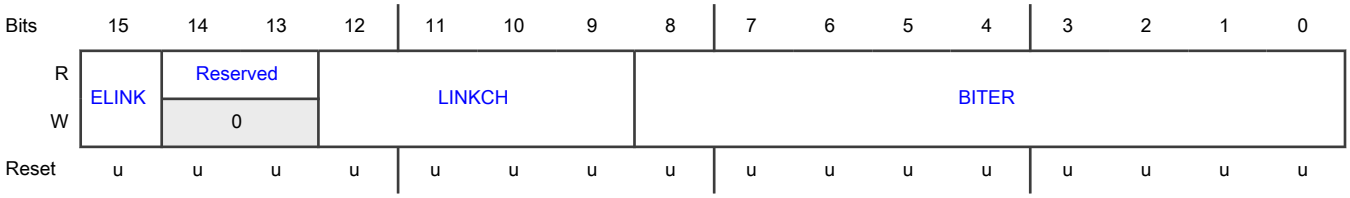
For n = 0 to 15:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-13 —	Reserved
12-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.7 EDMA_TCD register descriptions

71.6.7.1 EDMA_TCD memory map

DMA2_TCD base address: 4030_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 7000h	Channel Control and Status (CH0_CSR - CH7_CSR)	32	RW	0000_0000h
4h - 7004h	Channel Error Status (CH0_ES - CH7_ES)	32	RW	0000_0000h
8h - 7008h	Channel Interrupt Status (CH0_INT - CH7_INT)	32	RW	0000_0000h
Ch - 700Ch	Channel System Bus (CH0_SBR - CH7_SBR)	32	RW	0000_8006h
10h - 7010h	Channel Priority (CH0_PRI - CH7_PRI)	32	RW	0000_0000h
14h - 7014h	Channel Multiplexor Configuration (CH0_MUX - CH7_MUX)	32	RW	0000_0000h
20h - 7020h	TCD Source Address (TCD0_SADDR - TCD7_SADDR)	32	RW	See section
24h - 7024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD7_SOFF)	16	RW	See section
26h - 7026h	TCD Transfer Attributes (TCD0_ATTR - TCD7_ATTR)	16	RW	See section
28h - 7028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD7_NBYTES_MLOFFNO)	32	RW	See section
28h - 7028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD7_NBYTES_MLOFFYES)	32	RW	See section
2Ch - 702Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD7_SLAST_SDA)	32	RW	See section
30h - 7030h	TCD Destination Address (TCD0_DADDR - TCD7_DADDR)	32	RW	See section
34h - 7034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD7_DOFF)	16	RW	See section
36h - 7036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD7_CITER_ELINKNO)	16	RW	See section
36h - 7036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD7_CITER_ELINKYES)	16	RW	See section
38h - 7038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD7_DLAST_SGA)	32	RW	See section
3Ch - 703Ch	TCD Control and Status (TCD0_CSR - TCD7_CSR)	16	RW	See section
3Eh - 703Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD7_BITER_ELINKNO)	16	RW	See section
3Eh - 703Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD7_BITER_ELINKYES)	16	RW	See section

71.6.7.2 Channel Control and Status (CH0_CSR - CH7_CSR)

Offset

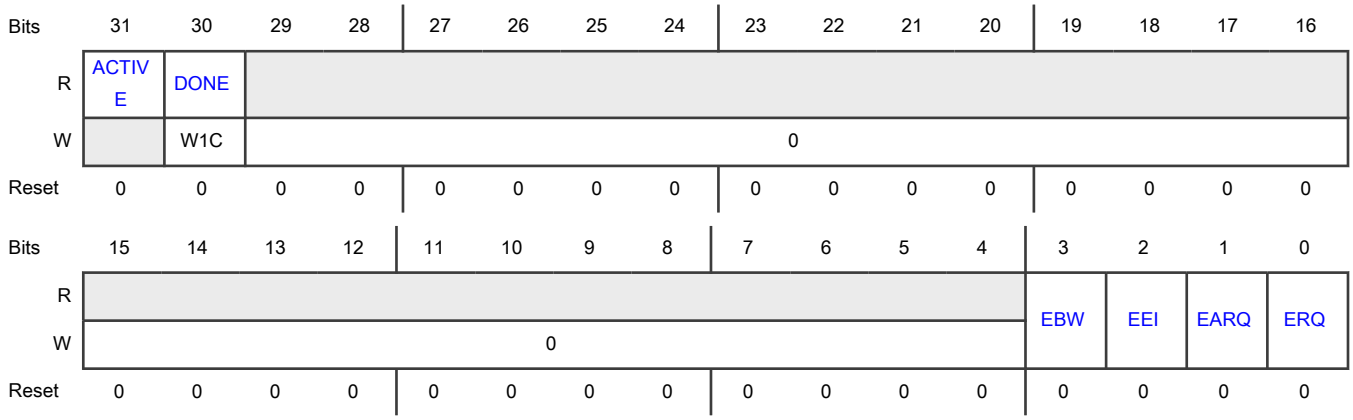
For n = 0 to 7:

Register	Offset
CHn_CSR	0h + (n × 1000h)

Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

Diagram



Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt 1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

71.6.7.3 Channel Error Status (CH0_ES - CH7_ES)

Offset

For n = 0 to 7:

Register	Offset
CHn_ES	4h + (n × 1000h)

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The

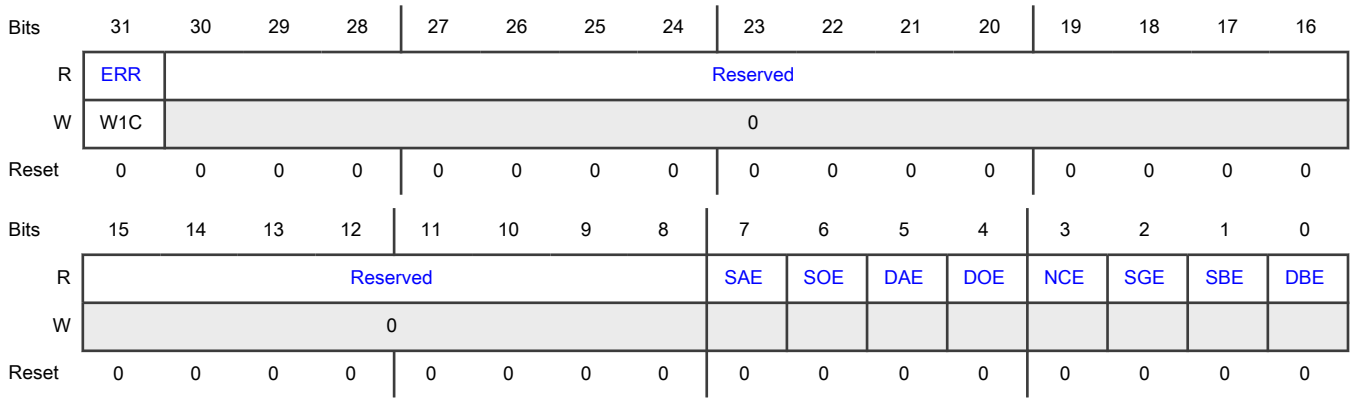
contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

71.6.7.4 Channel Interrupt Status (CH0_INT - CH7_INT)

Offset

For n = 0 to 7:

Register	Offset
CHn_INT	8h + (n × 1000h)

Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

71.6.7.5 Channel System Bus (CH0_SBR - CH7_SBR)

Offset

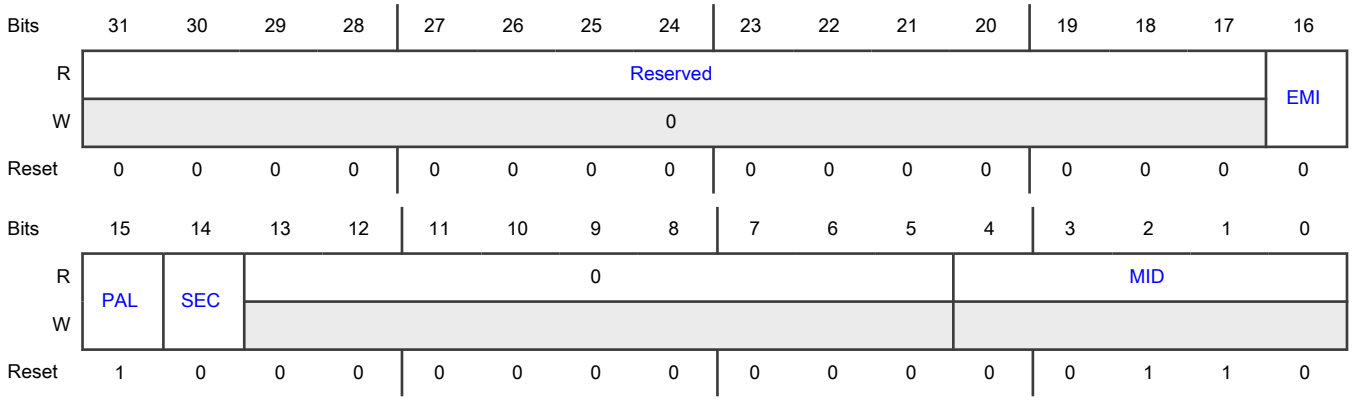
For n = 0 to 7:

Register	Offset
CHn_SBR	Ch + (n × 1000h)

Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication is disabled 1b - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14 SEC	Security Level DMA's security level on the system bus when the channel is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - Nonsecure protection level for DMA transfers 1b - Secure protection level for DMA transfers
13-5 —	Reserved
4-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's security attributes, TCD*n*_SBR[SEC].

71.6.7.6 Channel Priority (CH0_PRI - CH7_PRI)

Offset

For n = 0 to 7:

Register	Offset
CHn_PRI	10h + (n × 1000h)

Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

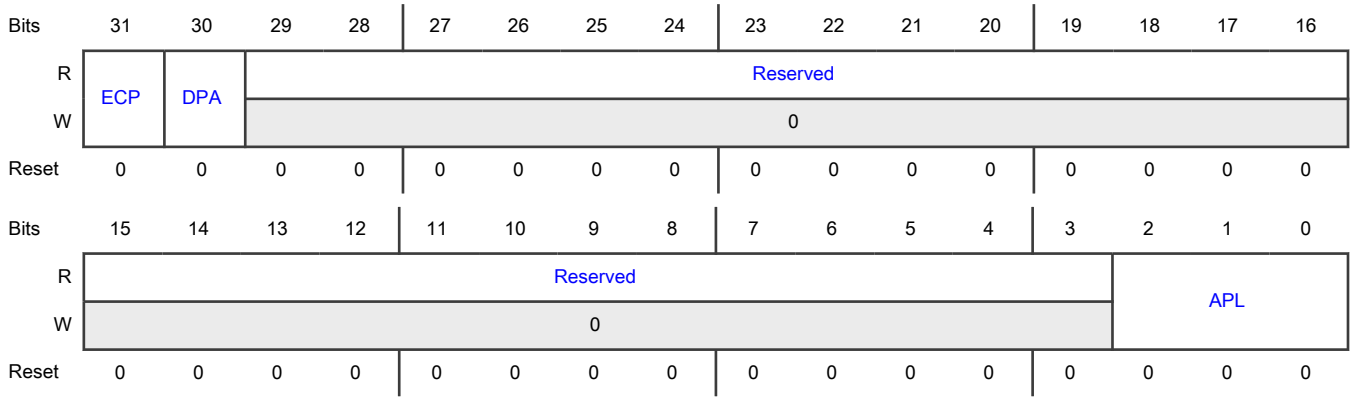
The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Diagram



Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

71.6.7.7 Channel Multiplexor Configuration (CH0_MUX - CH7_MUX)

Offset

For n = 0 to 7:

Register	Offset
CHn_MUX	14h + (n × 1000h)

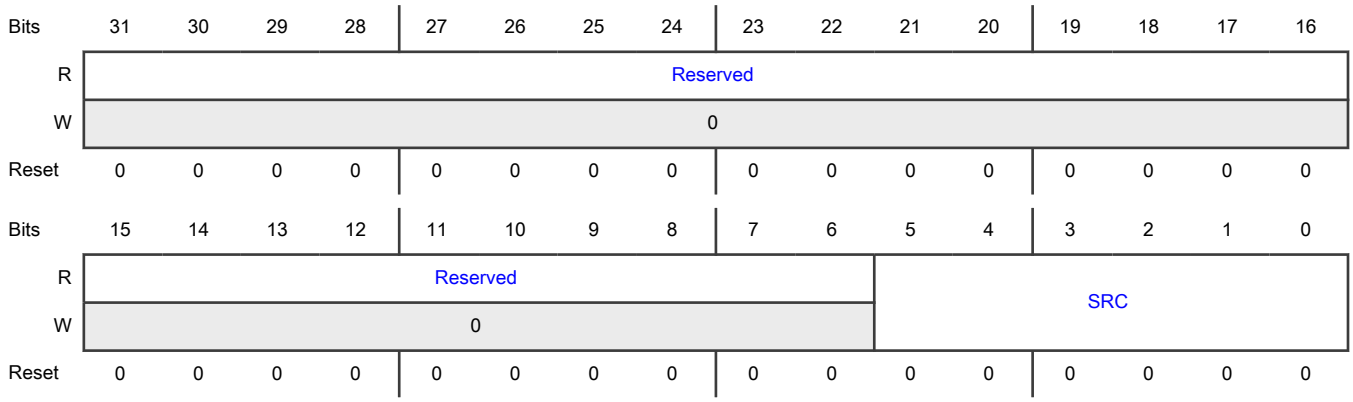
Function

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by any channel, a mux configuration of 0 (SRC = 0) will be written.

All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

71.6.7.8 TCD Source Address (TCD0_SADDR - TCD7_SADDR)

Offset

For n = 0 to 7:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

Function

This register contains the address for the read transactions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

71.6.7.9 TCD Signed Source Address Offset (TCD0_SOFF - TCD7_SOFF)

Offset

For n = 0 to 7:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

Function

This register contains the sign-extended value added to Source Address register after each read transaction.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

71.6.7.10 TCD Transfer Attributes (TCD0_ATTR - TCD7_ATTR)

Offset

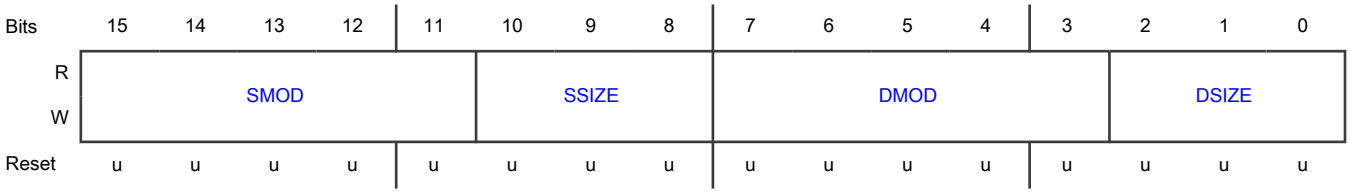
For n = 0 to 7:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

Function

This register contains size and option modulo addressing information for source and destination addresses.

Diagram



Fields

Field	Function
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000b - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - 16-byte 101b - 32-byte 110b - 64-byte 111b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

71.6.7.11 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD7_NBYTES_MLOFFNO)

Offset

For n = 0 to 7:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

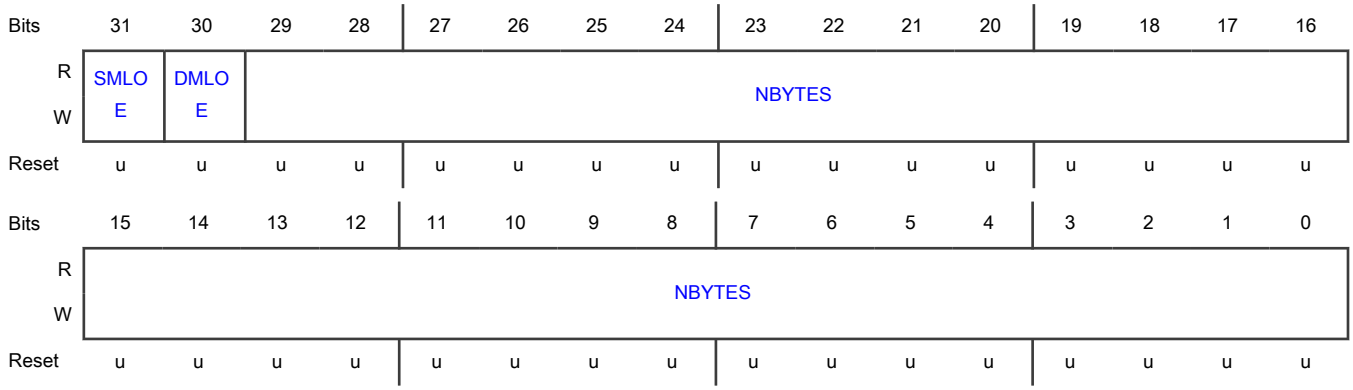
The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.7.12 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD7_NBYTES_MLOFFYES)

Offset

For n = 0 to 7:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

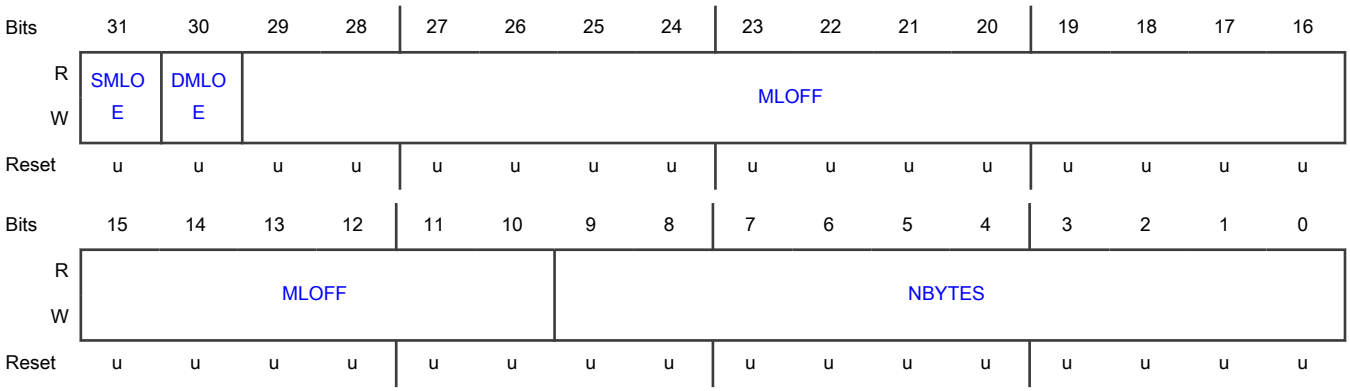
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.7.13 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD7_SLAST_SDA)

Offset

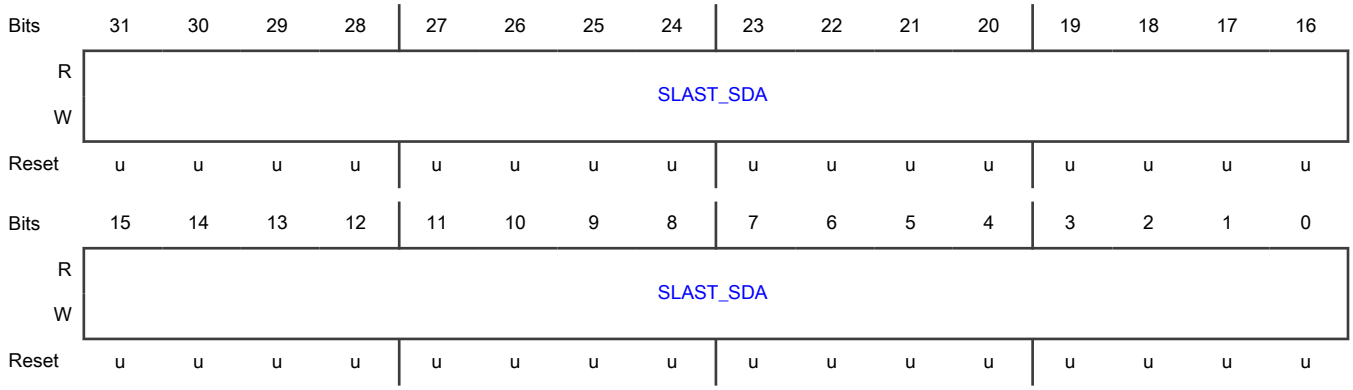
For n = 0 to 7:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

Function

This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Diagram



Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

71.6.7.14 TCD Destination Address (TCD0_DADDR - TCD7_DADDR)

Offset

For n = 0 to 7:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

Function

This register contains the address for the write transactions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

71.6.7.15 TCD Signed Destination Address Offset (TCD0_DOFF - TCD7_DOFF)

Offset

For n = 0 to 7:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

Function

This register contains the sign-extended value added to Destination Address register after each write transaction.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

71.6.7.16 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD7_CITER_ELINKNO)

Offset

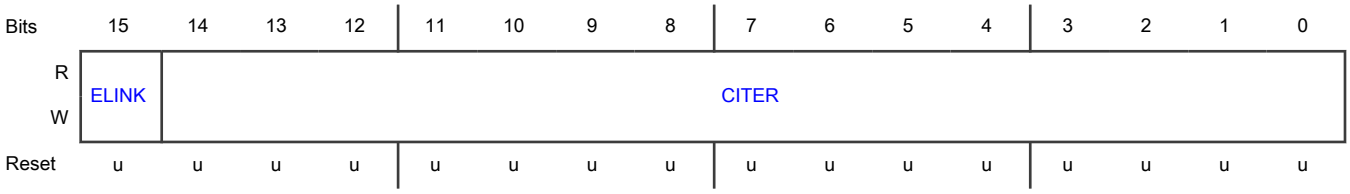
For n = 0 to 7:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.7.17 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD7_CITER_ELINKYES)

Offset

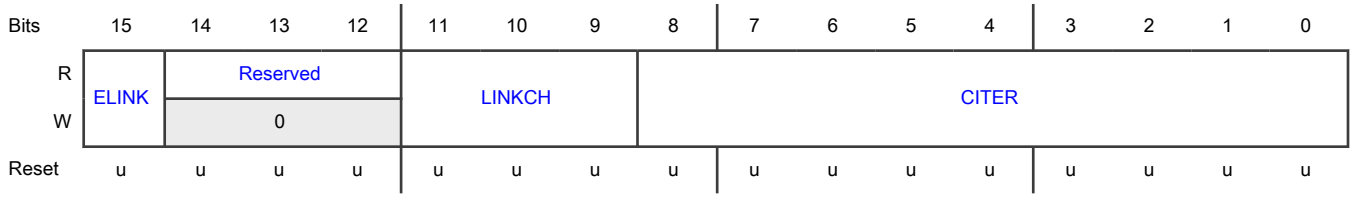
For n = 0 to 7:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-12 —	Reserved
11-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.7.18 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD7_DLAST_SGA)

Offset

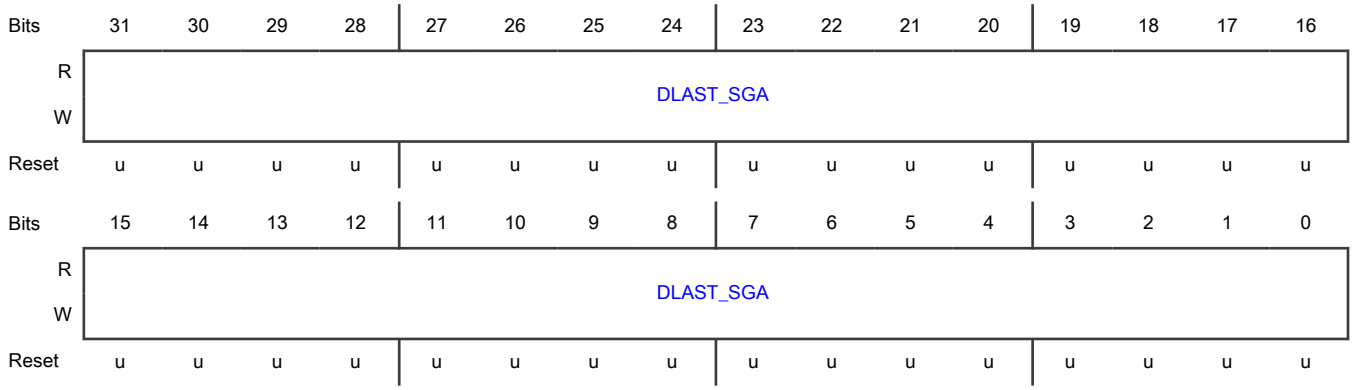
For n = 0 to 7:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

Function

This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Diagram



Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

71.6.7.19 TCD Control and Status (TCD0_CSR - TCD7_CSR)

Offset

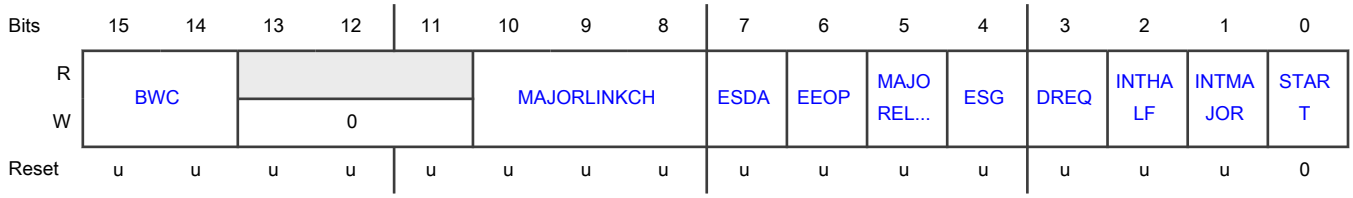
For n = 0 to 7:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

Function

This register is used to enable optional features.

Diagram



Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13-11 —	Reserved
10-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2	Enable Interrupt If Major Counter Half-complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
INTHALF	If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

71.6.7.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD7_BITER_ELINKNO)

Offset

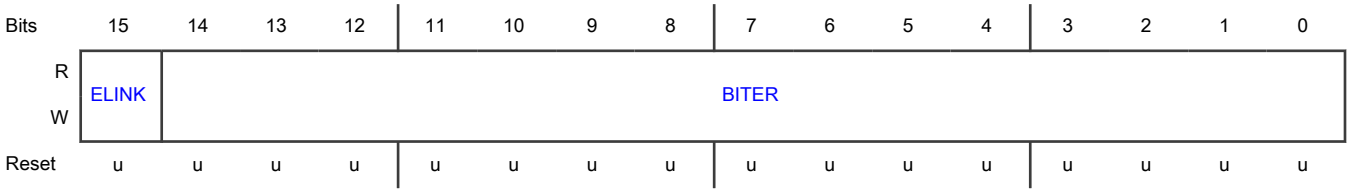
For n = 0 to 7:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.7.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD7_BITER_ELINKYES)

Offset

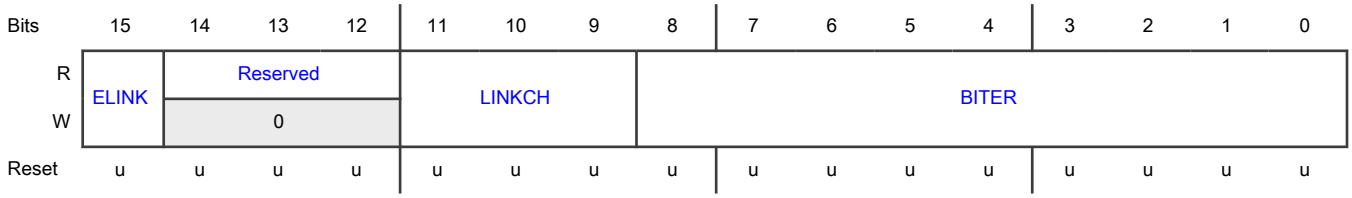
For n = 0 to 7:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-12 —	Reserved
11-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.8 EDMA_TCD register descriptions

71.6.8.1 EDMA_TCD memory map

DMA3_TCD base address: 4031_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 7000h	Channel Control and Status (CH0_CSR - CH7_CSR)	32	RW	0000_0000h
4h - 7004h	Channel Error Status (CH0_ES - CH7_ES)	32	RW	0000_0000h
8h - 7008h	Channel Interrupt Status (CH0_INT - CH7_INT)	32	RW	0000_0000h
Ch - 700Ch	Channel System Bus (CH0_SBR - CH7_SBR)	32	RW	0000_8007h
10h - 7010h	Channel Priority (CH0_PRI - CH7_PRI)	32	RW	0000_0000h
14h - 7014h	Channel Multiplexor Configuration (CH0_MUX - CH7_MUX)	32	RW	0000_0000h
20h - 7020h	TCD Source Address (TCD0_SADDR - TCD7_SADDR)	32	RW	See section
24h - 7024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD7_SOFF)	16	RW	See section
26h - 7026h	TCD Transfer Attributes (TCD0_ATTR - TCD7_ATTR)	16	RW	See section
28h - 7028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD7_NBYTES_MLOFFNO)	32	RW	See section
28h - 7028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD7_NBYTES_MLOFFYES)	32	RW	See section
2Ch - 702Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD7_SLAST_SDA)	32	RW	See section
30h - 7030h	TCD Destination Address (TCD0_DADDR - TCD7_DADDR)	32	RW	See section
34h - 7034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD7_DOFF)	16	RW	See section
36h - 7036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD7_CITER_ELINKNO)	16	RW	See section
36h - 7036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD7_CITER_ELINKYES)	16	RW	See section
38h - 7038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD7_DLAST_SGA)	32	RW	See section
3Ch - 703Ch	TCD Control and Status (TCD0_CSR - TCD7_CSR)	16	RW	See section
3Eh - 703Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD7_BITER_ELINKNO)	16	RW	See section
3Eh - 703Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD7_BITER_ELINKYES)	16	RW	See section

71.6.8.2 Channel Control and Status (CH0_CSR - CH7_CSR)

Offset

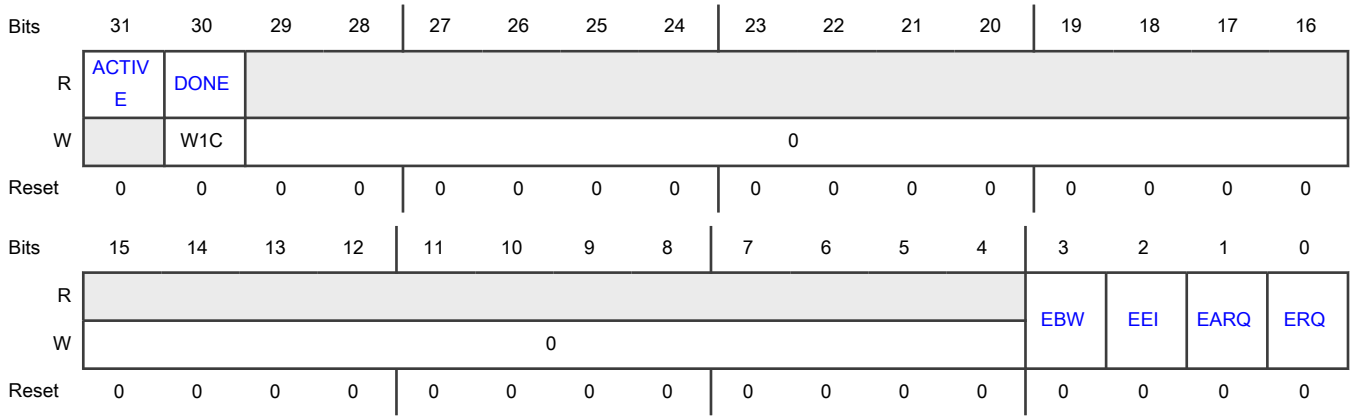
For n = 0 to 7:

Register	Offset
CHn_CSR	0h + (n × 1000h)

Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

Diagram



Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Function
	The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt 1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

71.6.8.3 Channel Error Status (CH0_ES - CH7_ES)

Offset

For n = 0 to 7:

Register	Offset
CHn_ES	4h + (n × 1000h)

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The

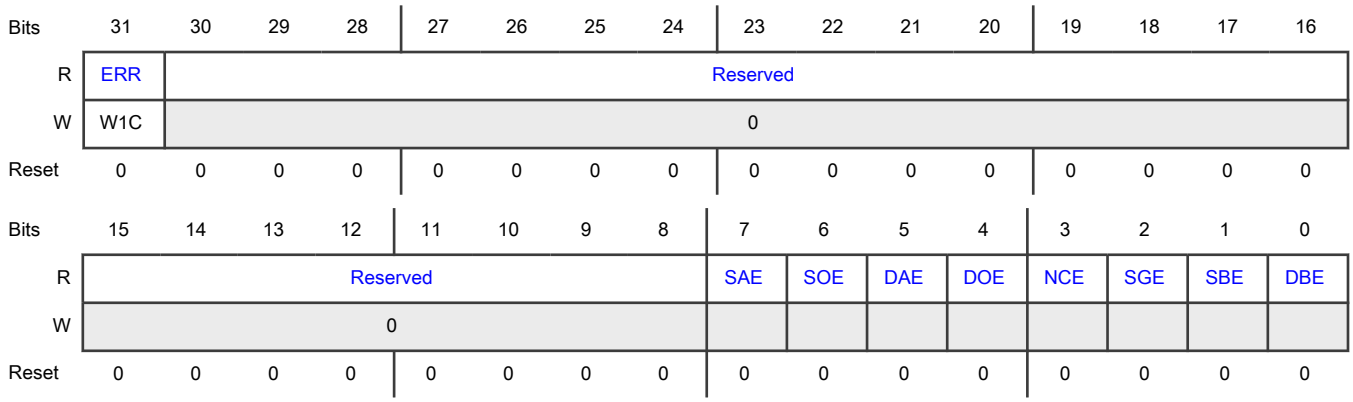
contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

Diagram



Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

71.6.8.4 Channel Interrupt Status (CH0_INT - CH7_INT)

Offset

For n = 0 to 7:

Register	Offset
CHn_INT	8h + (n × 1000h)

Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

71.6.8.5 Channel System Bus (CH0_SBR - CH7_SBR)

Offset

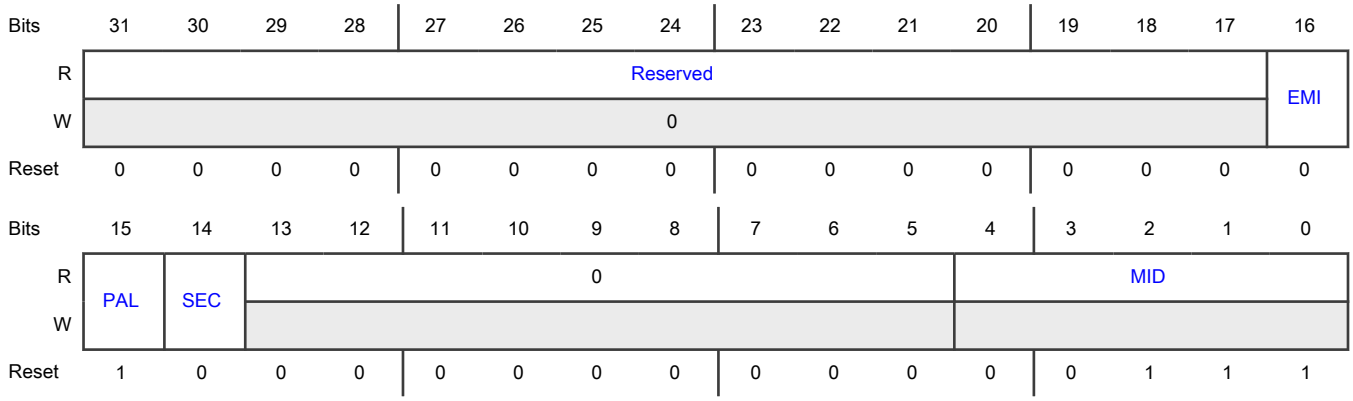
For n = 0 to 7:

Register	Offset
CHn_SBR	Ch + (n × 1000h)

Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0b - Initiator ID replication is disabled 1b - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14 SEC	Security Level DMA's security level on the system bus when the channel is active.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0b - Nonsecure protection level for DMA transfers 1b - Secure protection level for DMA transfers
13-5 —	Reserved
4-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's security attributes, TCD*n*_SBR[SEC].

71.6.8.6 Channel Priority (CH0_PRI - CH7_PRI)

Offset

For n = 0 to 7:

Register	Offset
CHn_PRI	10h + (n × 1000h)

Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH15_GRPRI).

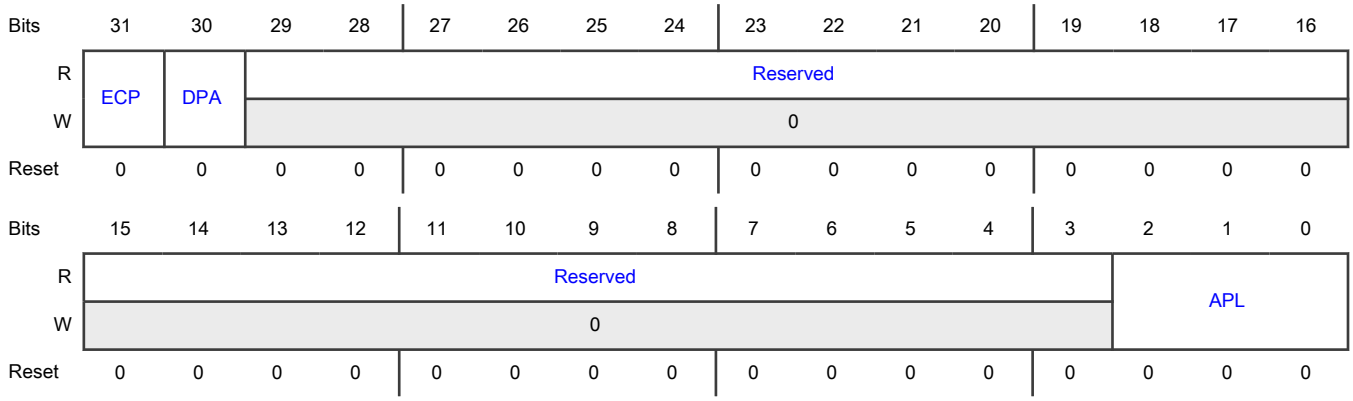
The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Diagram



Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

71.6.8.7 Channel Multiplexor Configuration (CH0_MUX - CH7_MUX)

Offset

For n = 0 to 7:

Register	Offset
CHn_MUX	14h + (n × 1000h)

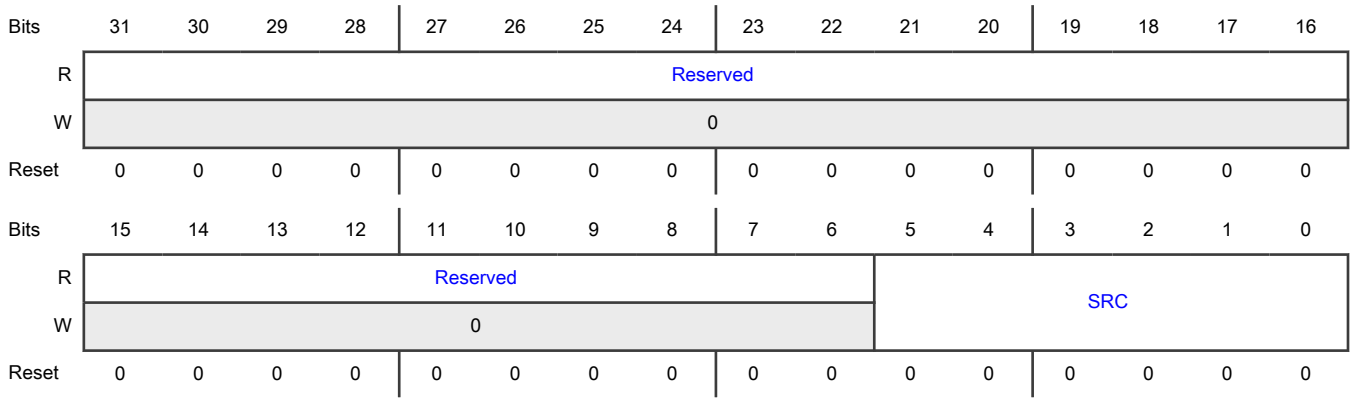
Function

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by any channel, a mux configuration of 0 (SRC = 0) will be written.

All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

71.6.8.8 TCD Source Address (TCD0_SADDR - TCD7_SADDR)

Offset

For n = 0 to 7:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

Function

This register contains the address for the read transactions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

71.6.8.9 TCD Signed Source Address Offset (TCD0_SOFF - TCD7_SOFF)

Offset

For n = 0 to 7:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

Function

This register contains the sign-extended value added to Source Address register after each read transaction.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

71.6.8.10 TCD Transfer Attributes (TCD0_ATTR - TCD7_ATTR)

Offset

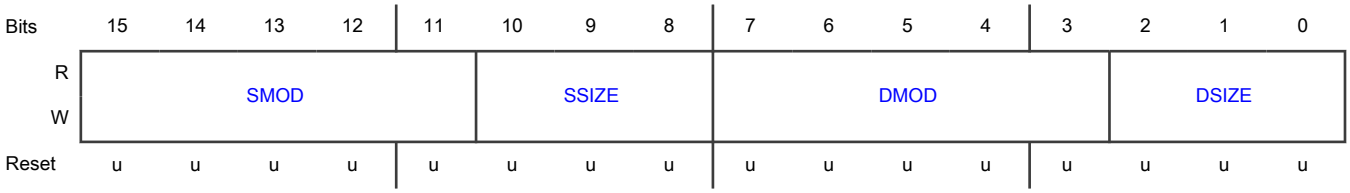
For n = 0 to 7:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

Function

This register contains size and option modulo addressing information for source and destination addresses.

Diagram



Fields

Field	Function
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000b - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - 16-byte 101b - 32-byte 110b - 64-byte 111b - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

71.6.8.11 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD7_NBYTES_MLOFFNO)

Offset

For n = 0 to 7:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

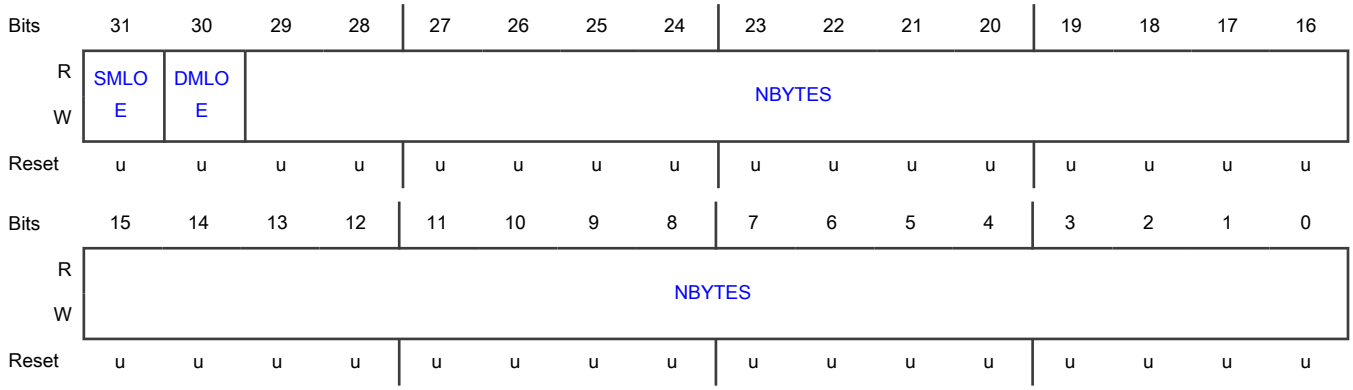
The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.8.12 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD7_NBYTES_MLOFFYES)

Offset

For n = 0 to 7:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

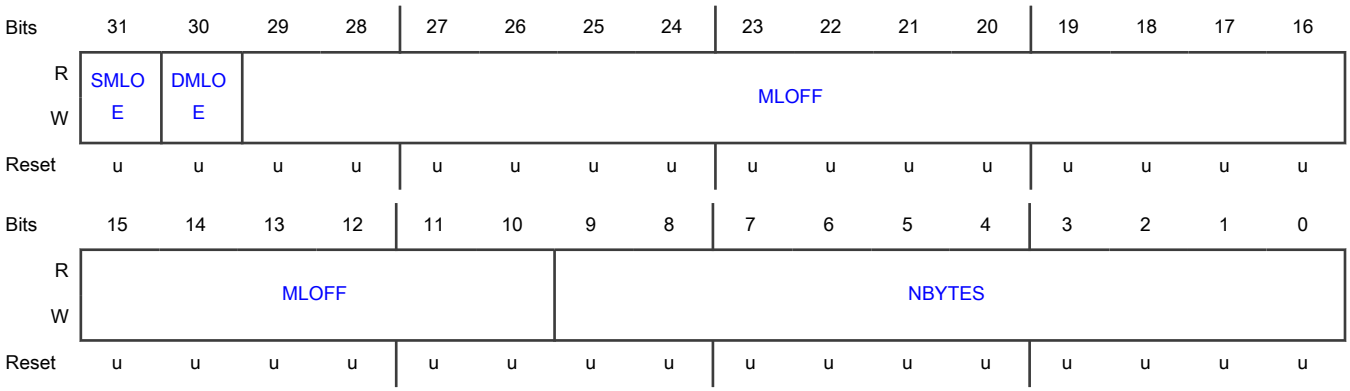
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

71.6.8.13 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD7_SLAST_SDA)

Offset

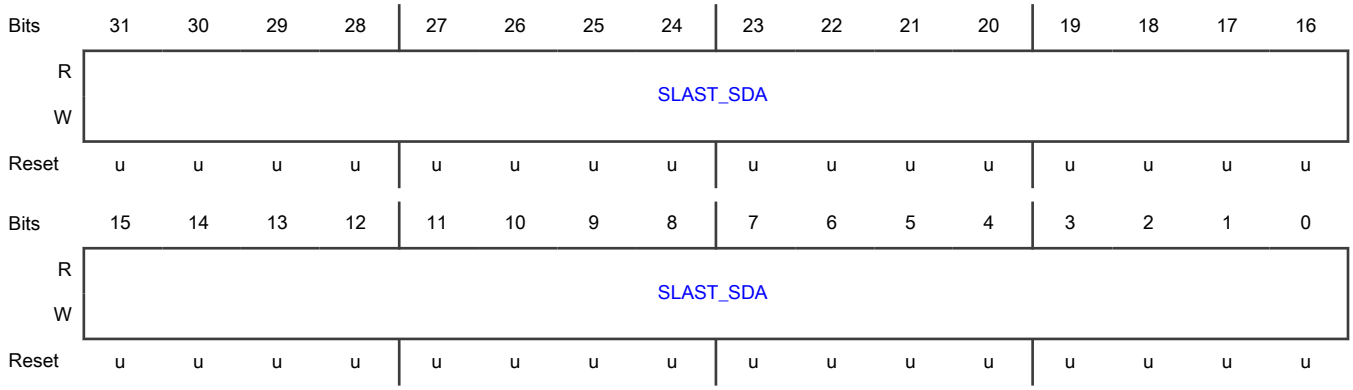
For n = 0 to 7:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

Function

This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Diagram



Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

71.6.8.14 TCD Destination Address (TCD0_DADDR - TCD7_DADDR)

Offset

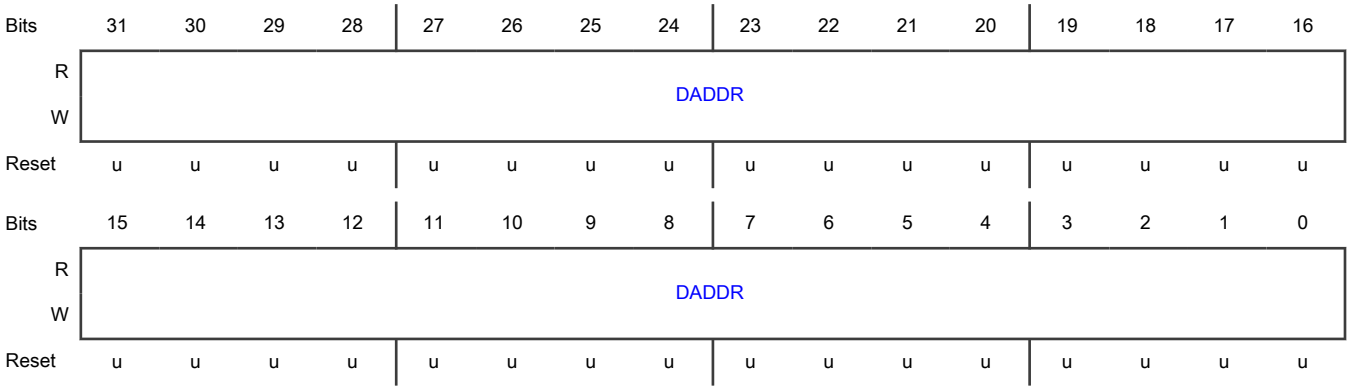
For n = 0 to 7:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

Function

This register contains the address for the write transactions.

Diagram



Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

71.6.8.15 TCD Signed Destination Address Offset (TCD0_DOFF - TCD7_DOFF)

Offset

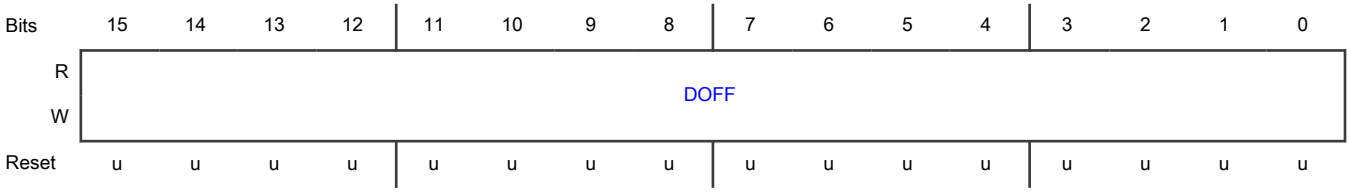
For n = 0 to 7:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

Function

This register contains the sign-extended value added to Destination Address register after each write transaction.

Diagram



Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

71.6.8.16 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD7_CITER_ELINKNO)

Offset

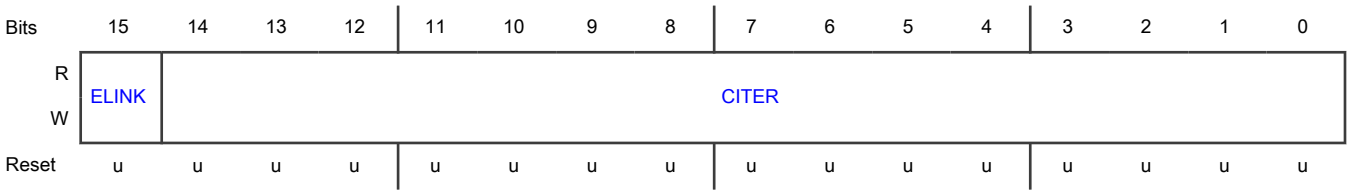
For n = 0 to 7:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.8.17 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD7_CITER_ELINKYES)

Offset

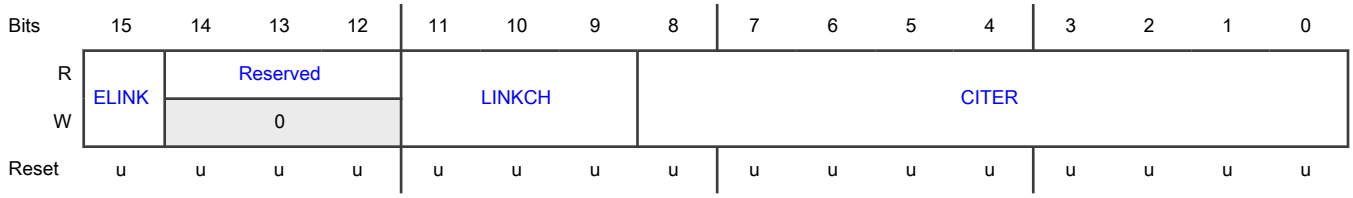
For n = 0 to 7:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	NOTE This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-12 —	Reserved
11-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

71.6.8.18 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD7_DLAST_SGA)

Offset

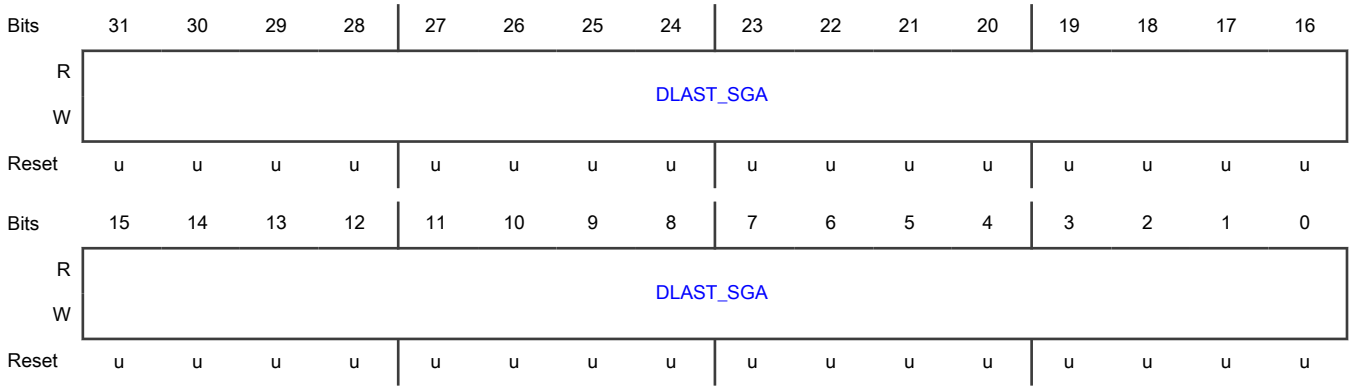
For n = 0 to 7:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

Function

This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Diagram



Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

71.6.8.19 TCD Control and Status (TCD0_CSR - TCD7_CSR)

Offset

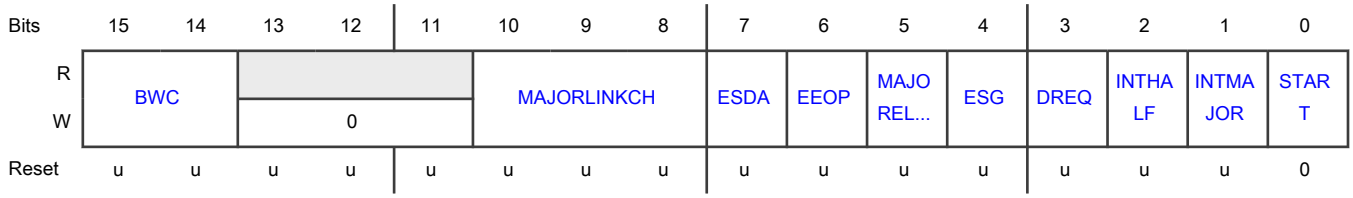
For n = 0 to 7:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

Function

This register is used to enable optional features.

Diagram



Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13-11 —	Reserved
10-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2	Enable Interrupt If Major Counter Half-complete

Table continues on the next page...

Table continued from the previous page...

Field	Function
INTHALF	If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

71.6.8.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD7_BITER_ELINKNO)

Offset

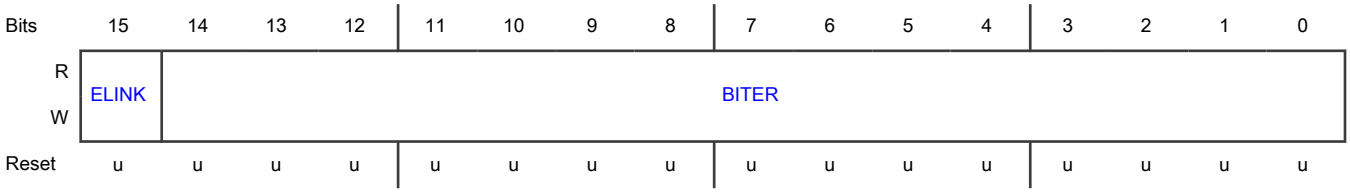
For n = 0 to 7:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

71.6.8.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD7_BITER_ELINKYES)

Offset

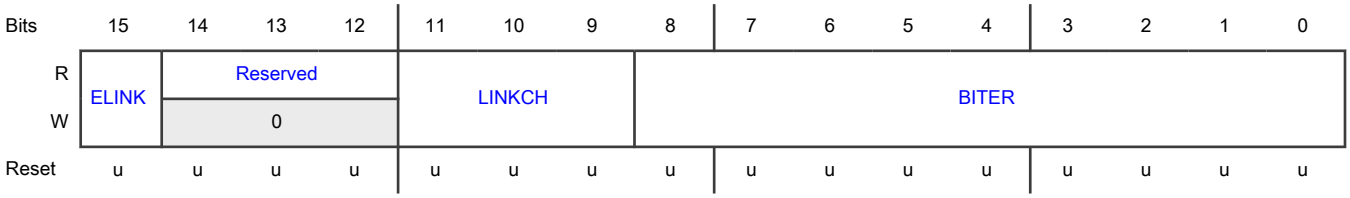
For n = 0 to 7:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-12 —	Reserved
11-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

Chapter 72
Network Interconnect Component (NIC)

72.1 Chip-specific NIC information

Table 700. Reference links to related information

Topic	Related module	Reference
Full description	NIC	NIC
Peripheral memory map		Memory map overview

72.1.1 Module instances

This chip contains three instances of the NIC module: NIC_DSP, NIC_MEDIA0, NIC_MEDIA1. Only NIC_MEDIA1 is configurable and has registers.

72.1.2 NIC diagram

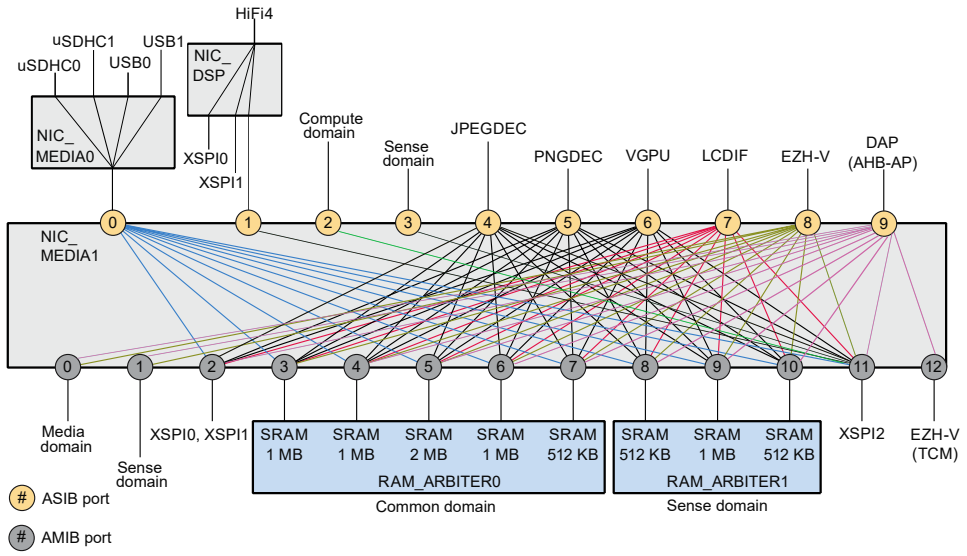


Figure 304. NIC diagram

72.1.3 Interconnection

Below table shows the NIC target interfaces, initiator interfaces, the chip initiator on each target interfaces, and the target memory or peripheral on each initiator interfaces.

+: There is a path between the corresponding target interface and initiator interface so that the SoC initiator on that target interface can access the memory or peripheral on the corresponding initiator interface.

		USB0-1 / uSDHC 0-1 (NIC MEDIA 0)	HiFi4	Compute domain	Sense domain	JPEGDEC	PNGDEC	VGPU	LCDIF	EZH-V	DAP (AHB-AP)
		ASIB0	ASIB1	ASIB2	ASIB3	ASIB4	ASIB5	ASIB6	ASIB7	ASIB8	ASIB9
Media domain	AMIB0									+	+
Sense domain	AMIB1									+	+
XSPI0-1	AMIB2	+				+	+	+	+	+	+
RAM_ARBITER0 Common domain shared SRAM P0-P9 1 MB	AMIB3	+				+	+	+	+	+	+
RAM_ARBITER0 Common domain shared SRAM P10-P11 1 MB	AMIB4	+				+	+	+	+	+	+
RAM_ARBITER0 Common domain shared SRAM P12-P13 2 MB	AMIB5	+				+	+	+	+	+	+
RAM_ARBITER0 Common domain shared SRAM P14-P15 1 MB	AMIB6	+				+	+	+	+	+	+
RAM_ARBITER0 Common domain shared SRAM P16-P17 512 KB	AMIB7	+				+	+	+	+	+	+
RAM_ARBITER1 Sense domain shared SRAM P18-P25 512 KB	AMIB8	+				+	+	+	+	+	+
RAM_ARBITER1 Sense domain shared SRAM P26-P27 1 MB	AMIB9	+				+	+	+	+	+	+
RAM_ARBITER1 Sense domain	AMIB10	+				+	+	+	+	+	+

Table continues on the next page...

Table continued from the previous page...

		USB0-1 / uSDHC 0-1 (NIC MEDIA 0)	HiFi4	Compute domain	Sense domain	JPEGDEC	PNGDEC	VGPU	LCDIF	EZH-V	DAP (AHB-AP)
		ASIB0	ASIB1	ASIB2	ASIB3	ASIB4	ASIB5	ASIB6	ASIB7	ASIB8	ASIB9
shared SRAM P28-P29 512 KB											
XSPI2	AMIB11	+	+	+	+	+	+	+	+	+	+
EZH-V (TCM)	AMIB12										+

72.2 Overview

NIC is an Arm CoreLink network interconnect component that switches any initiator access to the target memory or peripheral. It has initiator interfaces (ASIB0 to ASIB9) and target interfaces (AMIB0 to AMIB12). It supports Arm CoreLink QoS-400 Network Interconnect Advanced Quality of Service (QoS) to arbitrate and regulate the transactions from various SoC initiators.

More details can be found in the following documentation on Arm website:

- ARM CoreLink QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM CoreLink NIC-400 Network Interconnect Technical Reference Manual

  (DSU0026F_corelink_qos400_network_interconnect_r1p0_trm)

72.2.1 Block diagram

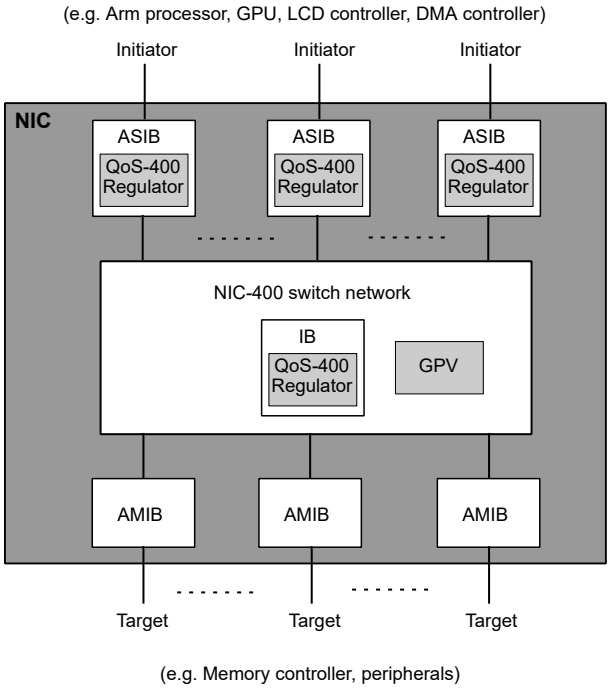


Figure 305. NIC block diagram

72.2.2 Features

- Programmable maximum for outstanding transactions as follows:
  - Separate maximum for read and write requests.
  - Combined maximum for all requests.
  - Fractional value to provide finer control.
- Regulation of read and write request issuing rates to meet programmed traffic specifications as follows:
  - Separate regulation for read and write requests.
  - Combined regulation for all requests.
- Regulation of read and write request QoS values to target a programmed transaction latency.
- Regulation of read and write request QoS values to target a programmed address request latency.

72.3 Functional description

72.3.1 QoS operations

72.3.1.1 Relationship with NIC

NIC includes a QoS value for each address, **awqos** for writes and **arqos** for reads.

Within the interconnect, the QoS values control arbitration. The QoS-400 supplies additional hardware that can both regulate the read and write requests and control the QoS value dynamically.

72.3.1.2 QoS regulators

NIC provides facilities to regulate transactions based on the following inter-related measures:

R_T	Issuing rate
L_T	Latency
N_T	Number of outstanding transactions

Assuming that the initiator is always trying to issue transaction requests, then R_T, L_T, and N_T are related by the following formula:

$$R_T = N_T \times 1/L_T$$

The formula is a variation on Little's Law, that relates queue length to arrival rate and time in the system. A three-dimensional surface plotted in a graph represents the relationship between the transaction issuing rate R_T, latency L_T, and the number of outstanding transactions N_T.

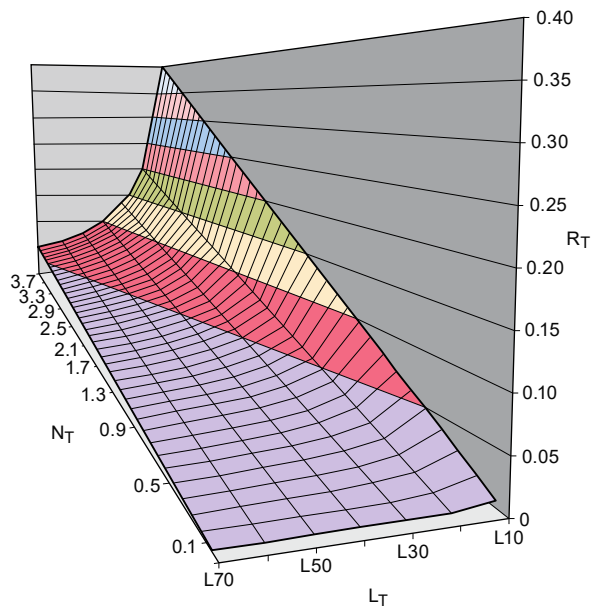


Figure 306. Relationship between R_T, L_T, and N_T

QoS-400 supports three regulation mechanisms based on the R_T, L_T, and N_T, measures. All three regulate the aggregate of the traffic through the ASIB or IB regardless of the eventual destination. This works best when the majority of the traffic is to a single shared resource, such as a DMC.

NOTE

A shared resource, such as a DMC, is often the most heavily loaded resource in a system.

The transaction rate regulators and the number of outstanding transactions set an upper bound on these measures. They prevent the measures from exceeding a limit that you program. They do this by holding back transaction requests whenever the limits are reached.

The transaction latency regulator adds a QoS value to the request, or modifies an existing one, and the interconnect uses that value as an arbitration priority. The latency regulator increases the QoS value when it observes an increase in latency.

Transaction latency specifies a target bandwidth in combination with a number of outstanding transactions using the formula from Little's Law. This might be easier to use where the number of outstanding transactions is limited.

Address request latency regulation directly sets a target period for address requests, that is equivalent to setting a target maximum bandwidth. The number of outstanding transactions can vary according to the transaction latency in accordance with Little's

Law. Consequently, as the transaction latency increases, the number of outstanding transactions also increases to maintain the bandwidth.

NIC QoS regulators provide mechanisms that control how NIC shares the resources of a target. You can use them either individually, or in combination, to control this sharing, and to prevent overloading the target.

One measure of resource loading is the number of requests in the queue waiting to be served. When most of the requests from a number of initiators all go to the same target, the sum of the number of outstanding transactions at each initiator corresponds to the number of requests waiting to be served at that target.

Regulating the number of outstanding transactions from each of the initiators therefore provides a direct means to distribute the resources of a target without overloading it. If the loading changes for any reason, for example when an initiator completes its task early, then:

- The load on the target is reduced.
- The queue length decreases.
- The average latency that the remaining initiators observe decreases.

Figure 306 shows that for a constant number of outstanding transactions, a decrease in latency corresponds to an increase in transaction issuing rate. In effect, some of the spare capacity has been distributed to the other initiators.

Another measure of the resource that a target provides is the rate at which it processes transactions. The transaction rate regulator enables you to limit the rate at which an initiator issues transactions, and therefore sets the proportion of the resource of the target that is requested. Figure 306 shows that for a constant issuing rate, an initiator can compensate for any increase in latency by increasing the number of outstanding transactions.

The third measure of latency is not under the direct control of an initiator because it depends on many factors in the interconnect and the target. The more heavily loaded a target is, the longer a transaction must wait for service, and the higher the latency. You can regulate the latency by adding a QoS value to transaction requests to indicate to the system and target when an initiator requires the latency to be reduced. The higher the value, the lower the latency required.

All three regulators can be active at the same time. For example, if the rate regulator is limiting an initiator to a particular rate, and the latency increases, then the number of outstanding transactions tends to increase to compensate. If you permit this to continue unchecked, then the initiator takes more than its share of the resource from the target. You can program the outstanding transaction regulator to prevent this. At the same time, the latency regulator increases the priority to decrease the latency, enabling the rate to recover, and the number of outstanding transactions to decrease.

ASIBn_QOS_CNTL permits you to enable or disable any combination of the three regulators in one programming step.

72.3.1.3 Transaction rate regulation

A variant of the standard internet Traffic SPECification (TSPEC) specifies transaction rate regulation using the following parameters:

p	Peak rate
b	Burstiness allowance
r	Average rate

You can independently program and enable the regulation of the read and write address channels with their own control bits. Alternatively, you can select combined regulation of the read and write address channels using another control bit.

The request arrival curve, that graph (A) TSPEC traffic upper bound shows, in Figure 307 represents the characteristics imposed on the request flow.

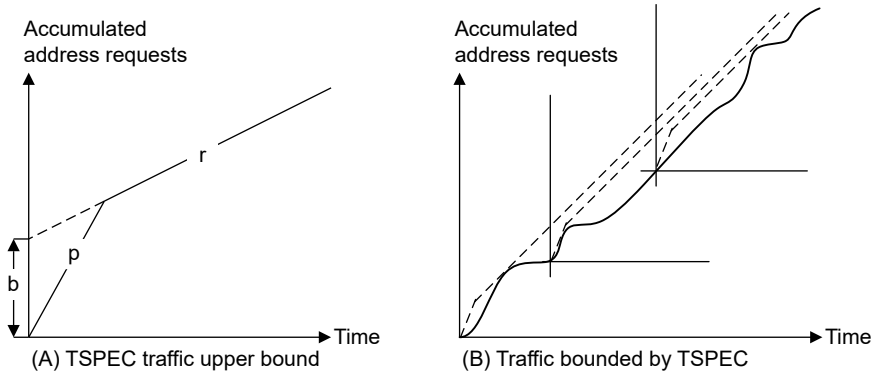


Figure 307. TSPEC traffic specification

The TSPEC parameters define an upper bound that applies over any time window.

(B) Traffic bounded by TSPEC in Figure 307 illustrates this, with the accumulated data curve bounded by the TSPEC curve from any point in the data sequence. You program the TSPEC parameter values for AW and AR request rates in separate sets of registers.

The regulators are disabled after reset. If you program the regulators while they are enabled, the new values take effect immediately. Alternatively, you can disable the regulators, update the values, and then re-enable the regulators.

Binary fractions in transfers per cycle provide the values for peak and average rates. For example:

- A value of 800h, that is, 0.5 in decimal, sets a rate of one transfer every two cycles.
- A value of 100h, that is, 0.0625 in decimal, sets a rate of one transfer every 16 cycles.
- A value of 000h sets a rate of one transfer per cycle, that is, no regulation. If you set either the burstiness to 0, or the average rate to 0, this disables the regulation of burstiness and average rate, (b,r).

In the same way, if you set the peak rate to 0, this disables the peak rate regulation. You can set:

- Peak rate only.
- Burstiness and average rate, without the peak rate.

72.3.1.3.1 Example: How to program the transaction rate regulator registers

This example describes a case where you set the following requirements:

- Transaction rate to use 4% of the available bandwidth.
- Each transaction to be a 16-beat burst.
- Peak rate, p.
- Burstiness allowance, b.
- Average rate registers, r.

It must be noted that the burstiness allowance is not related to the AXI burst length. It is a measure of traffic burstiness, in this case the burstiness of transfers on the AW or AR channels.

Transactions are regulated on AW and AR transfers, so you can use the burst length to calculate the AR or AW transfer rate from the required bandwidth as follows:

- The required data bandwidth is 4% that is equal to four data transfers in every 100 cycles.
- One transaction has 16 beats of data, so requires one transaction every 16/4 * 100 cycles.

  This gives one transaction every 400 cycles.

One transfer every 400 clock cycles is equivalent to 0.0025 transfers per cycle. You must convert this into a 12 bit binary fraction to give the value to program ASIBn_AW_R and ASIBn_AR_R with `0b000000001010`.

NOTE

To calculate the 12-bit binary from the decimal fraction use the following equation:

```
<12-bit binary fraction> = 12^12 * <decimal fraction> = 4096 *
<decimal fraction>
```

For example, converting the decimal fraction 0.0025, would give 4096 * 0.0025 or 10.24. Rounding this to the nearest whole number gives 10 or 0b0000 0000 1010 as a 12-bit binary number.

This value is an approximation limited in accuracy by the number of bits available in the average rate register. See ASIBn_AW_R and ASIBn_AR_R.

Example 1

In this example, the nearest approximation gives a rate of 1 transaction every 409.6 clock cycles or 16/409.6*100 = 3.9% of available bandwidth. If you do not set all three TSPEC values, you can set either the peak rate only, or the burstiness and average rate, without the peak rate.

The peak rate register, p is only 8 bits, not 12 bits as in the average rate register, r, so the previously calculated value does not fit. See ASIBn_AW_R and ASIBn_AR_R. Therefore, you can achieve hard regulation at one transaction in 409 clock cycles by setting the following:

- **p** `0b00000000`. See ASIBn_AW_P and ASIBn_AR_P.
- **r** `0b000000001010`. See ASIBn_AW_R and ASIBn_AR_R.
- **b** `0b0000000000000001`. See ASIBn_AW_B and ASIBn_AR_B.

Example 2

The burstiness allowance, combined with the peak rate and average rate, enables variance in the issuing rate from that initiator during different system loadings. For example, set the values as follows:

- **p** `0b00000001`, that is, 1 in 256. See ASIBn_AW_P and ASIBn_AR_P.
- **r** `0b000000001010`, that is, 1 in 409.6. See ASIBn_AW_R and ASIBn_AR_R.
- **b** `0b0000000000000101`. See ASIBn_AW_B and ASIBn_AR_B.

This permits a maximum issuing rate of one request every 256 clock cycles until the burstiness allowance, b, has been used up, and then, an average issuing rate, r, of one request every 409.6 clock cycles.

The regulator keeps track of the burstiness allowance to control whether the issuing rate is limited to the peak rate, p, or to the average rate, r. The maximum burstiness allowance is configured by ASIBn_AW_B and ASIBn_AR_B. When regulation is enabled it is decremented on every transfer and incremented at the average rate. While the burstiness allowance is non-zero the peak rate is allowed. In this example, the burstiness allowance starts at 5, so, at the peak issuing rate, the allowance reaches 0 after:

b * (p/(p-r)) = 5 * (0.0039/(0.0039 - 0.00244) = 13 transfers

When the allowance has reached zero the issuing rate is limited to the average rate. If the issuing rate from the initiator drops below the average rate, then the burstiness allowance increases until it reaches the programmed burstiness allowance, b.

72.3.1.3.2 Combined AW and AR transaction rate regulation

You can regulate the combined transaction rate from the AW and AR channels. When you select this mode, QoS ignores the AR channel registers and uses the AW channel values as the combined AW + AR transaction rates.

Because two channels support twice the rate of a single channel, QoS scales the parameters for combined regulation by a factor of two. For example, to specify a combined average rate of one transfer every eight cycles, set the value to 100h. This is equal to two transfers every 16 cycles. That is, the rate you program is half the required combined rate.

When the combined AW and AR channel traffic is so close to the transaction rate boundary that only one transfer is permitted, but both channels are requesting, then the regulator admits the AW channel and AR channel alternately.

72.3.1.4 Outstanding transaction regulation

The regulator enables you to program values for the number of AW and AR requests that it issues. You can enable this regulator by configuring ASIBn_QOS_CNTL[EN_AW_OT], ASIBn_QOS_CNTL[EN_AR_OT], and ASIBn_QOS_CNTL[EN_AWAR_OT].

72.3.1.4.1 Fractional outstanding transactions

You can characterize a sequence of transactions with periods when there are no outstanding transactions by using a fractional outstanding transaction number. For example, if requests occur every 100 cycles, but it only takes 50 cycles for the last response to arrive, then this would correspond to an average of 0.5 outstanding transactions. This generalizes to give an average value for any sequence of transactions where the number of outstanding transactions varies.

See ASIBn_MAX_OT registers for configuration details.

72.3.1.4.2 Combined AW and AR outstanding transaction regulation

You can program a maximum outstanding transaction capability for the combined AW and AR channels. This is in addition to the individual channel maximum, so that it is possible to set a combined limit that is lower than the sum of the individual channel limits. A combined value of zero, or greater than or equal to the sum of the individual configuration limits, has no effect.

For example, if the AW and AR configuration issuing capabilities are four and four, the default issuing capability is eight. Setting a combined value of six lowers the combined issuing capability, but leaves the individual channel configuration limits unaltered. Therefore, if there are four outstanding AW requests, then only two AR requests are permitted. Alternatively, if there are only two outstanding AW requests, then four AR requests can be made.

See ASIBn_MAX_COMB_OT registers for configuration details.

72.3.1.5 Transaction and address request latency regulation

72.3.1.5.1 Transaction latency regulation

axqos is either **arqos** or **awqos**. The transaction latency regulator modifies the **axqos** value of each transaction request. This overrides any **axqos** value that NIC might have specified. For example, if a transaction is given an **axqos** value that gives it a higher priority, then this tends to reduce the transaction latency of that transaction. In this way, a feedback loop is set up so that when the actual latency is higher than the target transaction latency, the **axqos** value is proportionately raised, and the larger the transaction latency discrepancy, the higher the priority.

NOTE

The regulator measures the time it takes for a transaction to complete.

Transaction latency regulation is useful for initiators that have performance that is directly dependent on transaction latency. For example, a processor might be stalled while it waits for data after a cache miss.

You program the target transaction latency separately for writes and reads. You enable transaction latency regulation by setting ASIBn_QOS_CNTL[MODE_AR_FC], ASIBn_QOS_CNTL[MODE_AW_FC], ASIBn_QOS_CNTL[EN_AR_FC], and ASIBn_QOS_CNTL[EN_AW_FC].

When you enable transaction latency regulation for reads or writes, the base product **axqos** values are not used. You set the range of **axqos** values used for transaction latency regulation by programming the minimum and maximum values for writes and reads in ASIBn_QOS_RANGE. You program a scaling factor to give control over how quickly the **axqos** values change. The smaller the scaling factor, the more slowly the **axqos** values change in response to changes in transaction latency. The scaling factor is specified in powers of two. See ASIBn_KI_FC.

72.3.1.5.2 Address request latency regulation

The address request latency regulator modifies the **axqos** value of each transaction request. This overrides any **axqos** value that the base product has specified. For example, if a transaction is given an **axqos** value that gives it a higher priority, then this reduces the latency between address requests. In this way, a feedback loop is set up so that when the actual latency is higher than the target latency, the **axqos** value is proportionately raised, and the larger the latency discrepancy, the higher the priority.

NOTE

- The regulator measures the time between address requests.
- **axqos** is either **arqos** or **awqos**.

Address request period regulation is useful for initiators that have minimum bandwidth requirements, for example, a GPU.

You program the target period separately for writes and reads. You enable address request period regulation by setting ASIBn_QOS_CNTL[MODE_AR_FC], ASIBn_QOS_CNTL[MODE_AW_FC], ASIBn_QOS_CNTL[EN_AR_FC], and ASIBn_QOS_CNTL[EN_AW_FC].

When you enable address request period regulation for reads or writes, the **axqos** values are not used. When you disable address request period regulation, the **axqos** values are passed through unmodified.

You set the range of **axqos** values used for address request period regulation by programming the minimum and maximum values for writes and reads in ASIBn_QOS_RANGE.

You program a scaling factor in ASIBn_KI_FC to give control over how quickly the **axqos** values change. The smaller the scaling factor, the more slowly the **axqos** values change in response to changes in latency. The scaling factor is specified in powers of two.

72.3.2 Clocking

NIC operations run off the bus clock that supports the peripherals within the same bus domain. See CLKCTL chapter for more information.

72.3.3 Reset

NIC reset is covered by the reset scheme for the chip and bus domain where it is implemented. See RSTCTL chapter for more information.

72.3.4 Interrupts

This module has no interrupts.

72.4 External signals

This module has no external signals.

72.5 Initialization

This module does not require initialization. Make sure that the associated bus clock and power are on, and follow programming considerations mentioned in QoS operations and Memory map and register definition.

72.6 Memory map and register definition

This section includes NIC memory map and detailed descriptions of all registers.

72.6.1 NIC register descriptions

Each target interface has one set of ASIB registers.

72.6.1.1 NIC memory map

NIC_MEDIA1 base address: 404C_0000h

Offset	Register	Width (In bits)	Access	Reset value
1_2100h - 1_B100h	Read Channel QoS Value (ASIB0_READ_QOS - ASIB9_READ_QOS)	32	RW	0000_0000h
1_2104h - 1_B104h	Write Channel QoS Value (ASIB0_WRITE_QOS - ASIB9_WRITE_QOS)	32	RW	0000_0000h
1_2108h - 1_B108h	Issuing Functionality Modification (ASIB0_FN_MOD - ASIB9_FN_MOD)	32	RW	0000_0000h
1_210Ch - 1_B10Ch	QoS Control (ASIB0_QOS_CNTL - ASIB9_QOS_CNTL)	32	RW	0000_0000h
1_2110h - 1_B110h	Maximum Number of Outstanding Transactions (ASIB0_MAX_OT - ASIB9_MAX_OT)	32	RW	0000_0000h
1_2114h - 1_B114h	Maximum Combined Outstanding Transactions (ASIB0_MAX_COMB_OT - ASIB9_MAX_COMB_OT)	32	RW	0000_0000h
1_2118h - 1_B118h	AW Channel Peak Rate (ASIB0_AW_P - ASIB9_AW_P)	32	RW	0000_0000h
1_211Ch - 1_B11Ch	AW Channel Burstiness Allowance (ASIB0_AW_B - ASIB9_AW_B)	32	RW	0000_0000h
1_2120h - 1_B120h	AW Channel Average Rate (ASIB0_AW_R - ASIB9_AW_R)	32	RW	0000_0000h
1_2124h - 1_B124h	AR Channel Peak Rate (ASIB0_AR_P - ASIB9_AR_P)	32	RW	0000_0000h
1_2128h - 1_B128h	AR Channel Burstiness Allowance (ASIB0_AR_B - ASIB9_AR_B)	32	RW	0000_0000h
1_212Ch - 1_B12Ch	AR Channel Average Rate (ASIB0_AR_R - ASIB9_AR_R)	32	RW	0000_0000h
1_2130h - 1_B130h	Feedback Controlled Target (ASIB0_TARGET_FC - ASIB9_TARGET_FC)	32	RW	0000_0000h
1_2134h - 1_B134h	Feedback Controlled Scale (ASIB0_KI_FC - ASIB9_KI_FC)	32	RW	0000_0000h
1_2138h - 1_B138h	QoS Range (ASIB0_QOS_RANGE - ASIB9_QOS_RANGE)	32	RW	0000_0000h

72.6.1.2 Read Channel QoS Value (ASIB0_READ_QOS - ASIB9_READ_QOS)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_READ_QOS	1_2100h + (n × 1000h)

Function

Sets the priority of the initiator's read.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												READ_QOS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3-0 READ_QOS	Read Channel QoS Value The priority level is used when the initiator's read transaction is being arbitrated by the NIC. 0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority

72.6.1.3 Write Channel QoS Value (ASIB0_WRITE_QOS - ASIB9_WRITE_QOS)

Offset

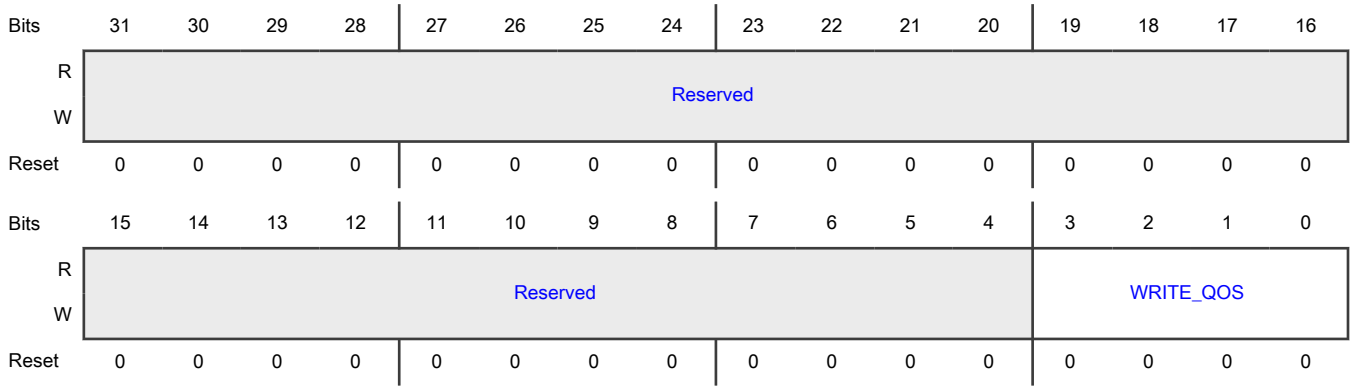
For n = 0 to 9:

Register	Offset
ASIBn_WRITE_QOS	1_2104h + (n × 1000h)

Function

Sets the priority of the initiator's write.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 WRITE_QOS	Write channel QoS value The priority level is used when the initiator's write transaction is being arbitrated by the NIC. 0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority

72.6.1.4 Issuing Functionality Modification (ASIB0_FN_MOD - ASIB9_FN_MOD)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_FN_MOD	1_2108h + (n × 1000h)

Function

Issuing override. This register sets the block issuing capability to one outstanding transaction. The register bits are active high and have the following purpose:

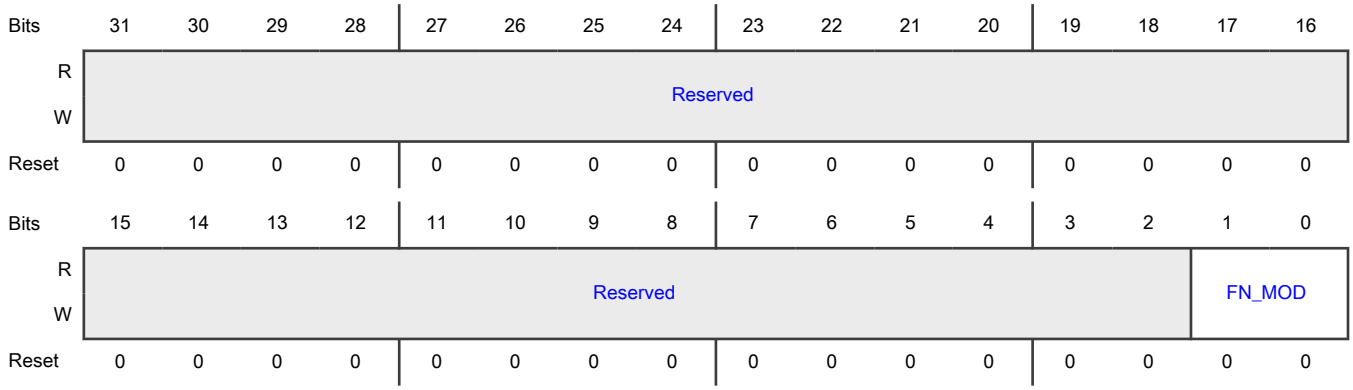
0 - Read issuing, READ_ISS_OVERRIDE.

1 - Write issuing, WRITE_ISS_OVERIDE.

NOTE

It is recommended to keep its reset value to obtain the best performance.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 FN_MOD	Issuing Functionality Modification Sets block issuing capability to one outstanding transaction. 00b - Default issuing 01b - Read issuing 10b - Write issuing 11b - Read and write issuing

72.6.1.5 QoS Control (ASIB0_QOS_CNTL - ASIB9_QOS_CNTL)

Offset

For n = 0 to 9:

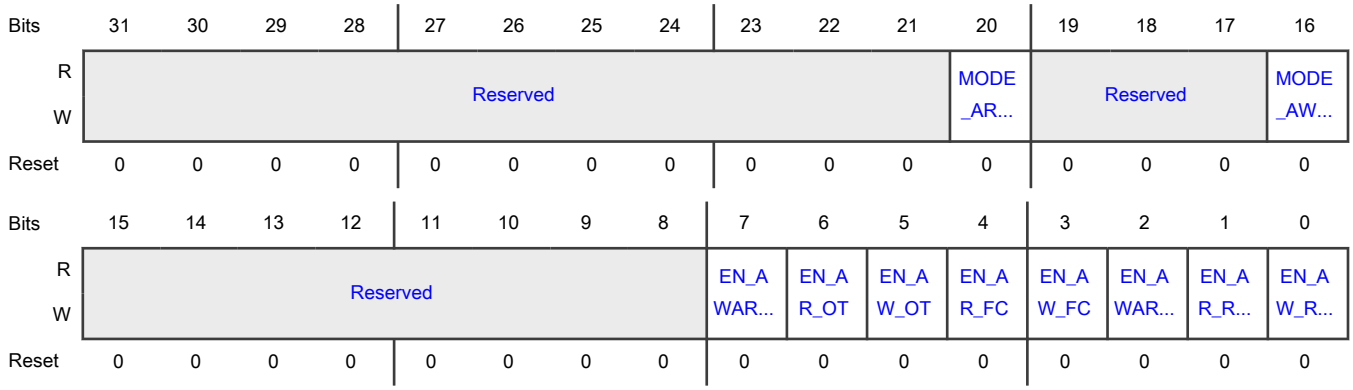
Register	Offset
ASIBn_QOS_CNTL	1_210Ch + (n × 1000h)

Function

Specifies the enable bits for all the regulators. By default, all of the bits are set to 0, and no regulation is enabled.

Regulation only takes place when both the enable bit is set, and its corresponding regulation value is non-zero. The QoS regulators are reset whenever they are re-enabled. The transaction and address latency regulator enables ASIB*n*_QOS_CNTL[EN_AR_FC] and ASIB*n*_QOS_CNTL[EN_AW_FC] can be used to provide a soft reset of these regulators.

Diagram



Fields

Field	Function
31-21 —	Reserved
20 MODE_AR_FC	AR Feedback Control Mode Selects the feedback control regulation for AR of either transaction or address latency. 0b - Transaction latency 1b - Address latency
19-17 —	Reserved
16 MODE_AW_FC	AW Feedback Control Mode Selects the feedback control regulation for AW of either transaction or address latency. 0b - Transaction latency 1b - Address latency
15-8 —	Reserved
7 EN_AWAR_OT	Enable AW/AR Outstanding Transaction Regulation Enables combined regulation of outstanding transactions. 0b - Disables 1b - Enables
6 EN_AR_OT	Enable AR Outstanding Transaction Regulation Enables regulation of outstanding read transactions. 0b - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Function
	1b - Enables
5 EN_AW_OT	Enable AW Outstanding Transaction Regulation Enables regulation of outstanding write transactions. 0b - Disables 1b - Enables
4 EN_AR_FC	Enable AR Feedback Control Regulation Enables regulation of AR transaction or address latency using feedback control, depending on the value to which you set [MODE_AR_FC]. 0b - Disables 1b - Enables
3 EN_AW_FC	Enable AW Feedback Control Regulation Enables regulation of AW transaction or address latency using feedback control, depending on the value to which you set [MODE_AW_FC]. 0b - Disables 1b - Enables
2 EN_AWAR_RATE	Enable Combined AW And AR Transaction Rate Regulation Enables combined regulation of AW and AR channels using transaction rate. 0b - Disables 1b - Enables
1 EN_AR_RATE	Enable AR Transaction Rate Regulation Enables AR channel transaction rate regulation. 0b - Disables 1b - Enables
0 EN_AW_RATE	Enable AW Transaction Rate Regulation Enables AW channel transaction rate regulation. 0b - Disables 1b - Enables

72.6.1.6 Maximum Number of Outstanding Transactions (ASIB0_MAX_OT - ASIB9_MAX_OT)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_MAX_OT	1_2110h + (n × 1000h)

Function

Configures the maximum number of address requests for the AR and AW channels. See Outstanding transaction regulation for more information. Register is ignored if outstanding transaction regulation is not enabled (ASIBn_QOS_CNTL[EN_AWAR_OT], [EN_AR_OT], and [EN_AW_OT] all equal 0).

NOTE

The outstanding transaction limits have an integer part and a fractional part. The AW and AR outstanding transaction limits are enabled when you set the corresponding ASIB*n*_QOS_CNTL[EN_AW_OT] or [EN_AR_OT] fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		AR_MAX_OTI						AR_MAX_OTF							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		AW_MAX_OTI						AW_MAX_OTF							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29-24 AR_MAX_OTI	Integer Part of Maximum Outstanding AR Addresses Corresponds to the integer part of the maximum outstanding AR addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply.
23-16 AR_MAX_OTF	Fractional Part of Maximum Outstanding AR Addresses Corresponds to the fractional part of the maximum outstanding AR addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply. See the note about binary fraction representation in Example: How to program the transaction rate regulator registers for more information.
15-14 —	Reserved
13-8	Integer Part of Maximum Outstanding AW Addresses

Table continues on the next page...

Table continued from the previous page...

Field	Function
AW_MAX_OTI	Corresponds to the integer part of the maximum outstanding AW addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply.
7-0 AW_MAX_OTF	Fractional Part of Maximum Outstanding AW Addresses Corresponds to the fractional part of the maximum outstanding AW addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply. See the note about binary fraction representation in Example: How to program the transaction rate regulator registers for more information.

72.6.1.7 Maximum Combined Outstanding Transactions (ASIB0_MAX_COMB_OT - ASIB9_MAX_COMB_OT)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_MAX_COMB_OT	1_2114h + (n × 1000h)

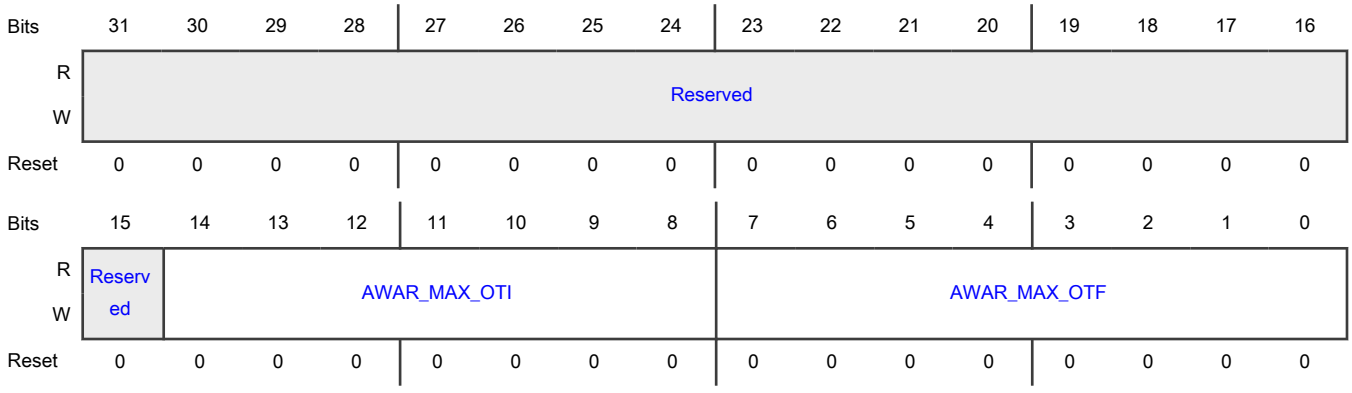
Function

Configures the maximum number of address requests for the AR and AW channels. The combined limit is applied after any individual channel limits. See Outstanding transaction regulation for more information. Register is ignored if outstanding transaction regulation is not enabled (ASIBn_QOS_CNTL[EN_AWAR_OT], [EN_AR_OT], and [EN_AW_OT] all equal 0).

NOTE

The outstanding transaction limits have an integer part and a fractional part. The regulation of the combined outstanding transaction limit also requires that you set ASIB*n*_QOS_CNTL[EN_AWAR_OT] field.

Diagram



Fields

Field	Function
31-15 —	Reserved
14-8 AWAR_MAX_OTI	Integer Part of Maximum Combined Outstanding AW and AR Addresses Corresponds to the integer part of the maximum combined outstanding AW and AR addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply.
7-0 AWAR_MAX_OTF	Fractional Part of Maximum Combined Outstanding AW and AR Addresses Corresponds to the binary fraction of the maximum combined outstanding AW and AR addresses. A value of 0 for both the integer and fractional parts disables the programmable regulation so that the configuration limits apply. See the note about binary fraction representation for more information.

72.6.1.8 AW Channel Peak Rate (ASIB0_AW_P - ASIB9_AW_P)

Offset

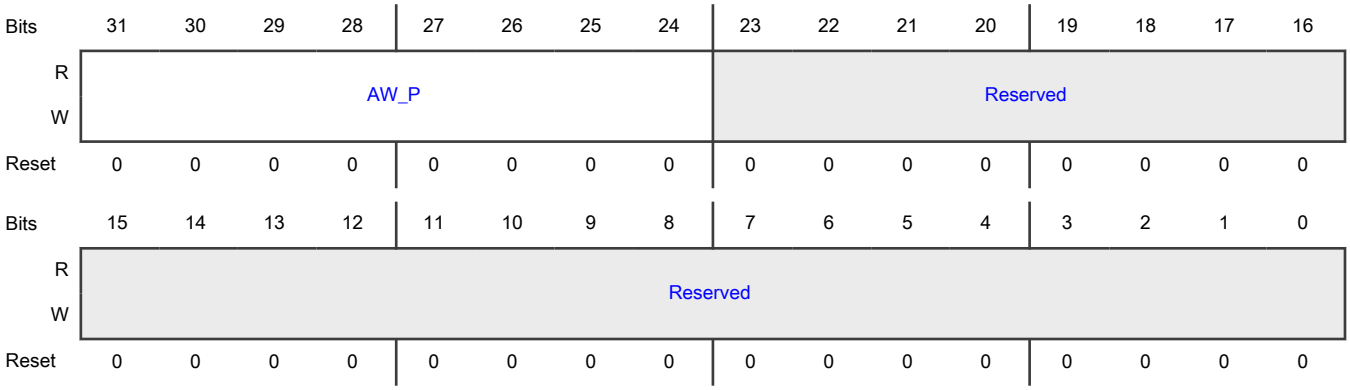
For n = 0 to 9:

Register	Offset
ASIBn_AW_P	1_2118h + (n × 1000h)

Function

Configures the peak number of transfers per cycle. See Transaction rate regulation for more information.

Diagram



Fields

Field	Function
31-24	AW Channel Peak Rate

Table continues on the next page...

Table continued from the previous page...

Field	Function
AW_P	Binary fraction of the peak number of transfers allowed per cycle by the transaction rate regulator. See Transaction rate regulation for more information.
23-0 —	Reserved

72.6.1.9 AW Channel Burstiness Allowance (ASIB0_AW_B - ASIB9_AW_B)

Offset

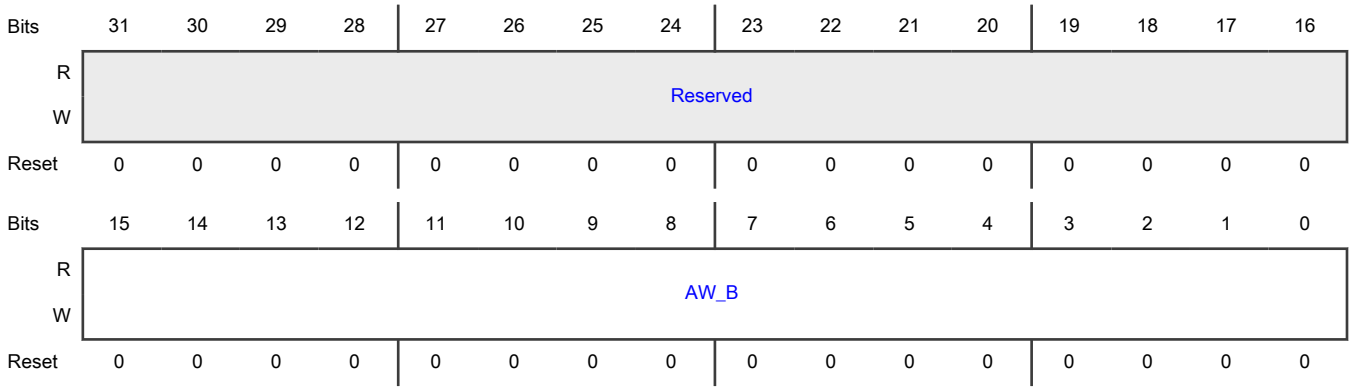
For n = 0 to 9:

Register	Offset
ASIBn_AW_B	1_211Ch + (n × 1000h)

Function

Configures burstiness allowance in number of transfers. See Transaction rate regulation for more information.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 AW_B	AW Channel Burstiness Configures the starting and maximum value for write burstiness used by the transaction rate regulator. See Transaction rate regulation for more information.

72.6.1.10 AW Channel Average Rate (ASIB0_AW_R - ASIB9_AW_R)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_AW_R	1_2120h + (n × 1000h)

Function

Configures the average number of transfers per cycle. See Transaction rate regulation for more information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AW_R												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 AW_R	AW Channel Average Rate Binary fraction of the average number of transfers per cycle used by the transaction rate regulator. See Transaction rate regulation for more information.
19-0 —	Reserved

72.6.1.11 AR Channel Peak Rate (ASIB0_AR_P - ASIB9_AR_P)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_AR_P	1_2124h + (n × 1000h)

Function

Configures the peak number of transfers per cycle. See Transaction rate regulation for more information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AR_P								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 AR_P	AR Channel Peak Rate Binary fraction of the peak number of transfers allowed per cycle by the transaction rate regulator. See Transaction rate regulation for more information.
23-0 —	Reserved

72.6.1.12 AR Channel Burstiness Allowance (ASIB0_AR_B - ASIB9_AR_B)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_AR_B	1_2128h + (n × 1000h)

Function

Configures burstiness allowance in number of transfers. See Transaction rate regulation for more information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AR_B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 AR_B	AR Channel Burstiness Configures the starting and max value for write burstiness used by the transaction rate regulator. See Transaction rate regulation for more information.

72.6.1.13 AR Channel Average Rate (ASIB0_AR_R - ASIB9_AR_R)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_AR_R	1_212Ch + (n × 1000h)

Function

Configures the average number of transfers per cycle. See Transaction rate regulation for more information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AR_R												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 AR_R	AR Channel Average Rate Binary fraction of the average number of transfers per cycle used by the transaction rate regulator. See Transaction rate regulation for more information.
19-0 —	Reserved

72.6.1.14 Feedback Controlled Target (ASIB0_TARGET_FC - ASIB9_TARGET_FC)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_TARGET_FC	1_2130h + (n × 1000h)

Function

Configures target latency in cycles for the regulation of reads and writes. A value of 0 corresponds to no regulation. See Transaction latency regulation and Address request latency regulation for more information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				AR_TGT_LATENCY											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				AW_TGT_LATENCY											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-16 AR_TGT_LATENCY	AR Channel Target Latency Target latency measured in number of bus clocks.
15-12 —	Reserved
11-0 AW_TGT_LATENCY	AW Channel Target Latency Target latency measured in number of bus clocks.

72.6.1.15 Feedback Controlled Scale (ASIB0_KI_FC - ASIB9_KI_FC)

Offset

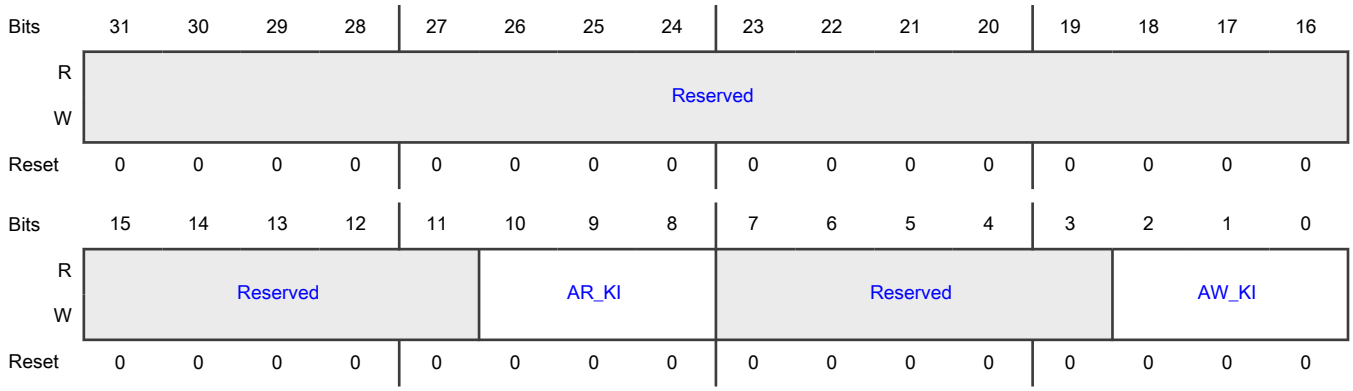
For n = 0 to 9:

Register	Offset
ASIBn_KI_FC	1_2134h + (n × 1000h)

Function

Configures the QoS scaling factor when transaction latency regulation or address request latency regulation is used. Scale factor is coded for powers of 2 in the range 2^{-3} to 2^{-10}, to match a 16-bit integrator. See Transaction latency regulation and Address request latency regulation for more information.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-8 AR_KI	arqos Scale Factor Specifies the latency regulation scale factor, power of 2 in the range 2^{-3} to 2^{-10}. arqos is incremented or decremented by this scale factor based on the initiators latency. 000b - 2^-3 001b - 2^-4 010b-110b - 2^-5 to 2^-9 111b - 2^-10
7-3 —	Reserved
2-0 AW_KI	awqos Scale Factor Specifies the latency regulation scale factor, power of 2 in the range 2^{-3} to 2^{-10}. 000b - 2^-3 001b - 2^-4

Table continues on the next page...

Table continued from the previous page...

Field	Function
	010b-110b - 2^-5 to 2^-9 111b - 2^-10

72.6.1.16 QoS Range (ASIB0_QOS_RANGE - ASIB9_QOS_RANGE)

Offset

For n = 0 to 9:

Register	Offset
ASIBn_QOS_RANGE	1_2138h + (n × 1000h)

Function

Configures the minimum and maximum values for the arqos and awqos signals that the latency regulators generate. See Transaction latency regulation and Address request latency regulation for more information. If the values in this register are changed to make the range narrower than either the previous upper or lower bounds then, you must reset the regulator through a soft reset by disabling and re-enabling it. If the values in this register are changed to make the range wider than either the previous upper or lower bounds then, the regulator does not require a soft reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				AR_MAX_QOS				Reserved				AR_MIN_QOS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				AW_MAX_QOS				Reserved				AW_MIN_QOS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 AR_MAX_QOS	Maximum arqos Maximum priority level used when the initiator's read transaction is being arbitrated by the NIC.

Table continues on the next page...

Table continued from the previous page...

Field	Function
	0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority
23-20 —	Reserved
19-16 AR_MIN_QOS	Minimum arqos Minimum priority level used when the initiator's read transaction is being arbitrated by the NIC. **WARNING** Setting this to a value greater than AR_MAX_QOS will cause unpredictable behavior. 0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority
15-12 —	Reserved
11-8 AW_MAX_QOS	Maximum awqos Maximum priority level used when the initiator's write transaction is being arbitrated by the NIC. 0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority
7-4 —	Reserved
3-0 AW_MIN_QOS	Minimum awqos Minimum priority level used when the initiator's write transaction is being arbitrated by the NIC. **WARNING** Setting this to a value greater than AW_MAX_QOS will cause unpredictable behavior. 0000b - The lowest priority 0001b-1110b - ... 1111b - The highest priority

72.7 Glossary

AMIB	AXI Manager Interface Block
ASIB	AXI Subordinate Interface Block

IB	Interface Block
DMC	Dynamic Memory Controller
AR	AXI Read
AW	AXI Write

Chapter 73
On-Chip OTP Controller (OCOTP_CTRL)

73.1 Chip-specific OCOTP_CTRL information

Table 701. Reference links to related information

Topic	Related module	Reference
Full description	OCOTP_CTRL	OCOTP_CTRL
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller
Fusemap		Fusemap

73.1.1 Module instances

This device has one instance of the OCOTP_CTRL module.

Table 702. OCOTP_CTRL instance

Features	OCOTP_CTRL
Located in	VDD2_COM
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

73.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software. The secure AHB controller controls the security level for access to peripherals and does default to secure and privileged access for all peripherals.

73.2 Overview

This section contains information describing the On-Chip OTP controller (OCOTP_CTRL) along with details about the block functionality and implementation. In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

73.2.1 Block Diagram

The system level block diagram for OCOTP is provided below.

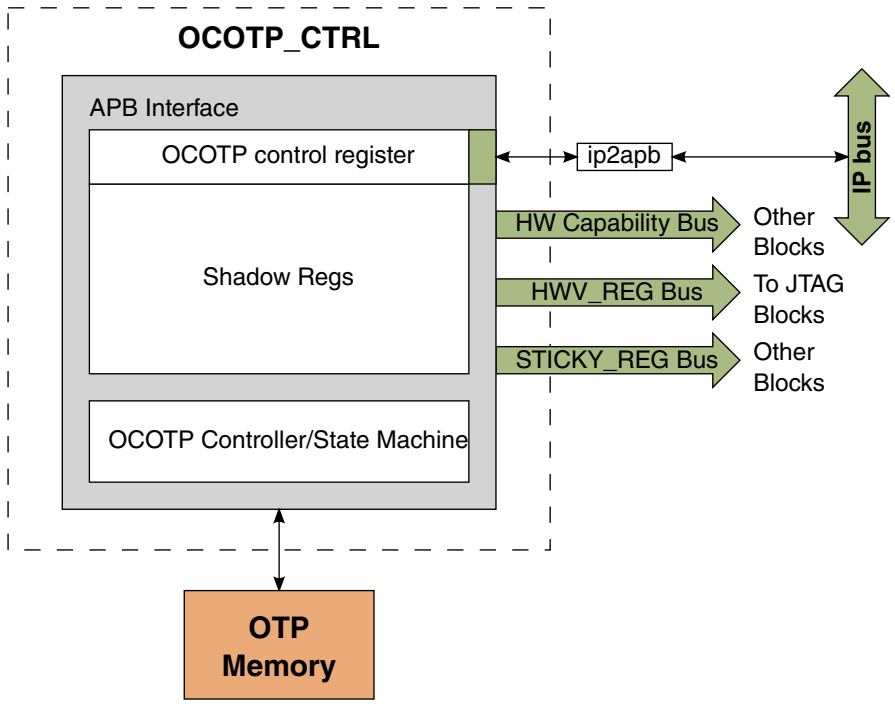


Figure 308. OCOTP System Level Block Diagram

73.2.2 Features

The OCOTP provides the following features :

- 32-bit word restricted program and read to
- Loading and housing of eFuse content into shadow registers
- Memory-mapped (restricted) access to shadow registers
- Generation of STICKY_REG which consists of sticky register bits
- Provides program-protect and read-protect of eFuse
- Provides read protection of shadow register
- Supports Error-Correcting Code (ECC) mode programming and reading for all of the user fuse words
- Supports redundancy mode programming and reading for all of the supplementary fuse words

73.3 Functional Description

73.3.1 Operations

The IP bus interface of the OCOTP has the following functions.

- Configure control registers for programming and reading all the fuse words
- Read shadow registers

OCOTP configuration for programs and reads is allocated on 32-bit word boundaries for software (SW) convenience. For writes, the 32-bit word reflects the "write-mask". OTP Memory bits not yet programmed are 0. Programmed bits are 1.

NOTE

For fuse words that use ECC, the fuse word can only be programmed one time, otherwise it will report single-bit error correction (SEC) / double-bit error detection (DED) issues in read or reload. Refer to the fusemap chapter to identifiy which fuse words use ECC for error correction.

NOTE

For fuse words that use redundant bits, the fuse word can be programmed more than one time (not the same fuse bit). Refer to the fusemap chapter to identifiy which fuse words use redundant bits for error correction.

73.3.1.1 Shadow Register Reload

All fuse words in efuse box are shadowed. This means that at reset, fuse values are read and then copied into memory-mapped shadow registers. If fuses are subsequently programmed, the shadow registers should be reloaded to keep them coherent with the fuse bank arrays.

The "reload shadows" feature allows the user to force a reload of all shadow registers without having to reset the device. To force a reload, complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a new access can be requested.
2. Set the CTRL[RELOAD_SHADOWS] bit. OCOTP will read all the fuses one by one and put it into the corresponding shadow register.
3. Wait for CTRL[BUSY] and CTRL[RELOAD_SHADOWS] to be cleared by the controller.
4. Check the values of OUT_STATUS[SEC_RELOAD] and OUT_STATUS[DED_RELOAD].
   - If OUT_STATUS[SEC_RELOAD] is set, it indicates at least one single bit correction happened when reloading.
   - If OUT_STATUS[DED_RELOAD] is set, it indicates at least one double-bit error happened when reloading. That is, one or more fuse words are not correct when reading from eFuse.

The controller will automatically clear the CTRL[RELOAD_SHADOWS] bit after the successful completion of the operation.

73.3.1.2 Fuse and Shadow Register Read

All shadow registers are always readable through the IPS bus except some secret key regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA. In addition, CTRL[ERROR] will be set. CTRL[ERROR] must be cleared by software before any new write, read, or reload access can be issued. Subsequent reads to unlocked shadow registers will still work successfully.

To read fuse words directly from the fusebox instead of the shadow registers, complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a read access can be requested.
2. Write the requested fuse address to CTRL[ADDR].
3. Set READ_CTRL[READ_FUSE] to start the read operation.
4. Wait for the controller to clear CTRL[BUSY]. A read request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition, CTRL[ERROR] will be set. CTRL[ERROR] must be cleared by software before any new access can be issued.
5. Read READ_FUSE_DATA to get fuse word value. READ_FUSE_DATA will be 0xBADABADA when CTRL[ERROR] is set.
6. After reading is completed, it is necessary to check OUT_STATUS[DED]. If it is 1, read fuse value is wrong, otherwise it is correct.

73.3.1.3 Fuse Writes

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank, complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a write access can be requested.
2. Write the requested address to CTRL[ADDR] and program the unlock code into CTRL[WR_UNLOCK]. The unlock code must be programmed for each write access. The unlock code is documented in the register description. Both the unlock code and address can be written in the same operation.
3. When programming a fuse word, OCOTP_CTRL will auto select ECC or redundancy mode to program the fuse word according to fuse map definition. In ECC mode, the 32 fuse bits in one word can only be written one time. In redundancy mode, the word can be written more than one time as long as they are different fuse bits. For the fuse word which needs be programmed with ECC mode, it is recommended to set CTRL[WORDLOCK] with unlock code and address in the same operation.
4. Write the data to the DATA register. This will automatically set CTRL[BUSY] and clear CTRL[WR_UNLOCK]. Bit fields with 1's will result in that OTP bit being programmed. Bit fields with 0's will be ignored. At the same time that the write is accepted, the controller makes an internal copy of CTRL[ADDR] which cannot be updated until the next write sequence is initiated. This copy guarantees that erroneous writes to CTRL[ADDR] will not affect an active write operation. During the write operation, DATA cannot be modified. If the requested fuse word is locked, the programming request will not be performed. Checking the appropriate LOCKEDn register before attempting to program is recommended
5. Wait for the controller to clear CTRL[BUSY]. A write request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition CTRL[ERROR] will be set. CTRL[ERROR] must be cleared by software before any new write access can be issued.
6. Check the values of OUT_STATUS[PROGFAIL] and OUT_STATUS[LOCKED]. If either of these two bits is set, it indicates that programming has failed.

It should be noted that write latencies to OTP are multiples of 10 micro-seconds per word. The write latency is based on the number of "1" bits being written. For example, to program half the fuse bits in one 32-bt word requires approximately 10 µs x 16. The RT700 family device datasheet provides a guaranteed maximum programming time for per bit

73.3.1.4 Bank redundancy vs ECC

There are two mechanisms used by the fuse block to ensure that its values are correct. Each of these is performed on a word by word basis, however, the selected option is hardwired on a bank by bank basis.

Fuses can generally only be set from 0 to 1 and the most common failure is a 1 appearing as a 0. The simplest mechanism to verify these failures is redundancy. Bank redundancy is when two bits are OR'd together so that if either is set, then the output value is a 1.

In this case only 16 of the 32 bits are available (bits 15:0 are automatically copied to 31:16) and checked on read. The value here is that each pair of bits can be programmed at different times. This allows for multiple programming operations on a word, so bits can be configured at different times.

NOTE

In shadow registers, the inverted version of [15:0] will show in [31:16]. In the fuse array, [31:16] is the same as [15:0].

The more complex ECC mechanism is used when higher numbers of bits must be stored and configured at the same time in a group of 32. When written, the ECC must be calculated to allow an ECC correction. When read, any one bit of failure will be corrected, 2 bits will be detected (and an error flag asserted) by the controller.

NOTE

As the ECC is generated for the whole word, the whole word must be setup at one time and individual bits cannot be modified independently.

73.3.1.5 Error Conditions

The CTRL[ERROR] bit will be set under the following conditions:

- A read is performed to a shadow register which has been read locked.
- A program is performed to a non-existent fuse word
- A read is performed to a fuse word which has been read locked.
- Overlapping access - a read, write, or reload is attempted while a read, write, or reload is already in progress.
- A write request to a protected or locked region.

73.3.1.6 Write Postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations (direct reads, shadow reloads, or other writes) following a write must wait at least 2 µs after CTRL[BUSY] clears. The delay guarantees programming voltages on-chip reach a steady state when exiting a write sequence.

A recommended software sequence to meet the postamble requirements is as follows:

1. Issue the write and poll for CTRL[BUSY].
2. After CTRL[BUSY] is clear, wait 2 µs.
3. Perform the next OTP operation.

73.3.2 Fuse Shadow Memory Footprint

The OTP memory footprint is shown in the following figure. The registers are grouped by lock region. Their names correspond to the fusemap names.

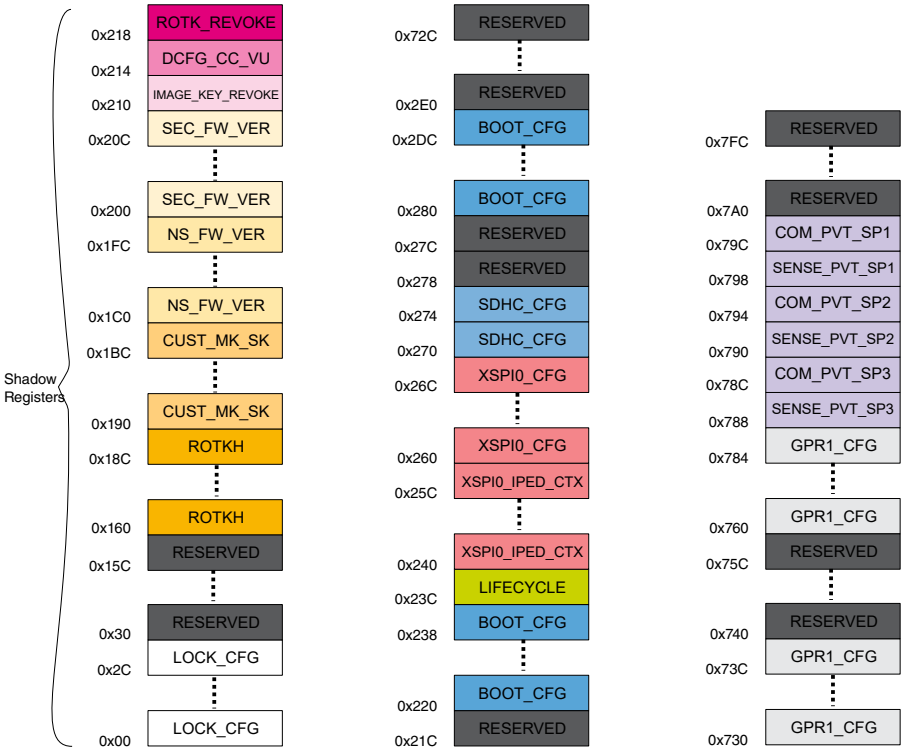


Figure 309. OTP Memory Footprint

73.3.3 Clocking

The table found here describes the clock sources for OCOTP. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 703. OCOTP Clocks

Clock name	Description
ipg_clk	Peripheral clock to configure registers and function module.

73.3.4 Resets

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time CTRL[BUSY] is set.

73.4 External Signals

There are no external signals pinned out for OCOTP.

73.5 Fuse Map

See the Fusemap chapter of this reference manual for more information.

73.6 Memory Map/Register Definition

The OCOTP Memory Map/Register Definition can be found here.

NOTE

Writing or reading an unimplemented register address in the OCOTP Controller will not send a bus error and read data will be 0.

73.6.1 register descriptions

73.6.1.1 OCOTP memory map

OCOTP base address: 4001_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 79Ch	OTP Shadow Register (OTP_SHADOW0 - OTP_SHADOW487)	32	RW	See section
800h	OTP Controller Control Register (HW_OCOTP_CTRL)	32	RW	0000_0000h
808h	OTP Controller Write Data Register (HW_OCOTP_WRITE_DATA)	32	RW	0000_0000h
80Ch	OTP Controller Read Ctrl Register (HW_OCOTP_READ_CTRL)	32	RW	0000_0000h
810h	OTP Controller Read Data Register (HW_OCOTP_READ_DATA)	32	RW	FFFF_FFFFh
814h	OTP Controller Clock Divider register (HW_OCOTP_CLK_DIV)	32	RW	0000_0001h
820h	OTP Controller Status Register (HW_OCOTP_STATUS)	32	RW	0000_0000h
828h	OTP Controller Version Register (HW_OCOTP_VERSION)	32	R	0800_0000h

Table continues on the next page...

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
82Ch	OTP Controller Nonmask Status1 Register (HW_OCOTP_NONMASK_STATUS1)	32	RW	0000_0000h
830h	OTP Controller Mask Ctrl1 Register (HW_OCOTP_MASK_CTRL1)	32	RW	0000_0000h
834h	OTP Controller Nonmask Status1 Register (HW_OCOTP_MASK_STATUS1)	32	RW	0000_0000h
87Ch	OTP Controller Lock Sticky Register0 (HW_OCOTP_LOCK_STICKY_0)	32	RW	0000_0000h
880h	OTP Controller Lock Sticky Register1 (HW_OCOTP_LOCK_STICKY_1)	32	RW	0000_0000h
884h	OTP Controller Lock Sticky Register2 (HW_OCOTP_LOCK_STICKY_2)	32	RW	0000_0000h
888h	OTP Controller Lock Sticky Register3 (HW_OCOTP_LOCK_STICKY_3)	32	RW	0000_0000h
88Ch	OTP Controller Lock Sticky Register4 (HW_OCOTP_LOCK_STICKY_4)	32	RW	0000_0000h
890h	OTP Controller Lock Sticky Register5 (HW_OCOTP_LOCK_STICKY_5)	32	RW	0000_0000h

73.6.1.2 OTP Shadow Register (OTP_SHADOW0 - OTP_SHADOW487)

Offset

For a = 0 to 487:

Register	Offset
OTP_SHADOWa	0h + (a × 4h)

Function

NOTE

Each fuse word corresponds to a dedicated OTP_SHADOWa (a is the fuse word index) register. Not all the OTP_SHADOWa registers are available, see Fusemap for details.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHADOW															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHADOW															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 SHADOW	OTP shadow register, fsb have read access of shadow 0-487.

73.6.1.3 OTP Controller Control Register (HW_OCOTP_CTRL)

Offset

Register	Offset
HW_OCOTP_CTRL	800h

Function
Control/address register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	WR_UNLOCK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WORD LOCK	SUPP ADD	RED_MUX_SEL		RELO AD_...	SRAM _EN...	PRNG _OU...	ADDR								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 WR_UNLOCK	Write 0x3E77 to enable OTP write accesses. NOTE: The write operation must be unlocked for each word by writing 0x3E77 to WR_UNLOCK field. Then writing to OTP_WRITE_DATA regsiter will automatically start the programming procedure.
15 WORDLOCK	Set to write-lock the fuse word when it's being programming.
14 SUPPADD	This bit should be set when programming pr reading the supplementary address space (aka test row).
13-12 RED_MUX_SEL	For fuse word with Redundancy mode, if set red_mux_sel to 0, low 16bit fuse OR high 16bit fuse value can be read, if set red_mux_sel to 1, low 16bit fuse value can be read, if set red_mux_sel to 2, high 16bit fuse value can be read. For fuse word with bit protect, if set red_mux_sel to 0, low 16bit fuse OR high 16bit fuse value can be read and low 16bit shadow OR high 16bit shadow can be read, if set red_mux_sel to 1, low 16bit fuse value can be read and low 16bit shadow can be read, if set red_mux_sel to 2, high 16bit fuse value can be read and high 16bit shadow value can be read.
11 RELOAD_SHADOWS	Set to force re-loading the shadow registers (HW/SW capability and LOCK). This operation will automatically set OTP_STATUS.BUSY. Once the shadow registers have been re-loaded, OTP_STATUS.BUSY and RELOAD_SHADOWS are automatically cleared by the controller.
10 SRAM_ENTROPY_DATA_EN	Default is 0: enable read value of sram entropy data, disable read when set to 1. This bit can only be cleared by POR.
9 PRNG_OUTPUT_DATA_EN	Default is 0: enable read value of prng out data, disable read when set to 1. This bit can only be cleared by POR.
8-0 ADDR	OTP word address for read/programming.

73.6.1.4 OTP Controller Write Data Register (HW_OCOTP_WRITE_DATA)

Offset

Register	Offset
HW_OCOTP_WRITE_DATA	808h

Function
OTP programming data register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRITE_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRITE_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 WRITE_DATA	Fuse word programming data. After the write operation is unlocked in OTP_CTRL register, writing dat to this register automatically start the programming proceduce.

73.6.1.5 OTP Controller Read Ctrl Register (HW_OCOTP_READ_CTRL)

Offset

Register	Offset
HW_OCOTP_READ_CTRL	80Ch

Function

OTP read start register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															READ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 READ	Writing 1 to start a fuse word read operation. It's cleared by hardware when the read operation is completed.

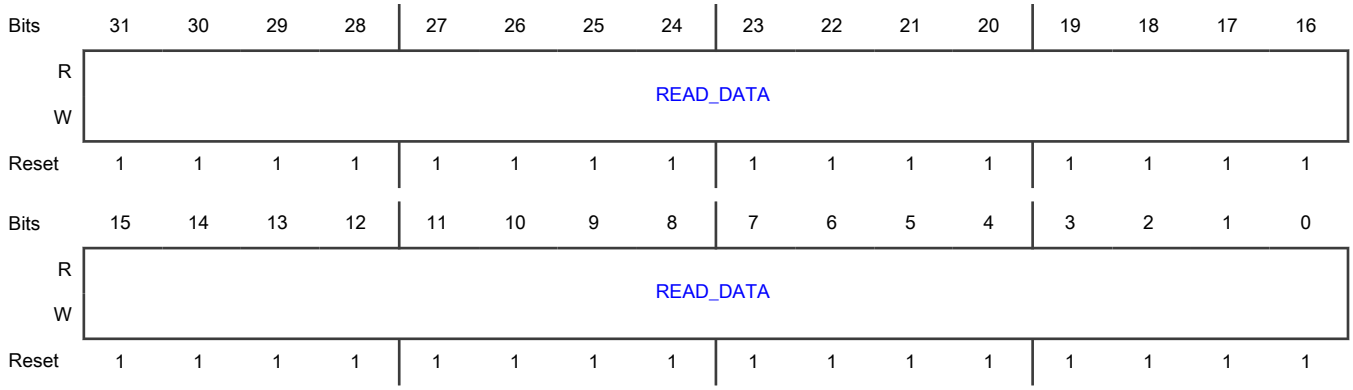
73.6.1.6 OTP Controller Read Data Register (HW_OCOTP_READ_DATA)

Offset

Register	Offset
HW_OCOTP_READ_DATA	810h

Function
OTP read data register.

Diagram



Fields

Field	Function
31-0 READ_DATA	Fuse word read data from read operation.When write any value to it, it will be clear to all 1.

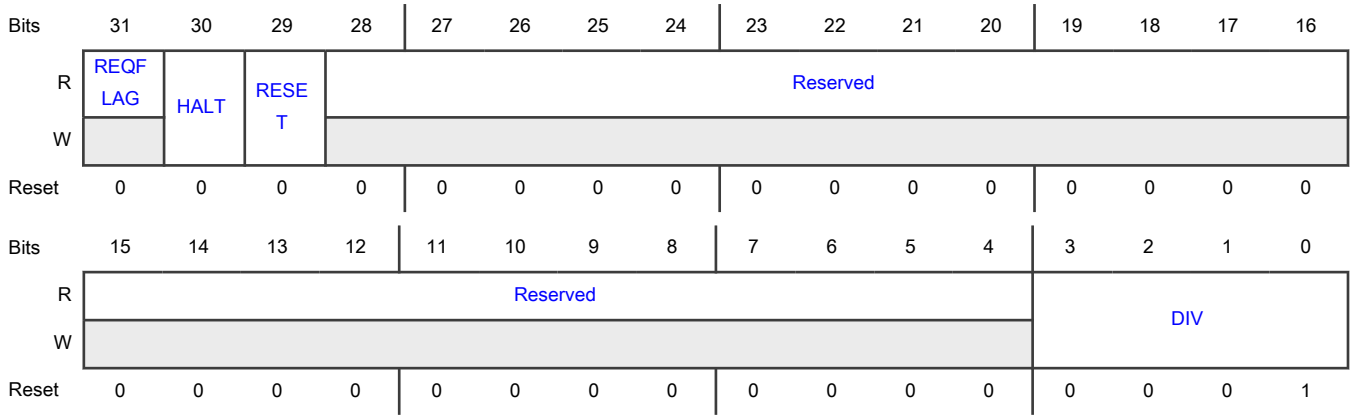
73.6.1.7 OTP Controller Clock Divider register (HW_OCOTP_CLK_DIV)

Offset

Register	Offset
HW_OCOTP_CLK_DIV	814h

Function
OTP clock divider register.

Diagram



Fields

Field	Function
31 REQFLAG	Divider status flag. Set when a change is made to the divider value, cleared when the change is complete.
30 HALT	Halts the divider counter. The intent is to allow the divider's clock source to be changed without the risk of a glitch at the output.
29 RESET	Resets the divider counter. Can be used to make sure a new divider value is used right away rather than completing the previous count.
28-4 —	Reserved
3-0 DIV	Clock divider value by 1 encoding. It's uesd to generate the clock to OTP memory (otp_clk) with apb_clk. The maximum otp_clk frequency is 120Mhz. 0 : Divide by 1 ... 15 : Divider by 16

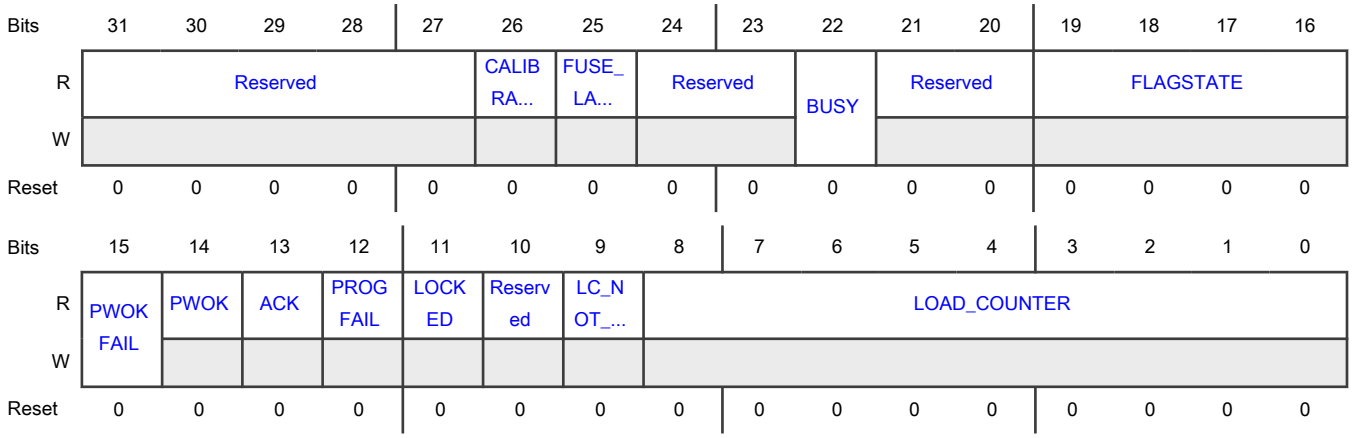
73.6.1.8 OTP Controller Status Register (HW_OCOTP_STATUS)

Offset

Register	Offset
HW_OCOTP_STATUS	820h

Function
Status register.

Diagram



Fields

Field	Function
31-27 —	Reserved
26 CALIBRATED	OTP CALIBRATED status.
25 FUSE_LATCHED	Indicate all shadows registers have been loaded with their corresponding fuse words when set by the controller after reset.
24-23 —	Reserved
22 BUSY	OTP controller status bit. When active, no new write or read access to OTP (including RELOAD_SHADOWS) can be performed. Cleared by the controller when the access completes. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller and cleared after all the shadow registers are successfully loaded.
21-20	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Function
—	
19-16 FLAGSTATE	OTP FLAGSTATE[3:0] value.
15 PWOKFAIL	OTP PWOK fail indicate. When PWOK transitions from 1 to 0, PWOKFAIL will be high, and write 1 to clear.
14 PWOK	OTP PWOK value.
13 ACK	OTP ACK value.
12 PROGFAIL	OTP PROGFAIL status. Write 1 to clear.
11 LOCKED	OTP LOCKED status during read/write operation. Write 1 to clear.
10 —	Reserved
9 LC_NOT_BLANK_STICKY	indicate lc not blank has been blown.
8-0 LOAD_COUNTER	store load fuse number when load finished.

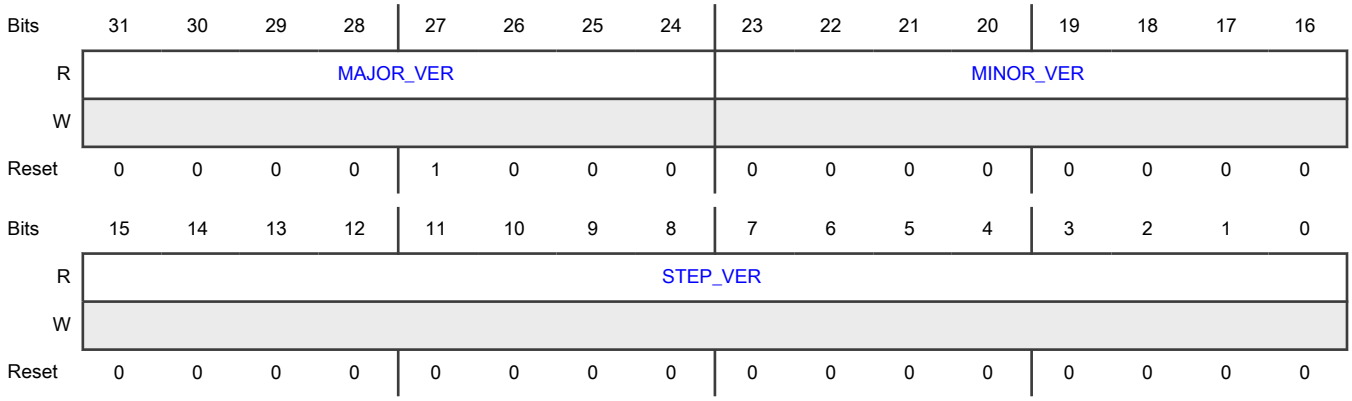
73.6.1.9 OTP Controller Version Register (HW_OCOTP_VERSION)

Offset

Register	Offset
HW_OCOTP_VERSION	828h

Function
VERSION ID register.

Diagram



Fields

Field	Function
31-24 MAJOR_VER	OTP controller major version.
23-16 MINOR_VER	OTP controller minor version.
15-0 STEP_VER	OTP controller step version.

73.6.1.10 OTP Controller Nonmask Status1 Register (HW_OCOTP_NONMASK_STATUS1)

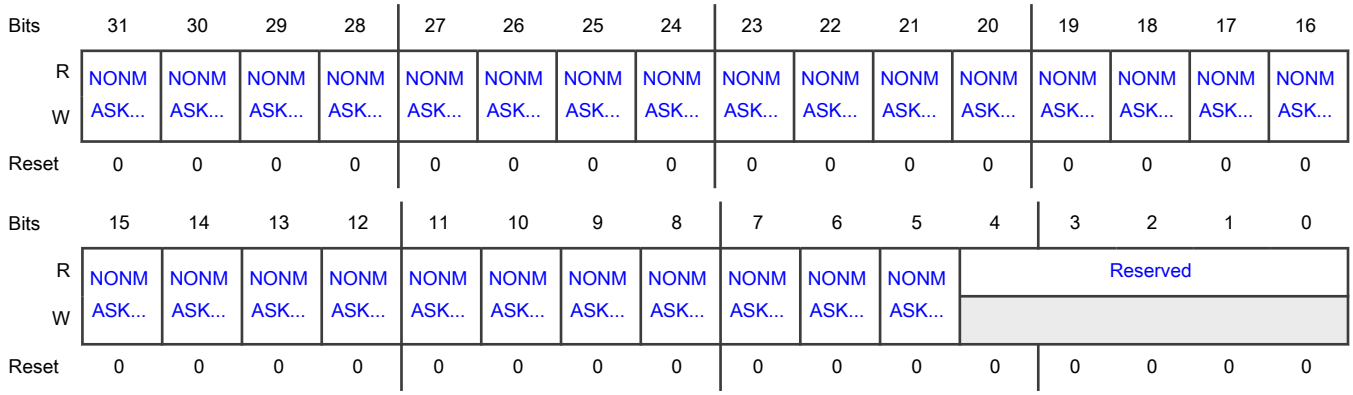
Offset

Register	Offset
HW_OCOTP_NONMASK_STATUS1	82Ch

Function

Status1 register without mask.

Diagram



Fields

Field	Function
31 NONMASK_SHADOW_NO_ACCESS	Asserted if access non-exist shadow register without mask. set 1 to clear.
30 NONMASK_FUSE_PROG_ERR	Asserted if fuse prog error is generated. set 1 to clear.
29 NONMASK_FUSE_READ_ERR	Asserted if fuse read error is generated. Set 1 to clear.
28 NONMASK_SHADOW_SRAM_WRITE_ERR	Asserted if shadow sram write error is generated. Set 1 to clear.
27 NONMASK_SHADOW_SRAM_READ_ERR	Asserted if shadow sram read error is generated. Set 1 to clear.
26 NONMASK_SHADOW_HVF_WRITE_ERR	Asserted if shadow hvf write error is generated. Set 1 to clear.
25	Asserted if shadow hvf read error is generated. Set 1 to clear.

Table continues on the next page...

Table continued from the previous page...

Field	Function
NONMASK_SHADOW_HVF_READ_ERR	
24 NONMASK_WRITE_DURING_RELOAD_ERR	Asserted if writing operation occurs during reload. Set 1 to clear.
23 NONMASK_RELOAD_REQ_ERR	Asserted if reload operation occurs when lmda_ocotp_no_reload equal to 1. Set 1 to clear.
22 NONMASK_FUSE_ACCESS_ERR	Asserted if fuse read/prog operation occurs when lmda_ocotp_no_fuse_access equal to 1. Set 1 to clear.
21 NONMASK_ECC_ZEROIZED_ERR	Asserted if zeroized fuse word is detetced. Set 1 to clear.
20 NONMASK_PRNG_O_FAULT	Asserted if output signal o_default of prng module is not equal to 1. Set 1 to clear.
19 NONMASK_BITPROTECT_ERR	Asserted if bit protect error is generated. Set 1 to clear.
18 NONMASK_COUNTER_ERR	Asserted if load counter is not equal to fuse word number when load finished. Set 1 to clear.
17 NONMASK_LOAD_ERR	Asserted if load glitch is detected. Set 1 to clear.
16 NONMASK_WRITE_ERR	Asserted if write glitch is detected. Set 1 to clear.

Table continues on the next page...

Table continued from the previous page...

Field	Function
15 NONMASK_CRC_SECURTY_PROTECT_ERR	Asserted if not in secured state when SECURITY PROTECTION bit in associated CRC_LUT register is set. Set 1 to clear.
14 NONMASK_CRC_WRITE_PROTECT_ERR	Asserted if write access occurs when WRITE PROTECTION bit in associated CRC_LUT register is set. Set 1 to clear.
13 NONMASK_CRC_ADDRESS_RANGE_ERR	Asserted if CRC_START_ADDR and CRC_END_ADDR bits in associated CRC_LUT register is out of range, both CRC_END_ADDR is smaller than CRC_START_ADDR OR CRC_SATRT_ADDR is larger than 124 OR CRC_END_ADDR is larger than 128 will cause out of range. Set 1 to clear.
12 NONMASK_CRC_LUT_SEL_ERR	Asserted if CRC_LUT_SEL bits in CRC_CTRL register is out range ([1,8]) when CRC_RUN bit in CRC_CTRL register is set. Set 1 to clear.
11 NONMASK_CRC_DONE	Asserted if CRC done. Set 1 to clear.
10 NONMASK_SEC	OTP SEC status during read operation. Set 1 to clear.
9 NONMASK_SEC_RELOAD	OTP SEC status during reload process. Set 1 to clear.
8 NONMASK_DED	OTP DED status during read operation. Set 1 to clear.
7 NONMASK_DED_RELOAD	OTP DED status during reload process. Set 1 to clear.
6 NONMASK_OTP_STATE_ERR	Asserted if state machine go to invalid state. Set 1 to clear.

Table continues on the next page...

Table continued from the previous page...

Field	Function
5 NONMASK_PBRICK_ERR	Asserted if ctrl_ocotp_nxp_zeroization_mask_fuse_wp and ctrl_ocotp_nxp_zeroization_mask_fuse_wp_n are both 0/1 OR ctrl_ocotp_oem_zeroization_mask_wp and ctrl_ocotp_oem_zeroization_mask_wp_n are both 0/1. Set 1 to clear.
4-0 —	Reserved

73.6.1.11 OTP Controller Mask Ctrl1 Register (HW_OCOTP_MASK_CTRL1)

Offset

Register	Offset
HW_OCOTP_MASK_CTRL1	830h

Function

Used for mask HW_OCOTP_NONMASK_STATUS1 register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	CTRL_MA...	Reserved	TMEOUT_...	TIMEOUT_COUNTER		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 CTRL_MASK_SHADOW_NO_ACCESS	Used for mask NONMASK_SHADOW_NO_ACCESS bit in NONMASK_STATUS1 register.
30	Used for mask NONMASK_FUSE_PROG_ERR bit in NONMASK_STATUS1 register.

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTRL_MASK_FUSE_PROG_ERR	
29 CTRL_MASK_FUSE_READ_ERR	Used for mask NONMASK_FUSE_READ_ERR bit in NONMASK_STATUS1 register.
28 CTRL_MASK_SHADOW_SRAM_WRITE_ERR	Used for mask NONMASK_SHADOW_SRAM_WRITE_ERR bit in NONMASK_STATUS1 register.
27 CTRL_MASK_SHADOW_SRAM_READ_ERR	Used for mask NONMASK_SHADOW_SRAM_READ_ERR bit in NONMASK_STATUS1 register.
26 CTRL_MASK_SHADOW_HVF_WRITE_ERR	Used for mask NONMASK_SHADOW_HVF_WRITE_ERR bit in NONMASK_STATUS1 register.
25 CTRL_MASK_SHADOW_HVF_READ_ERR	Used for mask NONMASK_SHADOW_HVF_READ_ERR bit in NONMASK_STATUS1 register.
24 CTRL_MASK_WRITE_DURING_RELOAD_ERR	Used for mask NONMASK_WRITE_DURING_RELOAD_ERR bit in NONMASK_STATUS1 register.
23 CTRL_MASK_RELOAD_REQ_ERR	Used for mask NONMASK_RELOAD_REQ_ERR bit in NONMASK_STATUS1 register.
22 CTRL_MASK_FUSE_ACCESS_ERR	Used for mask NONMASK_FUSE_ACCESS_ERR bit in NONMASK_STATUS1 register.
21	Used for mask NONMASK_ECC_ZEROIZED_ERR bit in NONMASK_STATUS1 register.

Table continues on the next page...

Table continued from the previous page...

Field	Function
CTRL_MASK_ECC_ZEROIZED_ERR	
20 CTRL_MASK_PRNG_O_FAULT	Used for mask NONMASK_PRNG_O_FAULT bit in NONMASK_STATUS1 register.
19 CTRL_MASK_BITPROTECT_ERR	Used for mask NONMASK_BITPROTECT_ERR bit in NONMASK_STATUS1 register.
18 CTRL_MASK_COUNTER_ERR	Used for mask NONMASK_COUNTER_ERR bit in NONMASK_STATUS1 register.
17 CTRL_MASK_LOAD_ERR	Used for mask NONMASK_LOAD_ERR bit in NONMASK_STATUS1 register.
16 CTRL_MASK_WRITE_ERR	Used for mask NONMASK_WRITE_ERR bit in NONMASK_STATUS1 register.
15 CTRL_MASK_CRC_SECURTY_PROTECT_ERR	Used for mask NONMASK_CRC_SECURTY_PROTECT_ERR bit in NONMASK_STATUS1 register.
14 CTRL_MASK_CRC_WRITE_PROTECT_ERR	Used for mask NONMASK_CRC_WRITE_PROTECT_ERR bit in NONMASK_STATUS1 register.
13 CTRL_MASK_CRC_ADDRESS_RANGE_ERR	Used for mask NONMASK_CRC_ADDRESS_RANGE_ERR bit in NONMASK_STATUS1 register.
12 CTRL_MASK_CRC_LUT_SEL_ERR	Used for mask NONMASK_CRC_LUT_SEL_ERR bit in NONMASK_STATUS1 register.

Table continues on the next page...

Table continued from the previous page...

Field	Function
11 CTRL_MASK_CRC_DONE	Used for mask NONMASK_CRC_DONE bit in NONMASK_STATUS1 register.
10 CTRL_MASK_SEC	Used for mask NONMASK_SEC bit in NONMASK_STATUS1 register.
9 CTRL_MASK_SEC_RELOAD	Used for mask NONMASK_SEC_RELOAD bit in NONMASK_STATUS1 register.
8 CTRL_MASK_DED	Used for mask NONMASK_DED bit in NONMASK_STATUS1 register.
7 CTRL_MASK_DED_RELOAD	Used for mask NONMASK_DED_RELOAD bit in NONMASK_STATUS1 register.
6 CTRL_MASK_OTP_STATE_ERR	Used for mask NONMASK_OTP_STATE_ERR bit in NONMASK_STATUS1 register.
5 CTRL_MASK_PBRICK_ERR	Used for mask NONMASK_PBRICK_ERR bit in NONMASK_STATUS1 register.
4 —	Reserved
3 TMEOUT_COUNTER_EN	Enable of timeout counter feature.
2-0 TIMEOUT_COUNTER	De-asserted cycle number of apb_ready.

73.6.1.12 OTP Controller Nonmask Status1 Register (HW_OCOTP_MASK_STATUS1)

Offset

Register	Offset
HW_OCOTP_MASK_STATUS1	834h

Function
Status1 register with mask.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MASK_SH...	MASK_FU...	MASK_FU...	MASK_SH...	MASK_SH...	MASK_SH...	MASK_SH...	MASK_WR...	MASK_RE...	MASK_FU...	MASK_EC...	MASK_PR...	MASK_BI...	MASK_CO...	MASK_LO...	MASK_WR...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MASK_CR...	MASK_CR...	MASK_CR...	MASK_CR...	MASK_CR...	MASK_SEC	SEC_REL...	MASK_DED	DED_REL...	MASK_OT...	MASK_PB...	Reserved				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MASK_SHADOW_NO_ACCESS	asserted if access non-exist shadow register with mask.
30 MASK_FUSE_PROG_ERR	Asserted if fuse prog error is generated.
29 MASK_FUSE_READ_ERR	Asserted if fuse read error is generated.
28 MASK_SHADOW_SRAM_WRITE_ERR	Asserted if shadow sram write error is generated.

Table continues on the next page...

Table continued from the previous page...

Field	Function
27 MASK_SHADOW_SRAM_READ_ERR	Asserted if shadow sram read error is generated.
26 MASK_SHADOW_HVF_WRITE_ERR	Asserted if shadow hvf write error is generated.
25 MASK_SHADOW_HVF_READ_ERR	Asserted if shadow hvf read error is generated.
24 MASK_WRITE_DURING_RELOAD_ERR	Asserted if writing operation occurs during reload.
23 MASK_RELOAD_REQ_ERR	Asserted if reload operation occurs when lmda_ocotp_no_reload equal to 1.
22 MASK_FUSE_ACCESS_ERR	Asserted if fuse read/prog operation occurs when lmda_ocotp_no_fuse_access equal to 1.
21 MASK_ECC_ZEROIZED_ERR	Asserted if zeroized fuse word is detetced.
20 MASK_PRNG_O_FAULT	Asserted if output signal o_default of prng module is not equal to 1.
19 MASK_BITPROTECT_ERR	Asserted if bit protect error is generated.
18 MASK_COUNTER_ERR	Asserted if load counter is not equal to fuse word number when load finished.
17	Asserted if load glitch is detected.

Table continues on the next page...

Table continued from the previous page...

Field	Function
MASK_LOAD_ERR	
16 MASK_WRITE_ERR	Asserted if write glitch is detected.
15 MASK_CRC_SECURTY_PROTECT_ERR	Asserted if not in secured state when SECURITY PROTECTION bit in associated CRC_LUT register is set.
14 MASK_CRC_WRITE_PROTECT_ERR	Asserted if write access occurs when WRITE PROTECTION bit in associated CRC_LUT register is set.
13 MASK_CRC_ADDRESS_RANGE_ERR	Asserted if CRC_START_ADDR and CRC_END_ADDR bits in associated CRC_LUT register is out of range, both CRC_END_ADDR is smaller than CRC_START_ADDR OR CRC_SATRT_ADDR is larger than 124 OR CRC_END_ADDR is larger than 128 will cause out of range.
12 MASK_CRC_LUT_SEL_ERR	Asserted if CRC_LUT_SEL bits in CRC_CTRL register is out range ([1,8]) when CRC_RUN bit in CRC_CTRL register is set.
11 MASK_CRC_DONE	Asserted if CRC done.
10 MASK_SEC	OTP SEC status during read operation.
9 SEC_RELOAD	OTP SEC status during reload process.
8 MASK_DED	OTP DED status during read operation.
7 DED_RELOAD	OTP DED status during reload process.
6	Asserted if state machine go to invalid state.

Table continues on the next page...

Table continued from the previous page...

Field	Function
MASK_OTP_STATE_ERR	
5 MASK_PBRICK_ERR	Asserted if ctrl_ocotp_nxp_zeroization_mask_fuse_wp and ctrl_ocotp_nxp_zeroization_mask_fuse_wp_n are both 0/1 OR ctrl_ocotp_oem_zeroization_mask_wp and ctrl_ocotp_oem_zeroization_mask_wp_n are both 0/1.
4-0 —	Reserved

73.6.1.13 OTP Controller Lock Sticky Register0 (HW_OCOTP_LOCK_STICKY_0)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_0	87Ch

Function
Lock sticky register0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LOCK_STICKY_BIT31	Lock sticky bit31.

Table continues on the next page...

Table continued from the previous page...

Field	Function
30 LOCK_STICKY_BIT30	Lock sticky bit30.
29 LOCK_STICKY_BIT29	Lock sticky bit29.
28 LOCK_STICKY_BIT28	Lock sticky bit28.
27 LOCK_STICKY_BIT27	Lock sticky bit27.
26 LOCK_STICKY_BIT26	Lock sticky bit26.
25 LOCK_STICKY_BIT25	Lock sticky bit25.
24 LOCK_STICKY_BIT24	Lock sticky bit24.
23 LOCK_STICKY_BIT23	Lock sticky bit23.
22 LOCK_STICKY_BIT22	Lock sticky bit22.
21 LOCK_STICKY_BIT21	Lock sticky bit21.
20 LOCK_STICKY_BIT20	Lock sticky bit20.

Table continues on the next page...

Table continued from the previous page...

Field	Function
19 LOCK_STICKY_BIT19	Lock sticky bit19.
18 LOCK_STICKY_BIT18	Lock sticky bit18.
17 LOCK_STICKY_BIT17	Lock sticky bit17.
16 LOCK_STICKY_BIT16	Lock sticky bit16.
15 LOCK_STICKY_BIT15	Lock sticky bit15.
14 LOCK_STICKY_BIT14	Lock sticky bit14.
13 LOCK_STICKY_BIT13	Lock sticky bit13.
12 LOCK_STICKY_BIT12	Lock sticky bit12.
11 LOCK_STICKY_BIT11	Lock sticky bit11.
10 LOCK_STICKY_BIT10	Lock sticky bit10.
9 LOCK_STICKY_BIT9	Lock sticky bit9.

Table continues on the next page...

Table continued from the previous page...

Field	Function
8 LOCK_STICKY_BIT8	Lock sticky bit8.
7 LOCK_STICKY_BIT7	Lock sticky bit7.
6 LOCK_STICKY_BIT6	Lock sticky bit6.
5 LOCK_STICKY_BIT5	Lock sticky bit5.
4 LOCK_STICKY_BIT4	Lock sticky bit4.
3 LOCK_STICKY_BIT3	Lock sticky bit3.
2 LOCK_STICKY_BIT2	Lock sticky bit2.
1 LOCK_STICKY_BIT1	Lock sticky bit1.
0 LOCK_STICKY_BIT0	Lock sticky bit0.

73.6.1.14 OTP Controller Lock Sticky Register1 (HW_OCOTP_LOCK_STICKY_1)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_1	880h

Function

Lock sticky register1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LOCK_STICKY_BIT63	Lock sticky bit63.
30 LOCK_STICKY_BIT62	Lock sticky bit62.
29 LOCK_STICKY_BIT61	Lock sticky bit61.
28 LOCK_STICKY_BIT60	Lock sticky bit60.
27 LOCK_STICKY_BIT59	Lock sticky bit59.
26 LOCK_STICKY_BIT58	Lock sticky bit58.
25 LOCK_STICKY_BIT57	Lock sticky bit57.

Table continues on the next page...

Table continued from the previous page...

Field	Function
24 LOCK_STICKY_BIT56	Lock sticky bit56.
23 LOCK_STICKY_BIT55	Lock sticky bit55.
22 LOCK_STICKY_BIT54	Lock sticky bit54.
21 LOCK_STICKY_BIT53	Lock sticky bit53.
20 LOCK_STICKY_BIT52	Lock sticky bit52.
19 LOCK_STICKY_BIT51	Lock sticky bit51.
18 LOCK_STICKY_BIT50	Lock sticky bit50.
17 LOCK_STICKY_BIT49	Lock sticky bit49.
16 LOCK_STICKY_BIT48	Lock sticky bit48.
15 LOCK_STICKY_BIT47	Lock sticky bit47.
14 LOCK_STICKY_BIT46	Lock sticky bit46.

Table continues on the next page...

Table continued from the previous page...

Field	Function
13 LOCK_STICKY_BIT45	Lock sticky bit45.
12 LOCK_STICKY_BIT44	Lock sticky bit44.
11 LOCK_STICKY_BIT43	Lock sticky bit43.
10 LOCK_STICKY_BIT42	Lock sticky bit42.
9 LOCK_STICKY_BIT41	Lock sticky bit41.
8 LOCK_STICKY_BIT40	Lock sticky bit40.
7 LOCK_STICKY_BIT39	Lock sticky bit39.
6 LOCK_STICKY_BIT38	Lock sticky bit38.
5 LOCK_STICKY_BIT37	Lock sticky bit37.
4 LOCK_STICKY_BIT36	Lock sticky bit36.
3 LOCK_STICKY_BIT35	Lock sticky bit35.

Table continues on the next page...

Table continued from the previous page...

Field	Function
2 LOCK_STICKY_BIT34	Lock sticky bit34.
1 LOCK_STICKY_BIT33	Lock sticky bit33.
0 LOCK_STICKY_BIT32	Lock sticky bit32.

73.6.1.15 OTP Controller Lock Sticky Register2 (HW_OCOTP_LOCK_STICKY_2)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_2	884h

Function

Lock sticky register2.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Lock sticky bit95.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT95	
30 LOCK_STICKY_BIT94	Lock sticky bit94.
29 LOCK_STICKY_BIT93	Lock sticky bit93.
28 LOCK_STICKY_BIT92	Lock sticky bit92.
27 LOCK_STICKY_BIT91	Lock sticky bit91.
26 LOCK_STICKY_BIT90	Lock sticky bit90.
25 LOCK_STICKY_BIT89	Lock sticky bit89.
24 LOCK_STICKY_BIT88	Lock sticky bit88.
23 LOCK_STICKY_BIT87	Lock sticky bit87.
22 LOCK_STICKY_BIT86	Lock sticky bit86.
21 LOCK_STICKY_BIT85	Lock sticky bit85.
20	Lock sticky bit84.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT84	
19 LOCK_STICKY_BIT83	Lock sticky bit83.
18 LOCK_STICKY_BIT82	Lock sticky bit82.
17 LOCK_STICKY_BIT81	Lock sticky bit81.
16 LOCK_STICKY_BIT80	Lock sticky bit80.
15 LOCK_STICKY_BIT79	Lock sticky bit79.
14 LOCK_STICKY_BIT78	Lock sticky bit78.
13 LOCK_STICKY_BIT77	Lock sticky bit77.
12 LOCK_STICKY_BIT76	Lock sticky bit76.
11 LOCK_STICKY_BIT75	Lock sticky bit75.
10 LOCK_STICKY_BIT74	Lock sticky bit74.
9	Lock sticky bit73.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT73	
8 LOCK_STICKY_BIT72	Lock sticky bit72.
7 LOCK_STICKY_BIT71	Lock sticky bit71.
6 LOCK_STICKY_BIT70	Lock sticky bit70.
5 LOCK_STICKY_BIT69	Lock sticky bit69.
4 LOCK_STICKY_BIT68	Lock sticky bit68.
3 LOCK_STICKY_BIT67	Lock sticky bit67.
2 LOCK_STICKY_BIT66	Lock sticky bit66.
1 LOCK_STICKY_BIT65	Lock sticky bit65.
0 LOCK_STICKY_BIT64	Lock sticky bit60.

73.6.1.16 OTP Controller Lock Sticky Register3 (HW_OCOTP_LOCK_STICKY_3)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_3	888h

Function
Lock sticky register3.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LOCK_STICKY_BIT127	Lock sticky bit127.
30 LOCK_STICKY_BIT126	Lock sticky bit126.
29 LOCK_STICKY_BIT125	Lock sticky bit125.
28 LOCK_STICKY_BIT124	Lock sticky bit124.
27	Lock sticky bit123.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT123	
26 LOCK_STICKY_BIT122	Lock sticky bit122.
25 LOCK_STICKY_BIT121	Lock sticky bit121.
24 LOCK_STICKY_BIT120	Lock sticky bit120.
23 LOCK_STICKY_BIT119	Lock sticky bit119.
22 LOCK_STICKY_BIT118	Lock sticky bit118.
21 LOCK_STICKY_BIT117	Lock sticky bit117.
20 LOCK_STICKY_BIT116	Lock sticky bit116.
19 LOCK_STICKY_BIT115	Lock sticky bit115.
18 LOCK_STICKY_BIT114	Lock sticky bit114.
17 LOCK_STICKY_BIT113	Lock sticky bit113.
16	Lock sticky bit112.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT112	
15 LOCK_STICKY_BIT111	Lock sticky bit111.
14 LOCK_STICKY_BIT110	Lock sticky bit110.
13 LOCK_STICKY_BIT109	Lock sticky bit109.
12 LOCK_STICKY_BIT108	Lock sticky bit108.
11 LOCK_STICKY_BIT107	Lock sticky bit107.
10 LOCK_STICKY_BIT106	Lock sticky bit106.
9 LOCK_STICKY_BIT105	Lock sticky bit105.
8 LOCK_STICKY_BIT104	Lock sticky bit104.
7 LOCK_STICKY_BIT103	Lock sticky bit103.
6 LOCK_STICKY_BIT102	Lock sticky bit102.
5	Lock sticky bit101.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT101	
4 LOCK_STICKY_BIT100	Lock sticky bit100.
3 LOCK_STICKY_BIT99	Lock sticky bit99.
2 LOCK_STICKY_BIT98	Lock sticky bit98.
1 LOCK_STICKY_BIT97	Lock sticky bit97.
0 LOCK_STICKY_BIT96	Lock sticky bit96.

73.6.1.17 OTP Controller Lock Sticky Register4 (HW_OCOTP_LOCK_STICKY_4)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_4	88Ch

Function
Lock sticky register4.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...	LOCK_ST...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LOCK_STICKY_BIT159	Lock sticky bit159.
30 LOCK_STICKY_BIT158	Lock sticky bit158.
29 LOCK_STICKY_BIT157	Lock sticky bit157.
28 LOCK_STICKY_BIT156	Lock sticky bit156.
27 LOCK_STICKY_BIT155	Lock sticky bit155.
26 LOCK_STICKY_BIT154	Lock sticky bit154.
25 LOCK_STICKY_BIT153	Lock sticky bit153.
24 LOCK_STICKY_BIT152	Lock sticky bit152.

Table continues on the next page...

Table continued from the previous page...

Field	Function
23 LOCK_STICKY_BIT151	Lock sticky bit151.
22 LOCK_STICKY_BIT150	Lock sticky bit150.
21 LOCK_STICKY_BIT149	Lock sticky bit149.
20 LOCK_STICKY_BIT148	Lock sticky bit148.
19 LOCK_STICKY_BIT147	Lock sticky bit147.
18 LOCK_STICKY_BIT146	Lock sticky bit146.
17 LOCK_STICKY_BIT145	Lock sticky bit145.
16 LOCK_STICKY_BIT144	Lock sticky bit144.
15 LOCK_STICKY_BIT143	Lock sticky bit143.
14 LOCK_STICKY_BIT142	Lock sticky bit142.
13 LOCK_STICKY_BIT141	Lock sticky bit141.

Table continues on the next page...

Table continued from the previous page...

Field	Function
12 LOCK_STICKY_BIT140	Lock sticky bit140.
11 LOCK_STICKY_BIT139	Lock sticky bit139.
10 LOCK_STICKY_BIT138	Lock sticky bit138.
9 LOCK_STICKY_BIT137	Lock sticky bit137.
8 LOCK_STICKY_BIT136	Lock sticky bit136.
7 LOCK_STICKY_BIT135	Lock sticky bit135.
6 LOCK_STICKY_BIT134	Lock sticky bit134.
5 LOCK_STICKY_BIT133	Lock sticky bit133.
4 LOCK_STICKY_BIT132	Lock sticky bit132.
3 LOCK_STICKY_BIT131	Lock sticky bit131.
2 LOCK_STICKY_BIT130	Lock sticky bit130.

Table continues on the next page...

Table continued from the previous page...

Field	Function
1 LOCK_STICKY_BIT129	Lock sticky bit129.
0 LOCK_STICKY_BIT128	Lock sticky bit128.

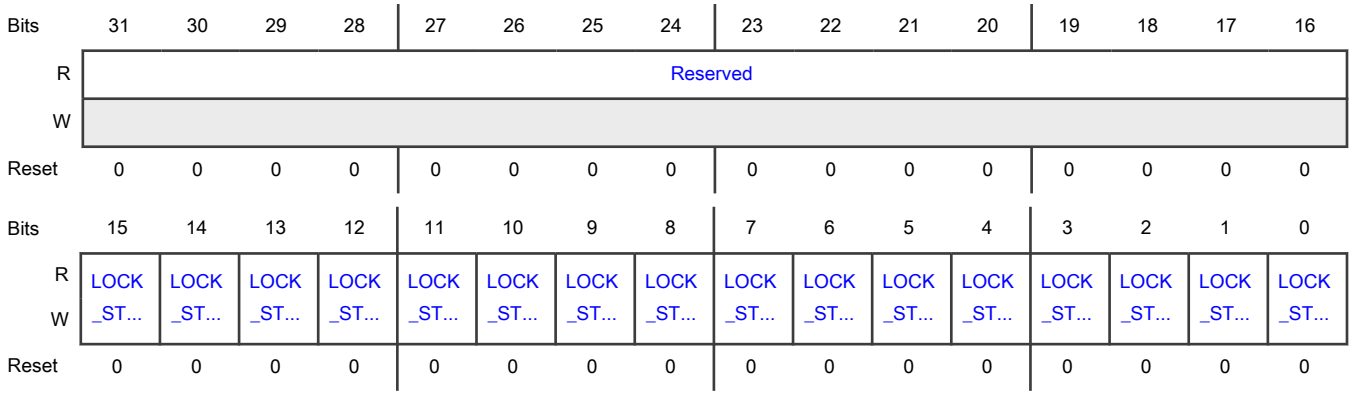
73.6.1.18 OTP Controller Lock Sticky Register5 (HW_OCOTP_LOCK_STICKY_5)

Offset

Register	Offset
HW_OCOTP_LOCK_STICKY_5	890h

Function
Lock sticky register5.

Diagram



Fields

Field	Function
31-16 —	Reserved
15	Lock sticky bit175.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT175	
14 LOCK_STICKY_BIT174	Lock sticky bit174.
13 LOCK_STICKY_BIT173	Lock sticky bit173.
12 LOCK_STICKY_BIT172	Lock sticky bit172.
11 LOCK_STICKY_BIT171	Lock sticky bit171.
10 LOCK_STICKY_BIT170	Lock sticky bit170.
9 LOCK_STICKY_BIT169	Lock sticky bit169.
8 LOCK_STICKY_BIT168	Lock sticky bit168.
7 LOCK_STICKY_BIT167	Lock sticky bit167.
6 LOCK_STICKY_BIT166	Lock sticky bit166.
5 LOCK_STICKY_BIT165	Lock sticky bit165.
4	Lock sticky bit164.

Table continues on the next page...

Table continued from the previous page...

Field	Function
LOCK_STICKY_BIT164	
3 LOCK_STICKY_BIT163	Lock sticky bit163.
2 LOCK_STICKY_BIT162	Lock sticky bit162.
1 LOCK_STICKY_BIT161	Lock sticky bit161.
0 LOCK_STICKY_BIT160	Lock sticky bit160.

Chapter 74
Cyclic Redundancy Check (CRC)

74.1 Chip-specific CRC information

Table 704. Reference links to related information

Topic	Related module	Reference
Full description	CRC	CRC
Peripheral memory map		Memory map overview
Power management	PMC	Power overview PMC
Clocking		Clock overview Clock controller

74.1.1 Module instances

This device has one instance of the CRC module, CRC.

Table 705. CRC instance

	CRC
Located in	VDD2_COMP
Access	CPU0, HiFi4, EZH-V, eDMA0, eDMA1

74.2 Overview

CRC generates 16-bit or 32-bit CRC codes output for error detection. You can calculate these codes for up to 32 bits input data at a time.

CRC provides a programmable polynomial and other parameters to meet 16-bit or 32-bit CRC standards.

74.2.1 Block diagram

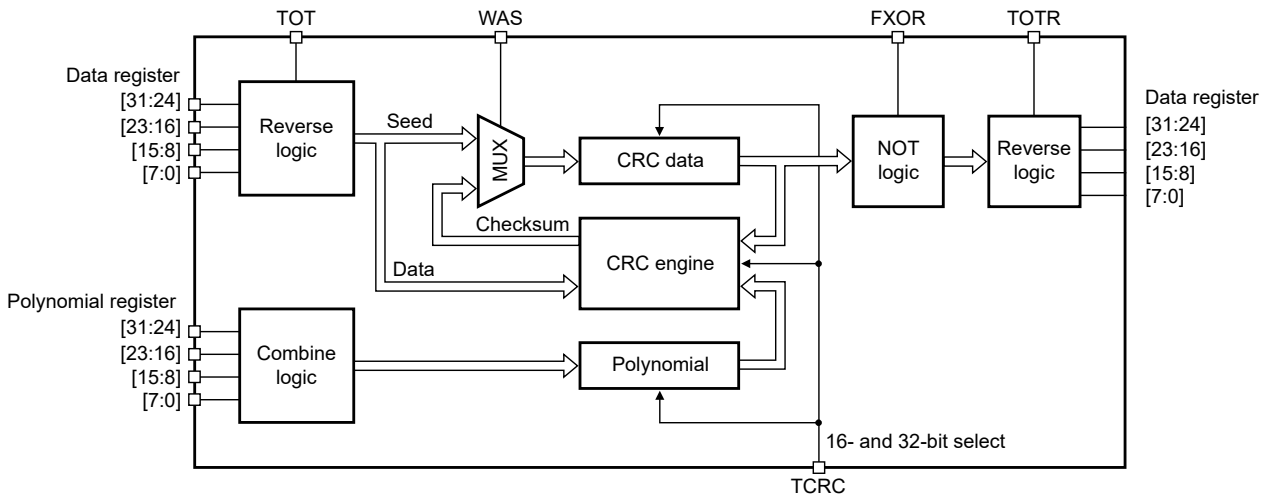


Figure 310. Block diagram

74.2.2 Features

CRC has the following features:

- Hardware CRC generator circuit using 16-bit or 32-bit programmable shift registers
- Programmable initial seed value and polynomial
- Transpose of input or output data (CRC result) in bitwise or bytewise (this option is required for certain CRC standards. You cannot perform a bytewise transpose operation when accessing DATA via 8-bit access.)
- Invert final CRC result
- 32-bit CPU register programming interface

74.3 Functional description

74.3.1 Modes of operation

The following sections describe various modes of operation that affect the functionality of CRC: Run mode and Low power mode.

74.3.1.1 Run mode

Run mode is the basic mode of operation.

74.3.1.2 Low-power modes

When the chip enters the low-power mode, the CRC module clock (ipg_clk and ipg_clk_s) is disabled and the in-progress CRC calculation stops. The calculation resumes after the CRC module clock is enabled or the chip exits low power mode via system reset.

NOTE

See the chip-specific power management chapter to know the specific low-power modes supported on your device.

74.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, you can program data values as 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes lead to an incorrect CRC calculation.

74.3.2.1 Calculating a 16-bit CRC

Perform these steps to calculate a 16-bit CRC:

1. Write 0 to CTRL[TCRC] to enable 16-bit CRC mode.
2. Program the transpose and complement options in Control (CTRL) as required for the CRC calculation.
3. Write a 16-bit polynomial to GPOLY[LOW].

   GPOLY[HIGH] is not usable in 16-bit CRC mode.
4. Write 1 to CTRL[WAS] to program the seed value.
5. Write a 16-bit seed to DATA[LU] and DATA[LL].

   DATA[HU] and DATA[HL] are not used.
6. Write 0 to CTRL[WAS] to start writing data values.
7. Write data values into DATA[LU], and DATA[LL]

   CRC is calculated on every data value write and the intermediate CRC result is stored back into DATA[LU] and DATA[LL]
8. After writing all the data values, read the final CRC result from DATA[LU] and DATA[LL].

CRC is calculated bytewise and two clocks are required to complete one CRC calculation.

The transpose and complement operations are performed on-the-fly when reading or writing values. See Transpose feature and Result complement for details.

74.3.2.2 Calculating a 32-bit CRC

Perform these steps to calculate a 32-bit CRC:

1. Write 1 to CTRL[TCRC] to enable 32-bit CRC mode.
2. Program the transpose and complement options in Control (CTRL) as required for CRC calculation. See Transpose feature and Result complement for details.
3. Write a 32-bit polynomial to GPOLY[HIGH] and GPOLY[LOW] .
4. Write 1 to CTRL[WAS] to program the seed value.
5. Write a 32-bit seed to DATA[HU], DATA[HL], DATA[LU], and DATA[LL].
6. Write 0 to CTRL[WAS] to start writing data values.
7. Write data values into DATA[HU], DATA[HL], DATA[LU], and DATA[LL]

   CRC is computed on every data value write and the intermediate CRC result is stored back into DATA .
8. After writing all the values, read the final CRC result from DATA[HU], DATA[HL], DATA[LU], and DATA[LL].

   CRC is calculated bytewise and two clocks are required to complete one CRC calculation.

The transpose and complement operations are performed on-the-fly when reading or writing values. See Transpose feature and Result complement for details.

74.3.3 Transpose feature

Transpose is not enabled by default. However, CRC requires input data and/or final checksum to be transposed. You have an option to configure each transpose operation separately to meet CRC standards. The data is transposed on-the-fly while being read or written.

Some protocols use the little-endian format for data stream to calculateCRC. In this case, transpose flips bits.

74.3.3.1 Types of transpose

CRC provides several types of transpose to flip bits and/or bytes for both writing input data and reading result separately using the CTRL[TOT] and CTRL[TOTR] according to the CRC calculation being used.

The following types of transpose are available for writing to and reading from DATA.

1. CTRL[TOT] or CTRL[TOTR] is 0.

   No transposition occurs.

2. CTRL[TOT] or CTRL[TOTR] is 1.

   Bits in a byte are transposed when bytes are not transposed.

   reg[31:0] becomes {reg[24:31], reg[16:23], reg[8:15], reg[0:7]}.

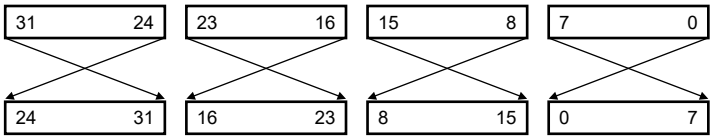


Figure 311. Transpose type 1b

3. CTRL[TOT] or CTRL[TOTR] is 10b.

   Both bits in bytes and bytes are transposed.

   reg[31:0] becomes {reg[0:7], reg[8:15], reg[16:23], reg[24:31]}.

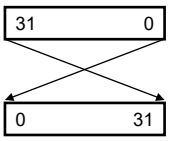


Figure 312. Transpose type 10b

4. CTRL[TOT] or CTRL[TOTR] is 11b.

   Bytes are transposed but bits are not transposed.

   reg[31:0] becomes {reg[7:0], reg[15:8], reg[23:16], reg[31:24]}.

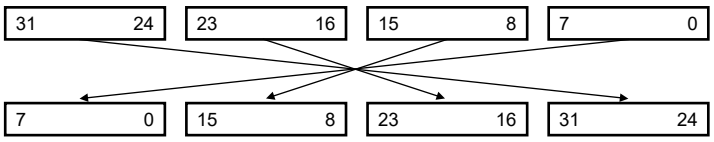


Figure 313. Transpose type 11b

NOTE

For 8-bit and 16-bit write accesses to Data (DATA), the data is transposed with 0s on the unused byte or bytes (taking 32 bits as a whole), but CRC is calculated on the valid byte(s) only. When reading the Data (DATA) for a 16-bit CRC result and using transpose options 10 and 11, the resulting value after transposition resides in DATA[HU] and DATA[HL]. You must account for this situation when reading the 16-bit CRC result, so reading 32 bits is preferred.

74.3.4 Result complement

When CTRL[FXOR] = 1, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in Data (DATA) every time Data (DATA) is read. When CTRL[FXOR] = 0,, reading Data (DATA) accesses the raw checksum value.

74.3.5 Clocking

Table 706. CRC clocks

Type of clock	Description
Bus clock (ipg_clk/ipg_clk_s)	ipg_clk_s controls the access to the CRC registers. ipg_clk and ipg_clk_s function the CRC module.

NOTE

See the chip-specific clocking chapter, to know the specific CRC clocks supported on your device.

74.3.6 Interrupts

This module has no interrupts.

74.4 External signals

There is no CRC signal that connects off chip.

74.5 Initialization

To enable CRC calculation, you must program:

- CTRL[WAS] .
- Polynomial (GPOLY).
- Parameters for transposition and CRC result inversion in the applicable registers.

Writing 1 to CTRL[WAS] enables you to program the seed value into CRC Data registers.

After writing all the data, you must wait for at least two clock cycles to read the data from CRC Data (DATA) register.

After a CRC calculation completes, you can reinitialize the module for a new CRC computation by again writing 1 to CTRL[WAS] and programming a new, or previously used, seed value. You must set all other parameters before programming the seed value and subsequent data values.

74.6 Use cases

The following tables use the little-endian format.

74.6.1 CTRL programming

The following table shows Control (CTRL) programming for 16-bit CRC.

Table 707. CTRL programming for 16-bit CRC

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-16_CCITT_FALSE	1021h	FFFFh	0	0	0000h	0h	0h	0h
CRC-16_ARC	8005h	0000h	1	1	0000h	1h	2h	0h

Table continues on the next page...

Table 707. CTRL programming for 16-bit CRC (continued)

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-16_AUG_CCITT	1021h	1D0Fh	0	0	0000h	0h	0h	0h
CRC-16_BUYPASS	8005h	0000h	0	0	0000h	0h	0h	0h
CRC-16_CCITT_ZERO	1021h	0000h	0	0	0000h	0h	0h	0h
CRC-16_CDMA2000	C867h	FFFFh	0	0	0000h	0h	0h	0h
CRC-16_DDS_110	8005h	800Dh	0	0	0000h	0h	0h	0h
CRC-16_DECT_X	589h	0000h	0	0	0000h	0h	0h	0h
CRC-16_DNP	3D65h	0000h	1	1	FFFFh	1h	2h	1h
CRC-16_EN_13757	3D65h	0000h	0	0	FFFFh	0h	0h	1h
CRC-16_GENIBUS	1021h	FFFFh	0	0	FFFFh	0h	0h	1h
CRC-16_MAXIM	8005h	0000h	1	1	FFFFh	1h	2h	1h
CRC-16_MCRF4XX	1021h	FFFFh	1	1	0000h	1h	2h	0h
CRC-16_RIELLO	1021h	B2AAh	1	1	0000h	1h	2h	0h
CRC-16_T10_DIF	8BB7h	0000h	0	0	0000h	0h	0h	0h
CRC-16_TELEDISK	A097h	0000h	0	0	0000h	0h	0h	0h
CRC-16_TMS37157	1021h	89ECh	1	1	0000h	1h	2h	0h
CRC-16_USB	8005h	FFFFh	1	1	FFFFh	1h	2h	1h
CRC-16_A	1021h	C6C6h	1	1	0000h	1h	2h	0h
CRC-16_KERMIT	1021h	0000h	1	1	0000h	1h	2h	0h
CRC-16_MODBUS	8005h	FFFFh	1	1	0000h	1h	2h	0h
CRC-16_X_25	1021h	FFFFh	1	1	FFFFh	1h	2h	1h
CRC-16_XMODEM	1021h	0000h	0	0	0000h	0h	0h	0h

The following table shows Control (CTRL) programming for 32-bit CRC.

Table 708. CTRL programming for 32-bit CRC

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-32	04C11DB7h	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32_BZIP2	04C11DB7h	FFFFFFFFh	0	0	FFFF_FFFFh	0h	0h	1h
CRC-32C	1EDC6F41h	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32D	A833982Bh	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32_MPEG-2	04C11DB7h	FFFFFFFFh	0	0	0000_0000h	0h	0h	0h
CRC-32_POSIX	04C11DB7h	00000000h	0	0	FFFF_FFFFh	0h	0h	1h
CRC-32Q	814141ABh	00000000h	0	0	0000_0000h	0h	0h	0h

Table continues on the next page...

Table 708. CTRL programming for 32-bit CRC (continued)

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-32_JAMCRC	04C11DB7h	FFFFFFFFh	1	1	0000_0000h	1h	2h	0h
CRC-32_XFER	000000AFh	00000000h	0	0	0000_0000h	0h	0h	0h

74.6.2 Expected read data fields

The following table shows the expected read data fields for 16-bit CRC.

Table 709. Expected read data fields for 16-bit CRC

Algorithm	Data (DATA)
CRC16_CCITT_FALSE	[31:16] = Unknown[15:0] = Valid data
CRC16_ARC	[31:16] = Valid data [15:0] = Unknown
CRC16_AUG_CCITT	[31:16] = Unknown [15:0] = Valid data
CRC16_BUYPASS	[31:16] = Unknown [15:0] = Valid data
CRC16_CCITT_ZERO	[31:16] = Unknown [15:0] = Valid data
CRC16_CDMA2000	[31:16] = Unknown [15:0] = Valid data
CRC16_DDS_110	[31:16] = Unknown [15:0] = Valid data
CRC16_DECT_X	[31:16] = Unknown [15:0] = Valid data
CRC16_DNP	[31:16] = Valid data [15:0] = Unknown
CRC-16_EN_13757	[31:16] = Unknown [15:0] = Valid data
CRC-16_GENIBUS	[31:16] = Unknown [15:0] = Valid data
CRC-16_MAXIM	[31:16] = Valid data [15:0] = Unknown
CRC-16_MCRF4XX	[31:16] = Valid data [15:0] = Unknown
CRC-16_RIELLO	[31:16] = Valid data [15:0] = Unknown
CRC-16_T10_DIF	[31:16] = Unknown [15:0] = Valid data
CRC-16_TELEDISK	[31:16] = Unknown [15:0] = Valid data
CRC-16_TMS37157	[31:16] = Valid data [15:0] = Unknown
CRC-16_USB	[31:16] = Valid data [15:0] = Unknown
CRC-16_A	[31:16] = Valid data [15:0] = Unknown
CRC-16_KERMIT	[31:16] = Valid data [15:0] = Unknown
CRC-16_MODBUS	[31:16] = Valid data [15:0] = Unknown
CRC-16_X_25	[31:16] = Valid data [15:0] = Unknown
CRC-16_XMODEM	[31:16] = Unknown [15:0] = Valid data

The following table shows the expected read data fields for 32-bit CRC.

Table 710. Expected read data fields for 32-bit CRC

Algorithm	Data (DATA)
CRC-32	[31:0] = Valid data
CRC-32_BZIP2	[31:0] = Valid data
CRC-32C	[31:0] = Valid data
CRC-32D	[31:0] = Valid data
CRC-32_MPEG-2	[31:0] = Valid data
CRC-32_POSIX	[31:0] = Valid data
CRC-32Q	[31:0] = Valid data
CRC-32_JAMCRC	[31:0] = Valid data
CRC-32_XFER	[31:0] = Valid data

74.7 Memory map and register descriptions

NOTE

You must reconfigure CRC engine in case an IPS transfer error occurs (ips_xfr_err).

The CRC module generates a transfer error in the following cases:

- Write accesses to the register addresses that are not mapped to the peripherals but included in the address spaces of the peripherals.

74.7.1 CRC register descriptions

74.7.1.1 CRC memory map

CRC base address: 4015_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Data (DATA)	32	RW	FFFF_FFFFh
4h	Polynomial (GPOLY)	32	RW	0000_1021h
8h	Control (CTRL)	32	RW	0000_0000h

74.7.1.2 Data (DATA)

Offset

Register	Offset
DATA	0h

Function

Configures the value of seed, data, and checksum. When CTRL[WAS] = 1, any write to this register is regarded as the seed value. When CTRL[WAS] becomes 0, any write to this register is regarded as data for general CRC calculation.

In 16-bit CRC mode, DATA[HU] and DATA[HL] are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, you can program to write 8 bits, 16 bits, or 32 bits in big endian order, provided all bytes are contiguous, with the MSB of data value written first.

After writing all data values, you can read the CRC result from DATA register. In 16-bit CRC mode, the CRC result is available in DATA[LU] and DATA[LL]. In 32-bit CRC mode, all fields contain the result. After writing all data, you must wait for at least two clock cycles to read the data from CRC data (DATA) register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HU								HL							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LU								LL							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24 HU	Upper Part of High Byte Generates CRC checksum in both 16-bit and 32-bit CRC modes if CTRL[WAS] = 0. • In 16-bit CRC mode (CTRL[TCRC] = 0), this field is not used for programming a seed value. • In 32-bit CRC mode (CTRL[TCRC] = 1), the values written to this field are part of the seed value when CTRL[WAS] = 1.
23-16 HL	Lower Part of High Byte Generates CRC checksum in both 16-bit and 32-bit CRC modes if CTRL[WAS] = 0. • In 16-bit CRC mode (CTRL[TCRC] = 0), this field is not used for programming a seed value. • In 32-bit CRC mode (CTRL[TCRC] = 1), the values written to this field are part of the seed value when CTRL[WAS] = 1.
15-8 LU	Upper Part of Low Byte Generates CRC checksum when CTRL[WAS] = 0. When CTRL[WAS] = 1,the values written to this field are part of the seed value.
7-0	Lower Part of Low Byte

Table continues on the next page...

Table continued from the previous page...

Field	Function
LL	Generates CRC checksum when CTRL[WAS] = 0. When CTRL[WAS] = 1, the values written to this field are part of the seed value.

74.7.1.3 Polynomial (GPOLY)

Offset

Register	Offset
GPOLY	4h

Function

Configures polynomial value for CRC calculation.

- Sets the upper 16 bits of polynomial that are used only in 32-bit CRC mode. Writes to this field are ignored in 16-bit CRC mode.
- Sets the lower 16 bits of polynomial that are used in both 16-bit and 32-bit CRC modes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOW															
Reset	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

Fields

Field	Function
31-16 HIGH	High Half-Word Writable and readable in 32-bit CRC mode (CTRL[TCRC] = 1). You cannot write to this field in 16-bit CRC mode (CTRL[TCRC] = 0).
15-0 LOW	Low Half-Word Writable and readable in both 16-bit and 32-bit CRC modes.

74.7.1.4 Control (CTRL)

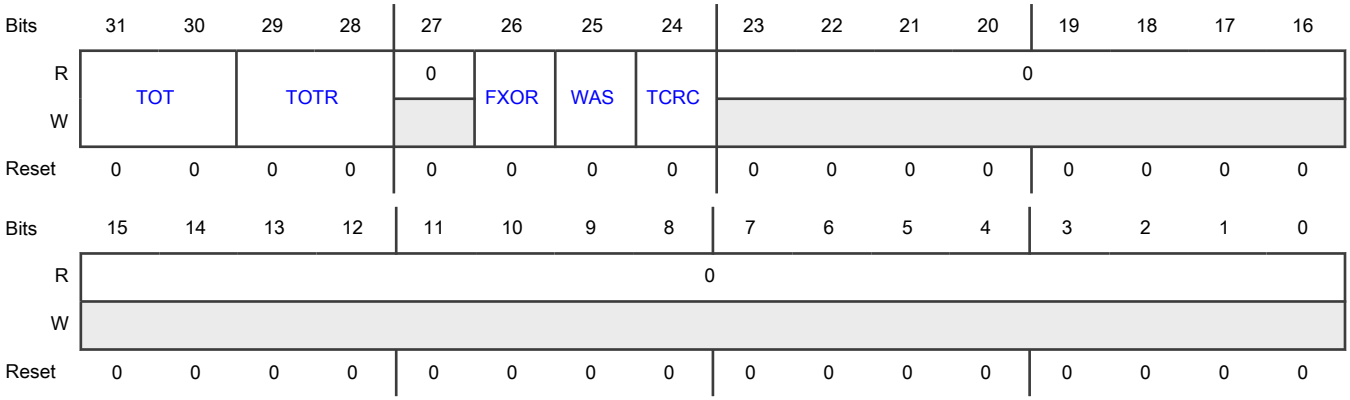
Offset

Register	Offset
CTRL	8h

Function

Sets control for CRC. You must write 1 to the appropriate fields of this register before starting a new CRC calculation, which you can initialize by writing 1 to CTRL[WAS] and then writing the seed into DATA.

Diagram



Fields

Field	Function
31-30 TOT	Transpose Type for Write Sets transpose type for the values written to DATA. See Transpose feature for the available transpose options. 00b - No transposition 01b - Bits in bytes are transposed, but bytes are not transposed. 10b - Both bits in bytes and bytes are transposed. 11b - Only bytes are transposed, no bits in a byte are transposed.
29-28 TOTR	Transpose Type for Read Sets transpose type for the values read from DATA . See Transpose feature for the available transpose options. 00b - No transposition 01b - Bits in bytes are transposed, but bytes are not transposed. 10b - Both bits in bytes and bytes are transposed. 11b - Only bytes are transposed, no bits in a byte are transposed.

Table continues on the next page...

Table continued from the previous page...

Field	Function
27 —	Reserved
26 FXOR	Complement Read of CRC Data Register Enables on-the-fly complementing of read data. Some CRC protocols require the final checksum to be XORed with FFFFFFFFh or FFFFh. 0b - Disables XOR on reading data. 1b - Inverts or complements the read value of the CRC Data.
25 WAS	Write as Seed Specifies whether writes to DATA are data values or seed values. When this field = 1, the value that you write to is considered as seed value. When this field = 0, the value that you write to is considered as data for CRC calculation. 0b - Data values 1b - Seed values
24 TCRC	TCRC Defines the width of CRC. 0b - 16 bits 1b - 32 bits
23-0 —	Reserved

Appendix A
Release notes

A.1 About This Manual changes

No substantial content changes.

A.2 Introduction changes

No substantial content changes.

A.3 Memory Map Overview changes

No substantial content changes.

A.4 Interrupts and DMA Events Assignments changes

No substantial content changes.

A.5 Fusemap changes

No substantial content changes.

A.6 External Signals and Pin Multiplexing changes

No substantial content changes.

A.7 Cortex-M33 Platform Overview changes

No substantial content changes.

A.8 External Memory Controllers Overview changes

No substantial content changes.

A.9 Boot ROM changes

- Updated the description of imageType in Table 47.
- Updated the definition of BOOT_SPEED in Table 44.
- Updated Image version and anti-rollback.

A.10 ROM API chapter changes

No substantial content changes.

A.11 ISP changes

No substantial content changes.

A.12 Power Overview changes

No substantial content changes.

A.13 Clock Overview changes

No substantial content changes.

A.14 Graphics Overview changes

No substantial content changes.

A.15 Timers Overview changes

No substantial content changes.

A.16 System Debug changes

No substantial content changes.

A.17 Audio Overview changes

No substantial content changes.

A.18 Analog Overview changes

No substantial content changes.

A.19 Security Overview changes

- Initial Draft
- replaced references to CSS and CSSV2 to ELS
- Updated descriptions to the Boot ROM Security Features in Modules.
- Updated the modules for anomaly detection and response in Modules.

A.20 IO PAD Controller (IOPCTL)

A.20.1 Chip-specific IOPCTL information changes

No substantial content changes.

A.20.2 IOPCTL module changes

No substantial content changes.

A.21 Pin Interrupt and Pattern Match (PINT)

A.21.1 Chip-specific PINT information changes

No substantial content changes.

A.21.2 PINT module changes

No substantial content changes.

A.22 General-Purpose Input/Output (GPIO)

A.22.1 Chip-specific GPIO information changes

No substantial content changes.

A.22.2 GPIO module changes

No substantial content changes.

A.23 Performance Monitor (CMX_PERFMON)

A.23.1 Chip-specific CMX_PERFMON information changes

No substantial content changes.

A.23.2 CMX_PERFMON changes

No substantial content changes.

A.24 Cache Memory Controller (XCACHE)

A.24.1 Chip-specific XCACHE information changes

No substantial content changes.

A.24.2 XCACHE module changes

No substantial content changes.

A.25 Semaphores2 (SEMA42)

A.25.1 Chip-specific SEMA42 information changes

No substantial content changes.

A.25.2 SEMA42 module changes

No substantial content changes.

A.26 Messaging Unit (MU)

A.26.1 Chip-specific MU information changes

No substantial content changes.

A.26.2 MU module changes

No substantial content changes.

A.27 Power Management Controller (PMC)

A.27.1 Chip-specific PMC information changes

No substantial content changes.

A.27.2 PMC module changes

No substantial content changes.

A.28 Clock Controller (CLKCTL)

A.28.1 Chip-specific CLKCTL information changes

No substantial content changes.

A.28.2 CLKCTL module changes

No substantial content changes.

A.29 System Controller (SYSCON)

A.29.1 Chip-specific SYSCON information changes

No substantial content changes.

A.29.2 SYSCON module changes

No substantial content changes.

A.30 Sleep Controller (SLEEPCON)

A.30.1 Chip-specific SLEEPCON information changes

No substantial content changes.

A.30.2 SLEEPCON module changes

No substantial content changes.

A.31 Reset Controller (RSTCTL)

A.31.1 Chip-specific RSTCTL information changes

No substantial content changes.

A.31.2 RSTCTL module changes

No substantial content changes.

A.32 FRO Clock Generator (FRO)

A.32.1 Chip-specific FRO information changes

No substantial content changes.

A.32.2 FRO module changes

No substantial content changes.

A.33 Frequency Measurement (FREQME)

A.33.1 Chip-specific FREQME information changes

No substantial content changes.

A.33.2 FREQME module changes

No substantial content changes.

A.34 Nano Power 32 kHz Oscillator (OSC32KNP)

A.34.1 Chip-specific OSC32KNP information changes

No substantial content changes.

A.34.2 OSC32KNP module changes

No substantial content changes.

A.35 Real Time Clock Subsystem (RTC_SS)

A.35.1 Chip-specific RTC_SS information changes

No substantial content changes.

A.35.2 RTC_SS module changes

No substantial content changes.

A.36 Real Time Clock (RTC)

A.36.1 Chip-specific RTC information changes

No substantial content changes.

A.36.2 RTC module changes

No substantial content changes.

A.37 Process, Voltage and Temperature Sensor (PVTS)

A.37.1 Chip-specific PVTS information changes

- Updated the functions of PVTS fuses with the word index of 486 and 487 in PVTS fuse programming.

A.37.2 PVTS module changes

No substantial content changes.

A.38 Analog-to-Digital Converter (ADC)

A.38.1 Chip-specific ADC information changes

No substantial content changes.

A.38.2 ADC module changes

No substantial content changes.

A.39 Comparator (ACMP)

A.39.1 Chip-specific ACMP information changes

No substantial content changes.

A.39.2 ACMP module changes

No substantial content changes.

A.40 Ultra Secured Digital Host Controller (uSDHC)

A.40.1 Chip-specific uSDHC information changes

No substantial content changes.

A.40.2 uSDHC module changes

No substantial content changes.

A.41 External Serial Peripheral Interface (XSPI)

A.41.1 Chip-specific XSPI information changes

No substantial content changes.

A.41.2 XSPI module changes

No substantial content changes.

A.42 Cache Controller (CACHE64_CTRL)

A.42.1 Chip-specific CACHE64_CTRL information changes

No substantial content changes.

A.42.2 CACHE64_CTRL module changes

• Updated Figure 162. • Changed Clocks to Clocking and Resets to Reset. • Added new topic Interrupts - "CACHE_CTRL has no interrupts".

Table continues on the next page...

• Added new topics Operations and #unique_3865. Moved the topics Cache configuration and Cache control under Operations. • Replaced "master/slave" with "controller/target" in Modes of operation and Features.
• Updated Figure 162.
• Changed Cache Control (CCR) bit 2 and 3 to read-write from read-only.
• Changed CACHE_CTRL to CACHE64_CTRL throughout the blockguide.
• Added CCR[FRCWT] and CCR[FRCNOALLC] instead of reserved.
• Changed "master" to "manager" and "slave" to "subordinate" in Cache configuration and Clocking.
• Revised the note given in Initialization.
• Deleted topic "Submodules". • Revised CLCR[LCIVB] to change "modified" bit to "valid" bit. • Revised description for CLCR[CACHEADDR].

A.43 CACHE64 Policy Select (CACHE64_POLSEL)

A.43.1 Chip-specific CACHE64_POLSEL module changes

No substantial content changes.

A.43.2 CACHE64_POLSEL module changes

No substantial content changes.

A.44 Memory Map Unit (MMU)

A.44.1 Chip-specific MMU information changes

No substantial content changes.

A.44.2 MMU module changes

No substantial content changes.

A.45 Neural Processing Unit (NPU)

A.45.1 Chip-specific NPU information changes

No substantial content changes.

A.45.2 eIQ Neutron N3-64 Neutron module changes

No substantial content changes.

A.46 EZ Handler RISC-V (EZH-V)

A.46.1 Chip-specific EZH-V information changes

No substantial content changes.

A.46.2 EZH-V module changes

No substantial content changes.

A.47 JPEG Decoder (JPEGDEC)

A.47.1 Chip-specific JPEGDEC information changes

No substantial content changes.

A.47.2 JPEGDEC module changes

No substantial content changes.

A.48 JPEG Decoder Wrapper (JPGDECWRP)

A.48.1 Chip-specific JPGDECWRP information changes

No substantial content changes.

A.48.2 JPGDECWRP module changes

No substantial content changes.

A.49 PNG Decoder (PNGDEC)

A.49.1 Chip-specific PNGDEC information changes

No substantial content changes.

A.49.2 PNGDEC module changes

No substantial content changes.

A.50 Vector Graphics Processing Unit (VGPU)

A.50.1 Chip-specific VGPU information changes

No substantial content changes.

A.50.2 VGPU module changes

No substantial content changes.

A.51 Universal Serial Bus Controller (USB)

A.51.1 Chip-specific USB information changes

No substantial content changes.

A.51.2 USB module changes

No substantial content changes.

A.52 Universal Serial Bus 2.0 PHY (USBPHY)

A.52.1 Chip-specific USBPHY information changes

No substantial content changes.

A.52.2 USBPHY module changes

No substantial content changes.

A.53 USB Device Charger Detection Module (USBDCD)

A.53.1 Chip-specific USBDCD information changes

No substantial content changes.

A.53.2 USBDCD module changes

No substantial content changes.

A.54 Mobile Industry Processor Interface and Display Serial Interface Controller (MIPI_DSI_HOST)

A.54.1 Chip-specific MIPI_DSI_HOST information changes

No substantial content changes.

A.54.2 MIPI_DSI_HOST module changes

No substantial content changes.

A.55 LCD Interface (LCDIF)

A.55.1 Chip-specific LCDIF information changes

No substantial content changes.

A.55.2 LCDIF module changes

No substantial content changes.

A.56 Sigma Delta Analog-to-Digital Converter (SDADC)

A.56.1 Chip-specific SDADC information changes

No substantial content changes.

A.56.2 SDADC module changes

No substantial content changes.

A.57 PDM Microphone Interface (MICFIL)

A.57.1 Chip-specific MICFIL information changes

No substantial content changes.

A.57.2 MICFIL module changes

No substantial content changes.

A.58 Synchronous Audio Interface (SAI)

A.58.1 Chip-specific SAI information changes

No substantial content changes.

A.58.2 SAI module changes

No substantial content changes.

A.59 HiFi 1 DSP Platform (HiFi1)

A.59.1 Chip-specific HiFi1 information changes

No substantial content changes.

A.59.2 HiFi1 module changes

No substantial content changes.

A.60 HiFi 4 DSP Platform (HiFi4)

A.60.1 Chip-specific HiFi4 information changes

No substantial content changes.

A.60.2 HiFi4 module changes

No substantial content changes.

A.61 Micro-Tick Timer (UTICK)

A.61.1 Chip-specific UTICK information changes

No substantial content changes.

A.61.2 UTICK module changes

No substantial content changes.

A.62 Standard Counter/Timers (CTIMER)

A.62.1 Chip-specific CTIMER information changes

No substantial content changes.

A.62.2 CTIMER module changes

No substantial content changes.

A.63 OS Event Timer (OSTIMER)

A.63.1 Chip-specific OSTIMER information changes

No substantial content changes.

A.63.2 OSTIMER module changes

No substantial content changes.

A.64 SCTimer (SCT)

A.64.1 Chip-specific SCT information changes

No substantial content changes.

A.64.2 SCT module changes

No substantial content changes.

A.65 Multi-Rate Timer (MRT)

A.65.1 Chip-specific MRT information changes

No substantial content changes.

A.65.2 MRT module changes

No substantial content changes.

A.66 Windowed Watchdog Timer (WWDT)

A.66.1 Chip-specific WWDT information changes

No substantial content changes.

A.66.2 WWDT module changes

No substantial content changes.

A.67 Flexible I/O (FLEXIO)

A.67.1 Chip-specific FLEXIO information changes

No substantial content changes.

A.67.2 FLEXIO module changes

No substantial content changes.

A.68 Low-Power Flexible Communications Interface (LP_FLEXCOMM)

A.68.1 Chip-specific LP_FLEXCOMM information changes

No substantial content changes.

A.68.2 LP_FLEXCOMM module changes

No substantial content changes.

A.68.3 LPI2C module changes

No substantial content changes.

A.68.4 LPSPI module changes

No substantial content changes.

A.68.5 LPUART module changes

No substantial content changes.

A.69 Improved Inter-Integrated Circuit (I3C)

A.69.1 Chip-specific I3C module changes

No substantial content changes.

A.69.2 I3C module changes

No substantial content changes.

A.70 Input Multiplexing (INPUTMUX)

A.70.1 Chip-specific INPUTMUX information changes

No substantial content changes.

A.70.2 INPUTMUX module changes

No substantial content changes.

A.71 Enhanced Direct Memory Access (eDMA)

A.71.1 Chip-specific eDMA information changes

No substantial content changes.

A.71.2 eDMA3 module changes

No substantial content changes.

A.72 Network Interconnect Component (NIC)

A.72.1 Chip-specific NIC information changes

No substantial content changes.

A.72.2 NIC module changes

No substantial content changes.

A.73 On-Chip OTP Controller (OCOTP_CTRL)

A.73.1 Chip-specific OCOTP_CTRL information changes

No substantial content changes.

A.73.2 OCOTP module changes

- Updated OTP_SHADOW registers.
- Added HW_OCOTP_CLK_DIV register.
- Updated Clocking.

A.74 Cyclic Redundancy Check (CRC)

A.74.1 Chip-specific CRC information changes

No substantial content changes.

A.74.2 CRC module changes

No substantial content changes.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

EdgeLock — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2025 NXP B.V. All rights reserved.

For more information, please visit: https://www.nxp.com

Date of release: 31 October 2025
Document identifier: IMXRT700RM