

Introduction

The PIC32CX-BZ2 family is a general purpose, low-cost, 32-bit Microcontroller (MCU) family of devices supporting multi-protocol wireless interfaces (Bluetooth® and Zigbee®), hardware-based security accelerator, transceiver and Power Management Unit (PMU).

The WBZ45 Module is a series of fully RF-certified wireless modules that contain the PIC32CX-BZ2 wireless MCUs with the following antenna options:

- PCB Antenna
- u.FL Connector for External Antenna

In addition to the Bluetooth® Low Energy (Bluetooth 5.2) and Zigbee (Zigbee 3.0) wireless protocol support, the PIC32CX-BZ2 devices and the WBZ45 modules also support a rich set of standard MCU peripherals, such as Analog-to-Digital Converter (ADC), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I²C), Quad I/O Serial Peripheral Interface (QSPI), Universal Asynchronous Receiver-Transmitter (UART) and Serial Wire Debug (SWD).

PIC32CX-BZ2 SoC Family Features

The following section lists the PIC32CX-BZ2 SoC related features:

Operating Conditions of MCUs

- 1.9V to 3.6V, -40°C to +125°C, 64 MHz
 - AEC Q100 Grade 1 qualified

Core: 64 MHz ARM® Cortex® -M4

- 2.42 Coremark®/MHz
- 4 KB Combined Instruction Cache and Data Cache
- 8-Zone Memory Protection Unit (MPU)
- Thumb®-2 Instruction Set
- Digital Signal Processing ASE Rev 2
- Nested Vector Interrupt Controller (NVIC)
- Embedded Trace Module (ETM) with Instruction Trace Stream
- Core Sight Embedded Trace Buffer (ETB)
- Trace Port Interface Unit (TPIU)
- IEEE® 754-Compliant Floating Point Unit (FPU)

Memories

- 1 MB On-Chip Self-Programmable Flash with:
 - Error Correction Code (ECC)
 - Prefetch module to speed up Flash accesses
 - 20-years of data retention support
- 32 KB NVR Flash (8 Sectors)

- Private/public boot code and data
- NV calibration data
- 128 KB Multi-Port Programmable QoS SRAM Main Memory
 - 64 KB of Error Correction Code (ECC) RAM option
- Up to 4 KB of Tightly Coupled Memory (TCM)
- Up to 8 KB Backup SRAM
- Single 32-bit Backup Register

System

- Power-on Reset (POR) and Brown-out Detect (BOD)
- Internal and External Clock Options
- External Interrupt Controller (EIC)
- Up to four External Interrupts
- One Non-maskable Interrupt
- 2-pin Serial Wire Debug (SWD) Programming, Test and Debugging Interface

Supported Connectivity Standards

- Complies with:
 - Bluetooth v5.2
 - IEEE 802.15.4, Zigbee 3.0
 - ETSI EN 300 328 and EN 300 440 Class 2
 - FCC CFR47 Part 15 and ARIB STD-T66

Power Supply

- Integrated PMU with:
 - Buck (DC-DC/switching) mode; supports High Power (PWM) and Low Power (PSK) mode
 - MLDO (linear) mode
- On-board 1.2V Voltage Regulator (CLDO)
- Power-on Reset (POR) and Brown-out Detect (BOD) on 3.3V and 1.2V Rails
- Run, Idle, Dream, Sleep, Deep Sleep and Extreme Deep Sleep Modes
- Sleep Walking Peripherals
- Embedded Buck/LDO Regulator Supporting On-the-Fly (OTF) Selection

2.4 GHz RF Transceiver

- Integrated 2.4 GHz Ultra-low Power RF Transceiver shared among Bluetooth, Zigbee Modems, Link (MAC) Controllers and Proprietary Modulation Schemes
- Integrated 16 MHz ± 20 ppm Crystal Oscillator (External Low Cost Crystal)
- Two PA Design Architecture (LPA (+5 dBm) and MPA (+12 dBm)) to Improve TX Power Efficiency
- Low RBOM Two-port TRX RFFE Architecture
 - Integrated balun (single-ended RF output) and TRX switch
- Hardware Radio Arbiter with Programmable QoS:
 - Resolution per packet level
 - Time-division coexistence between Bluetooth and 802.15.4
 - Based on shared transceiver and antenna

- Maintains connections of 802.15.4 and Bluetooth simultaneously

Bluetooth

- Bluetooth Low Energy 5.2 Certified
- Up to +12 dBm Programmable Transmit Output Power
- Typical Receiver Power Sensitivity:
 - -97 dBm for Bluetooth Low Energy 1 Mbps
 - -94 dBm for Bluetooth Low Energy 2 Mbps
 - -104 dBm for Bluetooth Low Energy 125 Kbps
 - -101 dBm for Bluetooth Low Energy 500 Kbps
 - Digital RSSI indicator (-50 dBm to -90 dBm)
- Bluetooth Supported Features:
 - 2M uncoded PHY
 - Long range (Coded PHY)
 - Channel selection algorithm #2
 - Advertising extensions, offloads CPU with hardware-based scheduler
 - High duty cycle non-connectible advertising
 - Data length extensions
 - Secure connections
 - Privacy upgrades (with hardware white-list support)
- ECDH P256 Hardware Engine for Link Key Generation when Bluetooth Pairing
- AES128 Hardware Module for Real-Time Bluetooth Payload Data Encryption
- Preprogrammed MAC Address
- Bluetooth Qualification Test Facility (BQTF) Certification
- Bluetooth Low Energy Profiles:
 - Bluetooth Low Energy peripheral and central roles
 - Bluetooth Low Energy APIs for application layer to implement standard or customize GATT based profiles/services
 - Microchip Transparent UART Service
 - Battery Service
 - Device Information Service
 - Custom Service
 - Multi-link and multi-role
- Bluetooth Low Energy Services:
 - Provisioning
 - Over-the-Air (OTA) update (also known as DFU)
 - Advertisement/Beacon
 - Personalized configuration
 - Alert notification service

802.15.4/Zigbee Modulation Scheme

- 802.15.4/Zigbee Physical Layer Service Unit (PSDU) Data Rate: 250 Kbps
- Programmable RX Mode

- -100 dBm RX sensitivity in the Continuous mode
- -98 dBm sensitivity in RPC mode
- RPC mode provides lower power consumption in RX mode to support California Green Energy Specification at the system level
- TX Output Power up to +12 dBm
- Hardware Assisted MAC
 - Auto acknowledge
 - Auto retry
 - Channel access back-off
- Preprogrammed MAC Address
- SFD Detection; Spreading; De-spreading; Framing; CRC-16 Computation
- Independent TX/RX Buffers for Improved CPU Offloading While Handling Zigbee Data
 - 128-byte TX and 128-byte RX frame buffer
- Hardware Security
 - Advanced Encryption Standard (AES)
 - True Random Number Generator (TRNG)
- Zigbee Stack Support
 - Zigbee 3.0 ready
 - Zigbee Pro 2017
 - Zigbee green power support (proxy, sink and multi-sensor)

Proprietary

- Proprietary Data Rates: 500 Kbps, 1 Mbps and 2 Mbps
- TX Output Power up to +12 dBm
- Receiver Sensitivity up to -96 dBm
- Hardware Assisted MAC
 - Auto acknowledge
 - Auto retry
 - Channel access back-off
- SFD Detection; Spreading; De-spreading; Framing; CRC-16 Computation
- Independent TX/RX Buffers for Improved CPU Offloading While Handling Zigbee Data
 - 128-byte TX and 128-byte RX frame buffer
- Hardware Security
 - Advanced Encryption Standard (AES)
 - True Random Number Generator (TRNG)

High Performance Peripherals

- 16-Channel Direct Memory Access Controller (DMAC)
 - Built-in CRC with memory CRC generation/monitor hardware support
- One Quad I/O Serial Peripheral Interface (QSPI)
 - execute-In-Place (xIP) support
 - Dedicated AHB memory zone

System Peripherals

- 32-Channel Event System
 - All channels can be connected to any event generator
 - All channels provide a pure asynchronous path
 - Twelve channels support synchronous and re-synchronous
- Four Serial Communication Interfaces (SERCOM), Each Configurable to Operate as:
 - USART with full-duplex and single-wire half-duplex configuration
 - ISO7816
 - I²C up to 1 MHz (three SERCOMs support I²C)
 - LIN Commander/Responder
 - RS485
 - SPI with configurable inter-byte space
- Four 16-bit Timers/Counters (TC), Each Configurable as:
 - 16-bit TC with two compare/capture channels
 - 8-bit TC with two compare/capture channels
 - 32-bit TC with two compare/capture channels
- Two 24-bit Timer/Counters for Control (TCC) with Extended Functions:
 - Up to six compare channels with optional complementary output
 - Generation of synchronized Pulse Width Modulation (PWM) pattern across port pins
 - Deterministic fault protection, fast decay and configurable dead-time between complementary output
 - Dithering that increases resolution with up to 5 bits and reduces quantization error
- One 16-bit Timer/Counters for Control (TCC) with Extended Functions:
 - Up to two compare channels with optional complementary output
- 32-bit Real Time Counter (RTCC) with Clock/Calendar Function
- Up to one Wake-up Pin with Tamper Detection and Debouncing Filter
- Watchdog Timer (WDT) with Window Mode
- Deadman Timer (DMT)
- CRC-32 Generator
- Frequency Meter (FREQM)
- One Configurable Custom Logic (CCL)
- One Analog Comparator (AC) with Window Compare function
- One Temperature Sensor (Die Temperature)

Advanced Analog

- 12-bit ADC SAR Module (ADC):
 - Up to Eight Analog Channels
 - Up to Two MSa/s conversion rate
 - Multiple trigger sources
 - Supports die temperature sensor built into RF-Analog (not an external “ambient” temperature sensor)
 - Two Analog Comparator (AC) with Window Compare Function or single Analog Comparator
 - One dedicated AC and second AC is shared with PMU Controller

Security

- AES Engine with Support for 128/192/256-bit Cryptographic Key
- One AES with 256-bit Key Length and up to 2 MB/s Data Rate
 - Five confidential modes of operation (ECB, CBC, CFB, OFB and CTR)
 - Supports counter with CBC-MAC mode
 - Galois Counter Mode (GCM)
- True Random Number Generator (TRNG)
- Public Key Cryptography Controller (PUKCC) and Associated Classical Public Key Cryptography Library (PUKCL)
 - RSA and DSA algorithm
 - Elliptic Curves Cryptography (ECC), ECC GF (2n) and ECCGF (p)
- Integrity Check Module (ICM) Based on Secure Hash Algorithm (SHA1, SHA224 and SHA256), DMA Assisted

Oscillators

- 16 MHz, ± 20 PPM Crystal/Resonator Oscillator or External Clock (POSC) for 2.4G RF Transceiver
- Shared System PLL with Bluetooth/Zigbee RF Data Converter PLL
- 32.768 kHz Ultra-low Power Internal Oscillator
- Higher Accuracy 32.768 kHz, ± 250 ppm Clock Options
 - POSC derived 32 kHz clock
 - 32.768 kHz crystal/resonator oscillator (SOSC)
 - External 32.768 kHz clock source
- 8 MHz Internal RC Oscillator (FRC)

I/O

- Flexible Peripheral Pin Select (PPS) Support
- High-Current Sink/Source on Most I/O Pins
- Configurable Open-Drain Output on Digital I/O Pins
- Up to 27 Programmable I/O Pins

Package

- PIC32CX1012BZ25048 Package:
 - 48-pin QFN
 - Size – 7 mm x 7 mm x 0.9 mm
- PIC32CX1012BZ24032 Package:
 - 32-pin QFN
 - Size - 5 mm x 5 mm x 1 mm

WBZ45 Module Features

The WBZ45 modules are wireless MCU modules with BLE 5.2 compliant and Zigbee 3.0 Radio.

The following section lists the WBZ45 Module related features, which complement SoC features:

WBZ45 Module Variants

- WBZ451 Module based on (PIC32CX1012BZ25048 SoC)
 - WBZ451PE (PCB antenna)

- WBZ451UE (u.FL)
- WBZ450 Module based on (PIC32CX1012BZ24032 SoC)
 - WBZ450PE (PCB antenna)
 - WBZ450UE (u.FL)
- WBZ451H Module based on (PIC32CX1012BZ25048 SoC)
 - WBZ451HPE (PCB antenna)
 - WBZ451HUE (u.FL)

Antenna

- On-Board PCB Antenna
- External Antenna

Clock Management

- Integrated 16 MHz POSC

System Peripheral, Advanced Analog and Security

- All features of SoC are accessible

Package and Operating Conditions

- WBZ451
 - 39-pin SMD package with Shield CAN
 - Size – 15.5 mm x 20.7 mm x 2.8 mm
- WBZ450
 - 30-pin SMD package with Shield CAN
 - Size – 13.4 mm x 18.7 mm x 2.8 mm
- WBZ451H
 - 39-pin SMD package with Shield CAN
 - Size – 15.5 mm x 26.8 mm x 2.8 mm
- Operating Conditions
 - 1.9V to 3.6V, -40°C to +85°C, DC to 64 MHz

Certifications

- WBZ451 Module Certified to FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC Radio Regulations
- WBZ450 Module Certified to FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC Radio Regulations
- WBZ451H Module Certified to FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC Radio Regulations
- RoHS and REACH Compliant

Table of Contents

Introduction.....	1
PIC32CX-BZ2 SoC Family Features.....	1
Operating Conditions of MCUs.....	1
Core: 64 MHz ARM® Cortex®-M4.....	1
Memories.....	1
System.....	2
Supported Connectivity Standards.....	2
Power Supply.....	2
2.4 GHz RF Transceiver.....	2
Bluetooth.....	3
802.15.4/Zigbee Modulation Scheme.....	3
Proprietary.....	4
High Performance Peripherals.....	4
System Peripherals.....	5
Advanced Analog.....	5
Security.....	6
Oscillators.....	6
I/O.....	6
Package.....	6
WBZ45 Module Features.....	6
WBZ45 Module Variants.....	6
Antenna.....	7
Clock Management.....	7
System Peripheral, Advanced Analog and Security.....	7
Package and Operating Conditions.....	7
Certifications.....	7
1. Ordering Information.....	18
1.1. PIC32CX-BZ2 SoC Ordering Information.....	18
1.2. WBZ45 Module Ordering Information.....	18
2. Configuration Summary.....	20
3. PIC32CX-BZ2 SoC Description.....	21
3.1. PIC32CX-BZ2 SoC Block Diagram.....	21
3.2. Pinout Diagram.....	21
4. WBZ45 Module Description.....	24
4.1. Pinout Diagram.....	25
4.2. Basic Connection Requirement.....	27
4.3. WBZ45 Module Placement Guidelines.....	32
4.4. WBZ45 Module Routing Guidelines.....	33
4.5. WBZ45 Module RF Considerations.....	34
4.6. WBZ45 Module Antenna Considerations.....	34
4.7. WBZ45 Module Reflow Profile Information.....	41
4.8. WBZ45 Module Assembly Considerations.....	41

5. Pinout and Signal Descriptions List.....	42
6. I/O Ports and Peripheral Pin Select (PPS).....	45
6.1. Control Registers.....	46
6.2. Peripheral Pin Select (PPS).....	49
6.3. Function Priority for Device Pins.....	62
6.4. I/O Ports Control Registers.....	67
6.5. Operation in Power Saving Modes.....	82
6.6. Results of Various Resets.....	83
7. Power Subsystem.....	84
7.1. Block Diagram.....	84
7.2. Voltage Regulators.....	85
7.3. Power Supply Modes.....	85
7.4. Typical Power Supply Connection for SoC.....	86
7.5. Typical Power Supply Connection for Module.....	86
7.6. Power-Up Sequence.....	87
8. Product Memory Mapping Overview.....	88
9. Prefetch Cache (PCHE).....	91
9.1. Overview.....	91
9.2. Features.....	91
9.3. Overview.....	91
9.4. Prefetch Behavior.....	93
9.5. Configurations.....	93
9.6. Predictive Prefetch Behavior.....	94
9.7. Coherency Support.....	94
9.8. Effects of Reset.....	94
9.9. Error Conditions.....	95
9.10. Operation in Power-saving Modes.....	96
9.11. Register Summary (PCHE).....	97
9.12. Register Description.....	97
10. Processor and Architecture.....	104
10.1. Cortex M4 Processor.....	104
10.2. Nested Vector Interrupt Controller (NVIC).....	106
10.3. High-Speed Bus System.....	110
11. Cortex M Cache Controller (CMCC).....	113
11.1. Overview.....	113
11.2. Features.....	113
11.3. Block Diagram.....	114
11.4. Signal Description.....	115
11.5. Product Dependencies.....	115
11.6. Functional Description.....	116
11.7. DEBUG Mode.....	118
11.8. RAM Properties.....	119
11.9. Register Summary.....	120
11.10. Register Description.....	120

12. Device Service Unit (DSU).....	133
12.1. Overview.....	133
12.2. Features.....	133
12.3. Block Diagram.....	133
12.4. Signal Description.....	133
12.5. Product Dependencies.....	134
12.6. Debug Operation.....	135
12.7. Chip Erase.....	136
12.8. Programming.....	137
12.9. Intellectual Property Protection.....	137
12.10. Device Identification.....	138
12.11. Functional Description.....	139
12.12. DSU Register Summary.....	142
12.13. Register Description.....	143
13. Clock and Reset Unit (CRU).....	164
13.1. Overview.....	164
13.2. Features.....	164
13.3. Block Diagram.....	165
13.4. System and Peripheral Clock Generation (CLKGEN).....	168
13.5. Idle Mode.....	168
13.6. Dream Mode.....	168
13.7. FRCDIV.....	169
13.8. RFPLL Wrapper.....	169
13.9. Start-up Considerations.....	169
13.10. Fail-Safe Clock Monitor.....	169
13.11. Fast RC Oscillator.....	169
13.12. Secondary Oscillator.....	170
13.13. Low Power RC Oscillator (LPRC).....	170
13.14. Reference Clock Generator.....	170
13.15. Resets.....	171
13.16. Register Summary.....	180
13.17. Register Description.....	182
14. RAM Error Correction Code (RAMECC).....	206
14.1. Overview.....	206
14.2. Features.....	206
14.3. Block Diagram.....	206
14.4. Signal Description.....	206
14.5. Product Dependencies.....	206
14.6. Functional Description.....	207
14.7. Register Summary - RAMECC.....	209
14.8. Register Description.....	209
15. Power Management Unit (PMU).....	216
15.1. Overview.....	216
15.2. Power Modes.....	216
15.3. PMU Register Summary.....	218
15.4. Deep Sleep Control Register	219

16. Watchdog Timer (WDT).....	221
16.1. Overview.....	221
16.2. Features.....	221
16.3. Applications.....	221
16.4. Block Diagram.....	223
16.5. Configuration.....	223
16.6. Register Summary - WDT.....	224
16.7. Register Description.....	224
17. Deadman Timer (DMT).....	226
17.1. Overview.....	226
17.2. Features.....	226
17.3. Block Diagram.....	226
17.4. DMT Operation.....	227
17.5. Register Summary.....	230
17.6. Register Description.....	231
18. System Configuration and Register Locking (CFG).....	240
18.1. Overview.....	240
18.2. Applications.....	241
18.3. CFG Register Summary.....	243
18.4. Register Description.....	244
19. Register Locking.....	278
19.1. System Lock Register.....	278
19.2. Register Summary.....	280
19.3. Register Description.....	280
20. Peripheral Module Disable Register (PMD).....	282
20.1. Overview.....	282
20.2. Enabling Peripherals.....	282
20.3. Registers and Bits.....	282
20.4. PMD Register.....	282
20.5. PMDx Initialization Values by Variant Name.....	285
21. Real-Time Counter and Calendar (RTCC).....	287
21.1. Overview.....	287
21.2. Features.....	287
21.3. Block Diagram.....	287
21.4. Signal Description.....	288
21.5. Product Dependencies.....	289
21.6. Functional Description.....	290
21.7. Register Summary - Mode 0 - 32-Bit Counter.....	301
21.8. Register Description - Mode 0 - 32-Bit Counter.....	302
21.9. Register Summary - Mode 1 - 16-Bit Counter.....	321
21.10. Register Description - Mode 1 - 16-Bit Counter.....	322
21.11. Register Summary - Mode 2 - Clock/Calendar.....	342
21.12. Register Description - Mode 2 - Clock/Calendar.....	343
22. Direct Memory Access Controller (DMAC).....	362

22.1. Overview.....	362
22.2. Features.....	362
22.3. Block Diagram.....	364
22.4. Signal Description.....	364
22.5. Product Dependencies.....	364
22.6. Functional Description.....	365
22.7. DMAC Register Summary.....	390
22.8. Register Description.....	394
22.9. DMAC Register Summary (SRAM).....	423
22.10. Register Description - SRAM.....	423
23. External Interrupt Controller (EIC).....	430
23.1. Overview.....	430
23.2. Features.....	430
23.3. Block Diagram.....	430
23.4. Signal Description.....	430
23.5. Product Dependencies.....	431
23.6. Functional Description.....	432
23.7. EIC Register Summary.....	438
23.8. Register Description.....	438
24. Flash Memory.....	453
24.1. Overview.....	453
24.2. Features.....	453
24.3. Functional Block Diagram.....	454
24.4. Flash Memory Addressing.....	454
24.5. Memory Configuration.....	454
24.6. Boot Flash Memory (BFM) Partitions.....	455
24.7. Program Flash Memory (PFM) Partitions.....	456
24.8. Error Correcting Code (ECC) and Flash Programming.....	456
24.9. Interrupts.....	457
24.10. Error Detection	457
24.11. NVMKEY Register Unlocking Sequence.....	458
24.12. Word Programming.....	459
24.13. Quad Word Programming.....	460
24.14. Row Programming.....	461
24.15. Page Erase.....	462
24.16. Program Flash Memory (PFM) Erase.....	464
24.17. Pre-Program.....	465
24.18. Device Code Protection bit (CP).....	465
24.19. Operation in Power-Saving Modes.....	465
24.20. Operation in Debug Mode.....	465
24.21. Effects of Various Resets.....	465
24.22. Control Registers.....	465
25. Integrity Check Monitor (ICM).....	484
25.1. Overview.....	484
25.2. Features.....	484
25.3. Block Diagram.....	485

25.4. Signal Description.....	485
25.5. Product Dependencies.....	485
25.6. Functional Description.....	486
25.7. Register Summary - ICM.....	499
25.8. Register Description.....	500
26. Peripheral Access Controller (PAC).....	518
26.1. Overview.....	518
26.2. Features.....	518
26.3. Block Diagram.....	518
26.4. Product Dependencies.....	518
26.5. Functional Description.....	519
26.6. Register Summary.....	523
26.7. Register Description.....	523
27. Frequency Meter (FREQM).....	539
27.1. Overview.....	539
27.2. Features.....	539
27.3. Block Diagram.....	539
27.4. Signal Description.....	539
27.5. Product Dependencies.....	539
27.6. Functional Description.....	540
27.7. Register Summary - FREQM.....	543
27.8. Register Description.....	543
28. Event System (EVSYS).....	553
28.1. Overview.....	553
28.2. Features.....	553
28.3. Block Diagram.....	553
28.4. Product Dependencies.....	553
28.5. Functional Description.....	555
28.6. Register Summary.....	562
28.7. Register Description.....	566
29. Serial Communication Interface (SERCOM).....	582
29.1. Overview.....	582
29.2. Features.....	582
29.3. Block Diagram.....	582
29.4. Signal Description.....	583
29.5. Product Dependencies.....	583
29.6. Functional Description.....	584
30. SERCOM Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART).....	589
30.1. Overview.....	589
30.2. USART Features.....	589
30.3. Block Diagram.....	590
30.4. Signal Description.....	590
30.5. Product Dependencies.....	590
30.6. Functional Description.....	592
30.7. Register Summary.....	606

30.8. Register Description.....	606
31. SERCOM Serial Peripheral Interface (SERCOM SPI).....	625
31.1. Overview.....	625
31.2. Features.....	625
31.3. Block Diagram.....	626
31.4. Signal Description.....	626
31.5. Product Dependencies.....	626
31.6. Functional Description.....	627
31.7. Register Summary.....	638
31.8. Register Description.....	638
32. SERCOM Inter-Integrated Circuit (SERCOM I ² C).....	654
32.1. Overview.....	654
32.2. Features.....	654
32.3. Block Diagram.....	655
32.4. Signal Description.....	655
32.5. Product Dependencies.....	655
32.6. Functional Description.....	657
32.7. Register Summary - I ² C Client.....	674
32.8. Register Description - I ² C Client.....	674
32.9. Register Summary - I ² C Host.....	689
32.10. Register Description - I ² C Host.....	689
33. Quad Serial Peripheral Interface (QSPI).....	707
33.1. Overview.....	707
33.2. Features.....	707
33.3. Block Diagram.....	708
33.4. Signal Description.....	708
33.5. Product Dependencies.....	708
33.6. Functional Description.....	710
33.7. Register Summary.....	726
33.8. Register Description.....	727
34. Configurable Custom Logic (CCL).....	748
34.1. Overview.....	748
34.2. Features.....	748
34.3. Block Diagram.....	748
34.4. Signal Description.....	749
34.5. Product Dependencies.....	749
34.6. Functional Description.....	750
34.7. Register Summary.....	759
34.8. Register Description.....	759
35. True Random Number Generator (TRNG).....	764
35.1. Overview.....	764
35.2. Features.....	764
35.3. Block Diagram.....	764
35.4. Signal Description.....	764
35.5. Product Dependencies.....	764

35.6. Functional Description.....	765
35.7. Register Summary.....	767
35.8. Register Description.....	767
36. Advanced Encryption Standard (AES).....	774
36.1. Overview.....	774
36.2. Features.....	774
36.3. Block Diagram.....	775
36.4. Signal Description.....	776
36.5. Product Dependencies.....	776
36.6. Functional Description.....	777
36.7. Register Summary.....	785
36.8. Register Description.....	787
37. Public Key Cryptography Controller (PUKCC).....	803
37.1. Overview.....	803
37.2. Product Dependencies.....	803
37.3. Functional Description.....	804
38. Analog-to-Digital Converter (ADC).....	915
38.1. Overview.....	915
38.2. ADC Operation.....	916
38.3. ADC Module Configuration.....	919
38.4. Additional ADC Functions.....	929
38.5. Interrupts.....	935
38.6. Power-Saving Modes of Operation.....	937
38.7. Effects of Reset.....	939
38.8. Transfer Function.....	939
38.9. ADC Sampling Requirements.....	939
38.10. Connection Considerations.....	940
38.11. Register Description.....	941
39. Analog Comparators (AC).....	985
39.1. Overview.....	985
39.2. Features.....	985
39.3. Block Diagram.....	986
39.4. Product Dependencies.....	986
39.5. Functional Description.....	988
39.6. Register Summary.....	998
39.7. Register Description.....	998
40. Timer/Counter (TC).....	1014
40.1. Overview.....	1014
40.2. Features.....	1014
40.3. Block Diagram.....	1015
40.4. Signal Description.....	1015
40.5. Product Dependencies.....	1016
40.6. Functional Description.....	1017
40.7. Register Description.....	1033

41. Timer/Counter for Control Applications (TCC).....	1093
41.1. Overview.....	1093
41.2. Features.....	1093
41.3. Block Diagram.....	1094
41.4. Signal Description.....	1094
41.5. Product Dependencies.....	1094
41.6. Functional Description.....	1096
41.7. Register Summary.....	1126
41.8. Register Description.....	1127
42. Zigbee Bluetooth Radio Subsystem (ZBT).....	1160
42.1. Overview.....	1160
42.2. Features.....	1160
42.3. Wireless Subsystem Top Level Diagram.....	1163
42.4. Bluetooth Link Controller	1164
42.5. Zigbee/Proprietary Data Rate Link Controller.....	1165
42.6. Radio Arbiter.....	1166
42.7. RF Physical Layer	1166
42.8. Frequency Synthesizer	1166
42.9. RFLDO.....	1167
43. Electrical Characteristics.....	1168
43.1. Absolute Maximum Electrical Characteristics.....	1168
43.2. DC Electrical Characteristics.....	1168
43.3. Thermal Specifications.....	1168
43.4. Power Supply DC Module Electrical Specifications.....	1169
43.5. Active Current Consumption DC Electrical Specifications (85°C).....	1171
43.6. Active Current Consumption DC Electrical Specifications (125°C).....	1172
43.7. Idle Current Consumption DC Electrical Specifications (85°C).....	1174
43.8. Idle Current Consumption DC Electrical Specifications (125°C).....	1174
43.9. Sleep Current Consumption DC Electrical Specifications (85°C).....	1176
43.10. Sleep Current Consumption DC Electrical Specifications (125°C).....	1177
43.11. Deep Sleep Current Consumption DC Electrical Specifications (85°C).....	1180
43.12. Deep Sleep Current Consumption DC Electrical Specifications (125°C).....	1181
43.13. XDS (Extreme Deep Sleep) Current Consumption DC Electrical Specifications (85°C).....	1182
43.14. XDS (Extreme Deep Sleep) Current Consumption DC Electrical Specifications (125°C).....	1183
43.15. I/O PIN AC/DC Electrical Specifications.....	1184
43.16. External XTAL and Clock AC Electrical Specifications.....	1185
43.17. XOSC32 AC Electrical Specifications.....	1187
43.18. Low Power Internal 32 kHz RC Oscillator AC Electrical Specifications.....	1189
43.19. FRC AC Electrical Specifications.....	1189
43.20. Frequency AC Electrical Specifications.....	1190
43.21. ADC Electrical Specifications.....	1191
43.22. Comparator AC Electrical Specifications.....	1194
43.23. SPI Module Electrical Specifications.....	1195
43.24. UART AC Electrical Specifications.....	1198
43.25. I ² C Module Electrical Specifications.....	1199
43.26. QSPI Module Electrical Specifications.....	1203
43.27. TCx Timer Capture Module AC Electrical Specifications.....	1205

43.28. TCCx Timer Capture Module AC Electrical Specifications.....	1206
43.29. FLASH NVM AC Electrical Specifications.....	1207
43.30. Frequency Meter AC Electrical Specifications.....	1207
43.31. Bluetooth Low Energy RF Characteristics.....	1208
43.32. Zigbee RF Characteristics.....	1220
44. Packaging Information.....	1231
44.1. PIC32CX1012BZ25048 SoC Packaging Information.....	1232
44.2. PIC32CX1012BZ24032 SoC Packaging Information.....	1235
44.3. WBZ451 Module Packaging Information.....	1238
44.4. WBZ450 Module Packaging Information	1241
45. Appendix A: Regulatory Approval.....	1244
45.1. United States.....	1244
45.2. Canada.....	1245
45.3. Europe.....	1247
46. Document Revision History.....	1249
Microchip Information.....	1250
The Microchip Website.....	1250
Product Change Notification Service.....	1250
Customer Support.....	1250
Microchip Devices Code Protection Feature.....	1250
Legal Notice.....	1250
Trademarks.....	1251
Quality Management System.....	1252
Worldwide Sales and Service.....	1253

1. Ordering Information

This chapter provides the ordering information of the PIC32CX-BZ2 SoC and the WBZ45 Module.

1.1 PIC32CX-BZ2 SoC Ordering Information

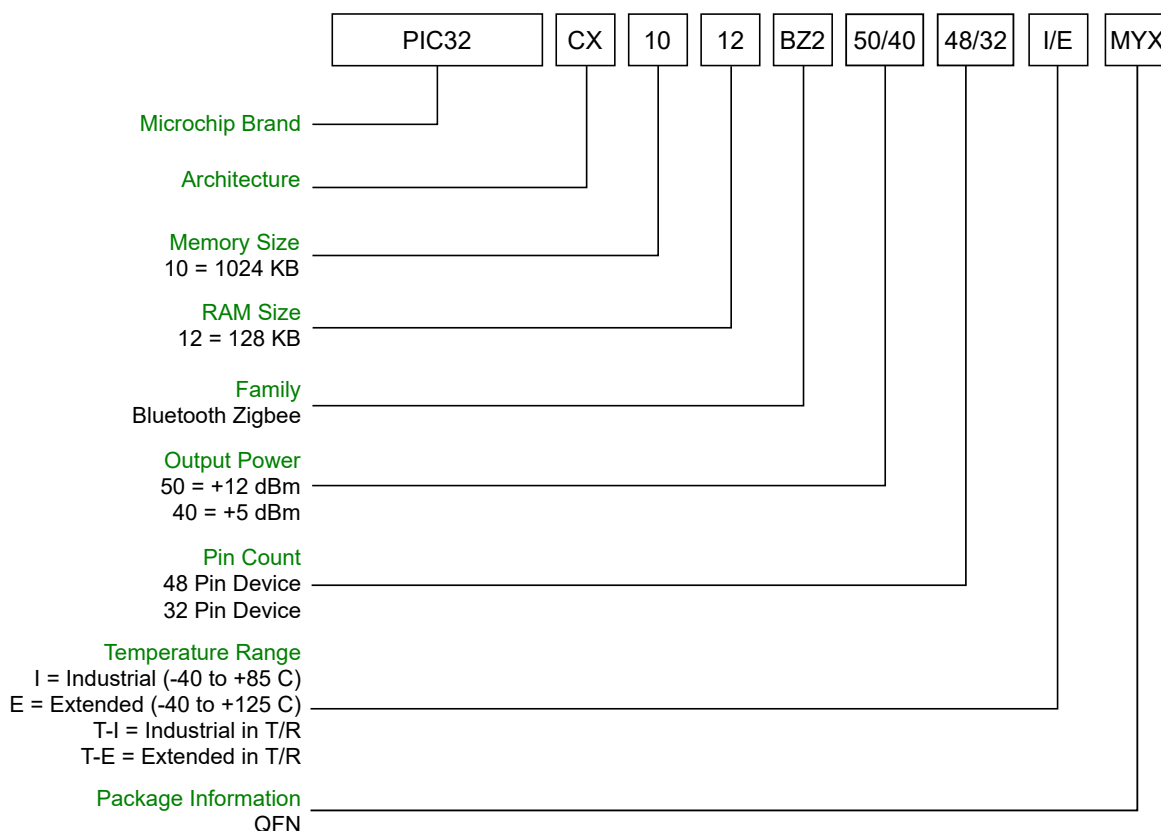
The following table describes the ordering information of the PIC32CX-BZ2 SoC.

Table 1-1. PIC32CX-BZ2 SoC Ordering Details

SoC Name	Pin and Package	Description	Ordering Code
PIC32CX1012BZ25048	48-pin and QFN (7 mm x 7 mm x 0.9 mm)	32-bit ARM®Cortex™-M4 with MCU Bluetooth/Zigbee Connectivity	PIC32CX1012BZ25048-I/MYX
PIC32CX1012BZ24032	32-pin and QFN (5 mm x 5 mm x 1 mm)		PIC32CX1012BZ24032-I/MYX

The following figure illustrates the details of the PIC32CX-BZ2 ordering information.

Figure 1-1. PIC32CXBZ2 Ordering Information



1.2 WBZ45 Module Ordering Information

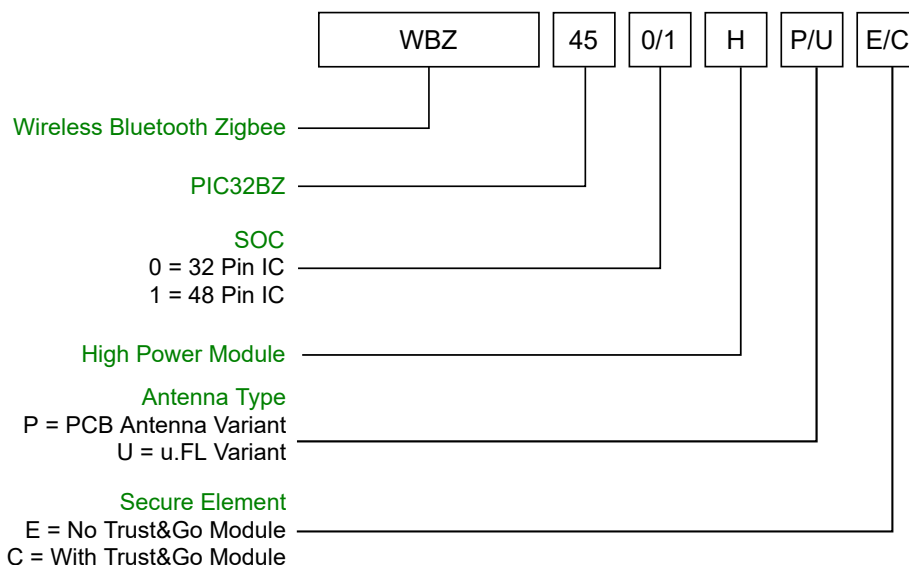
The following table describes the ordering information of the WBZ45 Module.

Table 1-2. WBZ45 Module Ordering Details

Model No.	Module SoC	Description	Regulatory Certification	Ordering Code
WBZ450PE	PIC32CX1012BZ24032-I/MYX	WBZ450 module with PCB antenna	FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC	WBZ450PE-I
WBZ450UE		WBZ450 module with U.FL connector for external antenna	FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC	WBZ450UE-I
WBZ451PE	PIC32CX1012BZ25048-I/MYX	WBZ451 module with PCB antenna	FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC	WBZ451PE-I
WBZ451UE		WBZ451 module with U.FL connector for external antenna	FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC	WBZ451UE-I
WBZ451HPE	PIC32CX1012BZ25048-I/MYX	WBZ451H module with PCB antenna	FCC, IC, RED, UKCA	
WBZ451HUE		WBZ451H module with U.FL connector for external antenna	FCC, IC, RED, UKCA	

The following figure illustrates the details of the WBZ45 ordering information.

Figure 1-2. WBZ451H Ordering Information



2. Configuration Summary

Table 2-1. PIC32CX-BZ2 and WBZ45 Family Features

Device	Program Memory (KB)	Data Memory (KB)	Pins	Package	Peripherals														Analog		Security			Wireless				
					SERCOM	Timer/Counter (TC)	TCC (24-bit/16-bit)	QSPI	DMA Channels	RTC	CCL/LUT	WDT	DMT	Frequency Meter	Event System (Channels)	External Interrupt Lines	GPIO Pins	Analog Comparators (Channels)		ADC (Channels)	Temperature Sensor	AES	TRNG	Public Key Cryptography (PUKCC)	Integrity Check Monitor	Max TX Power (dBm)	Bluetooth 5.2	802.15.4/Zigbee 3.0
PIC32CX1012BZ25048	1024	128	48	QFN	4											29 ⁽¹⁾	2	8							12			
PIC32CX1012BZ24032			32	QFN	2												16 ⁽¹⁾	1	6						4			
WBZ451			39	LGA	4	4	2/1	Y	16	Y	1/2	Y	Y	Y	32	4	29	2	8	Y	Y	Y	Y	Y	12	Y	Y	
WBZ450			30	LGA	2												16	1	6						4			
WBZ451H			39	LGA	4												26	2	8						20			

Note:

1. Only when SOSC is not used.

3. PIC32CX-BZ2 SoC Description

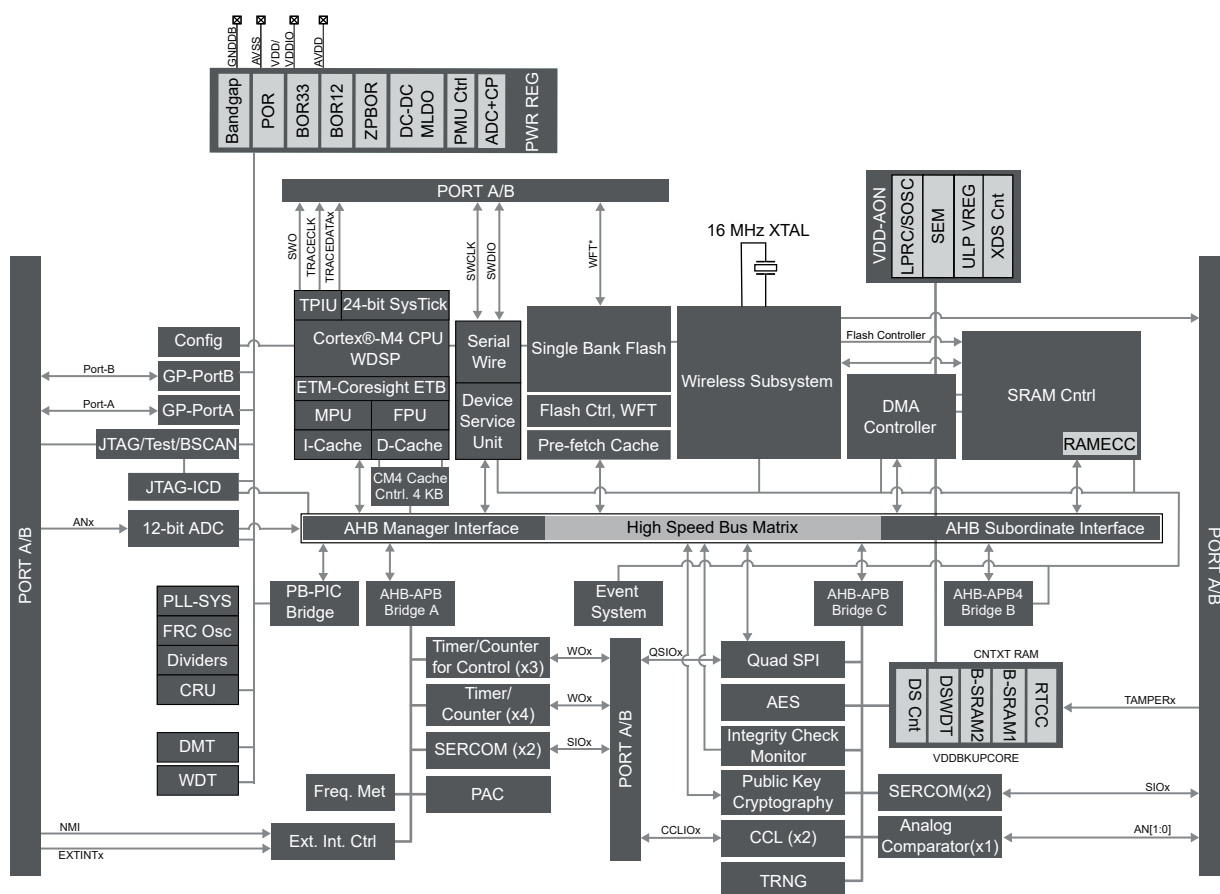
This chapter contains device-specific information for the PIC32CX-BZ2 SoC.

Note: Traditional AHB and APB documentation uses the terminology “Master” and “Slave.” The equivalent Microchip terminology used in this document is “Manager” and “Subordinate,” respectively.

3.1 PIC32CX-BZ2 SoC Block Diagram

The following figure illustrates the block diagram of the core and peripheral modules in the PIC32CX-BZ2 SoC.

Figure 3-1. PIC32CX-BZ2 SoC Block Diagram



3.2 Pinout Diagram

This section provides details on pin diagrams for each variant of the PIC32CX-BZ2 SoC.

Figure 3-2. PIC32CX1012BZ25048 SoC Pin Diagram

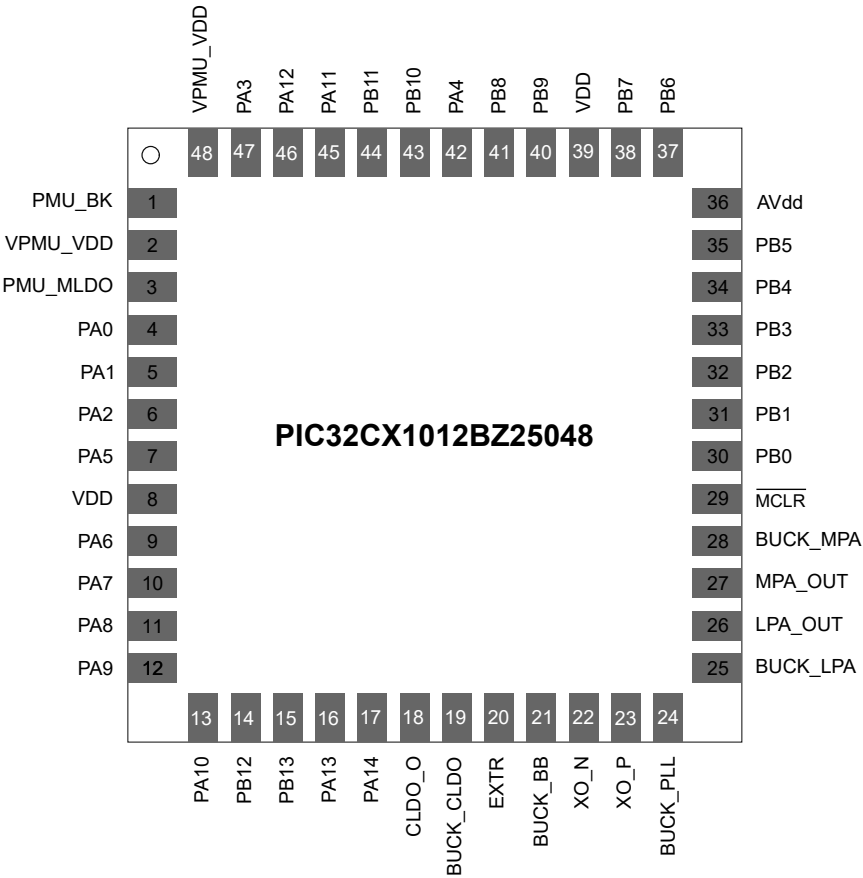
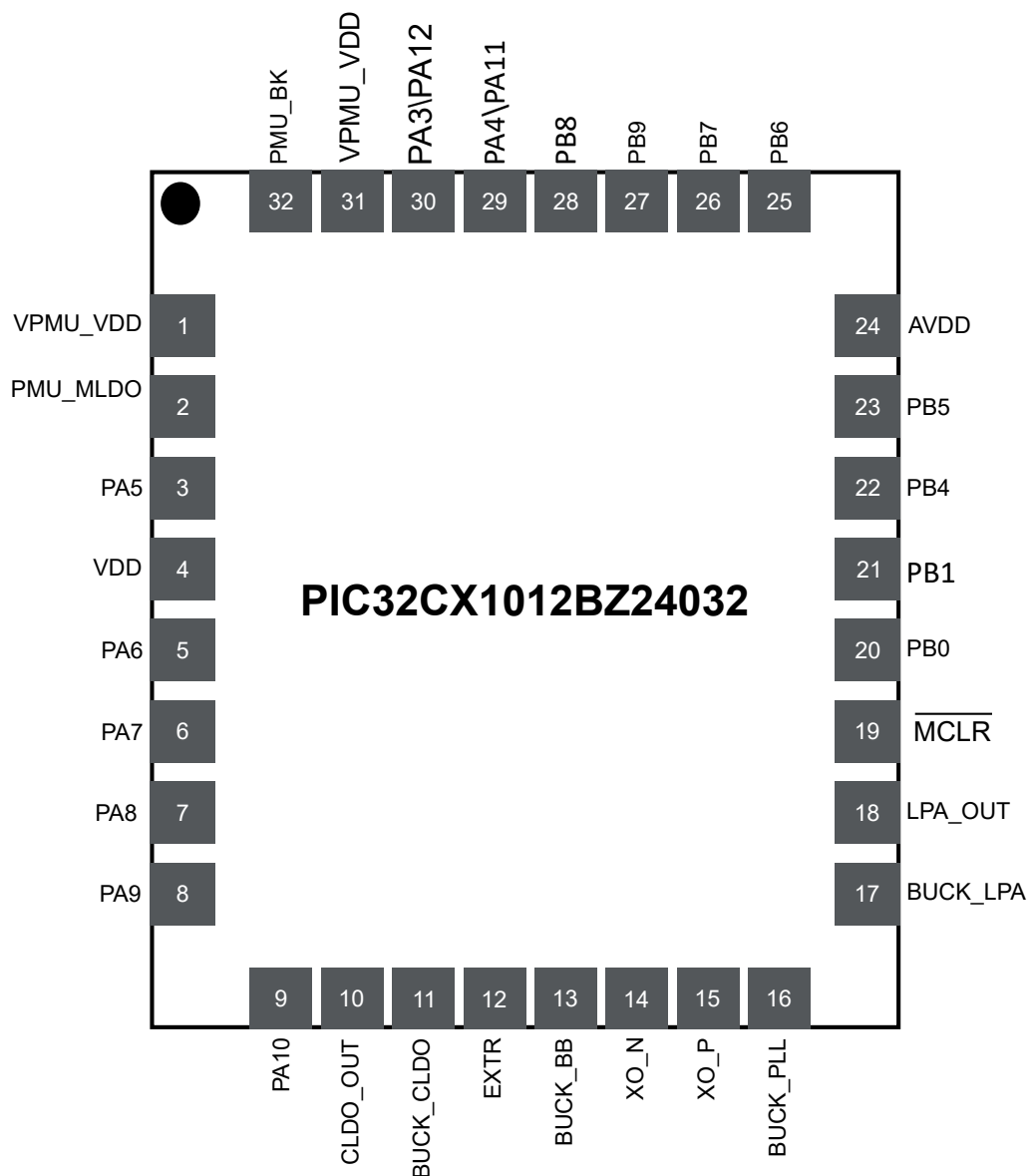


Figure 3-3. PIC32CX1012BZ24032 SoC Pin Diagram



Note:

1. It is required that the exposed paddle on the bottom of the SoC be connected to ground in the PCB.

4. WBZ45 Module Description

The WBZ45 Module contains the PIC32CX-BZ2 SoC with following antenna options:

- PCB antenna
- u.FL connector for external antenna

The following figure illustrates the WBZ45 Module block diagram.

Figure 4-1. WBZ45 Module Block Diagram

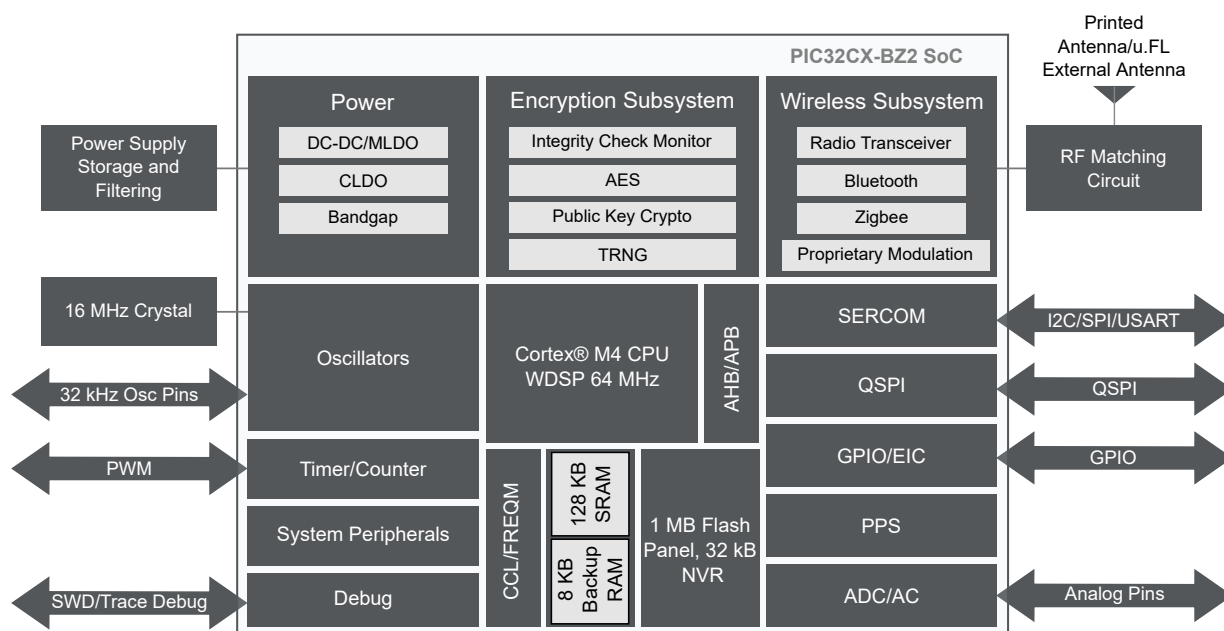
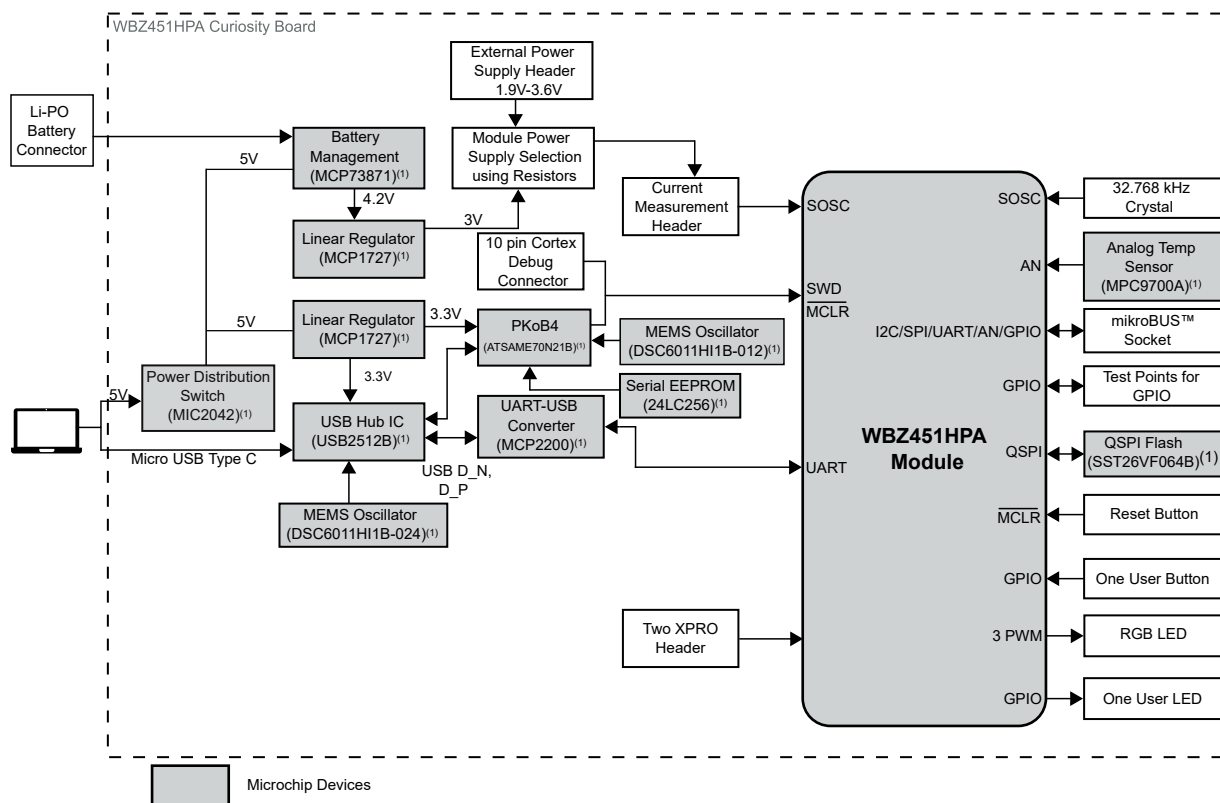


Figure 4-2. WBZ451H Module Block Diagram



4.1 Pinout Diagram

The following figures illustrate the module pinout diagram.

Figure 4-3. WBZ450 Module Pin Diagram

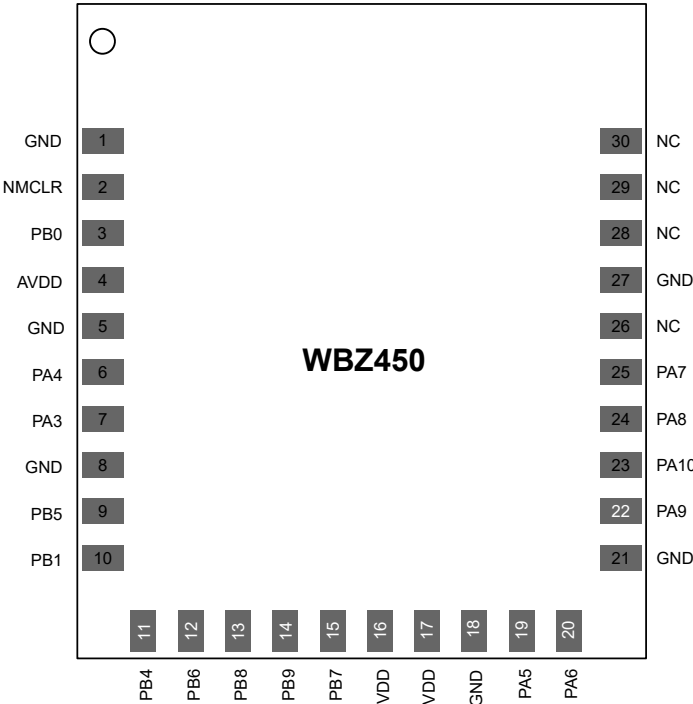


Figure 4-4. WBZ451 Module Pin Diagram

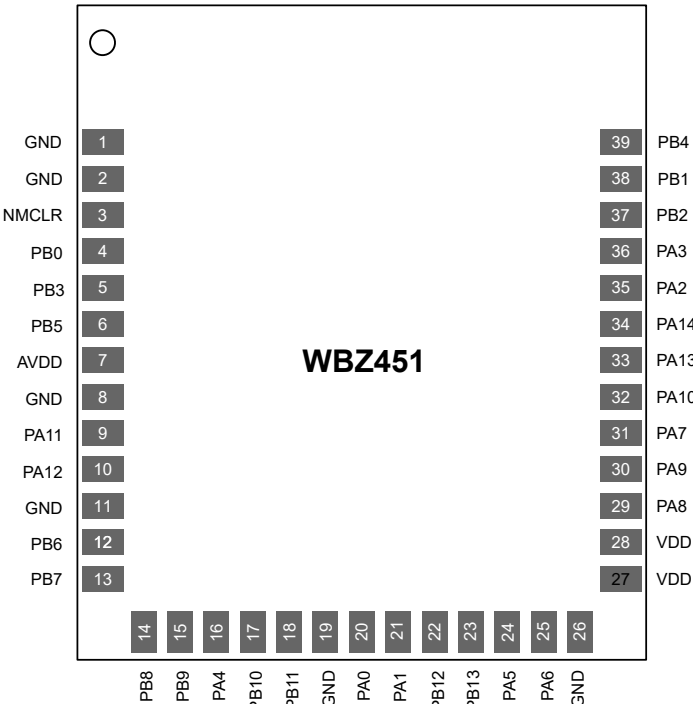
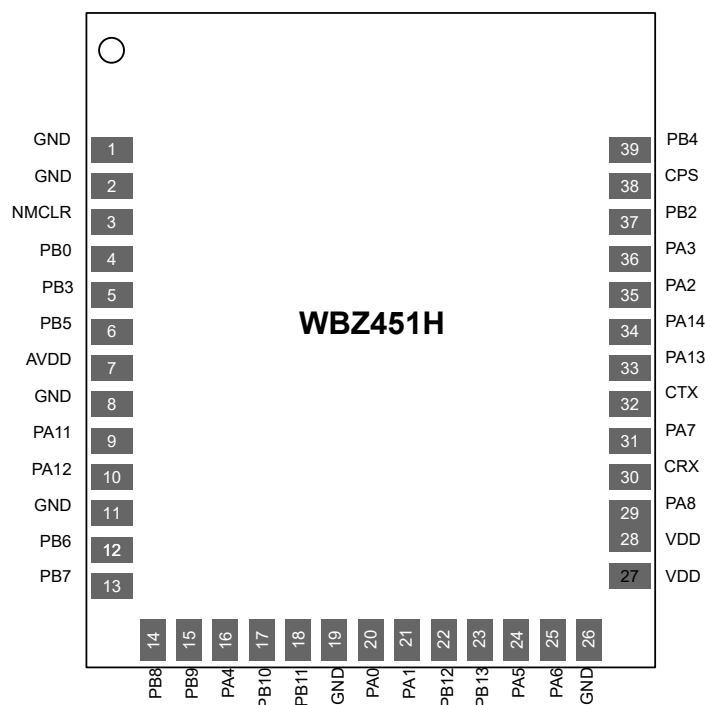


Figure 4-5. WBZ451H Module Pin Diagram



Note: It is required that the exposed paddle on the bottom of the module be connected to ground in the PCB.

4.2 Basic Connection Requirement

The WBZ45 Module requires attention to a minimal set of device pin connections before proceeding with development.

Figure 4-6. WBZ450 Module Basic Connection and Interface Diagram

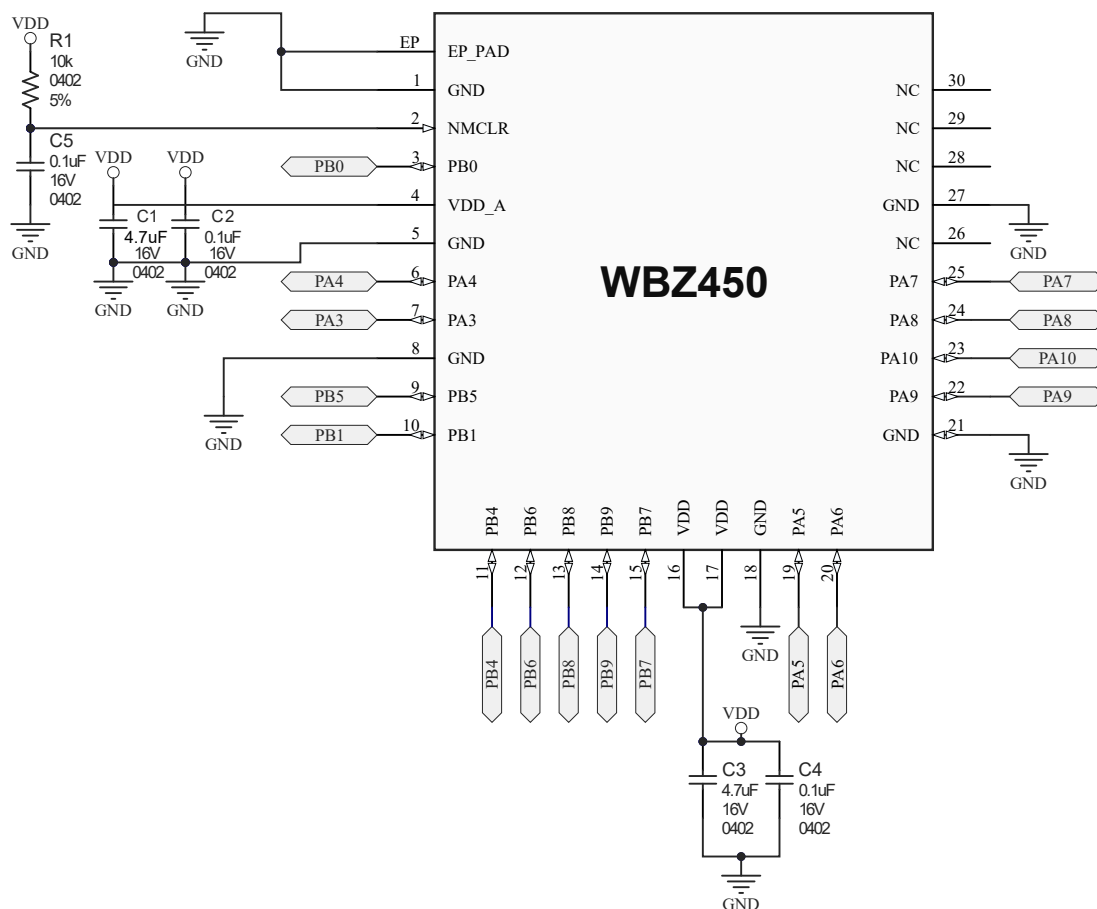
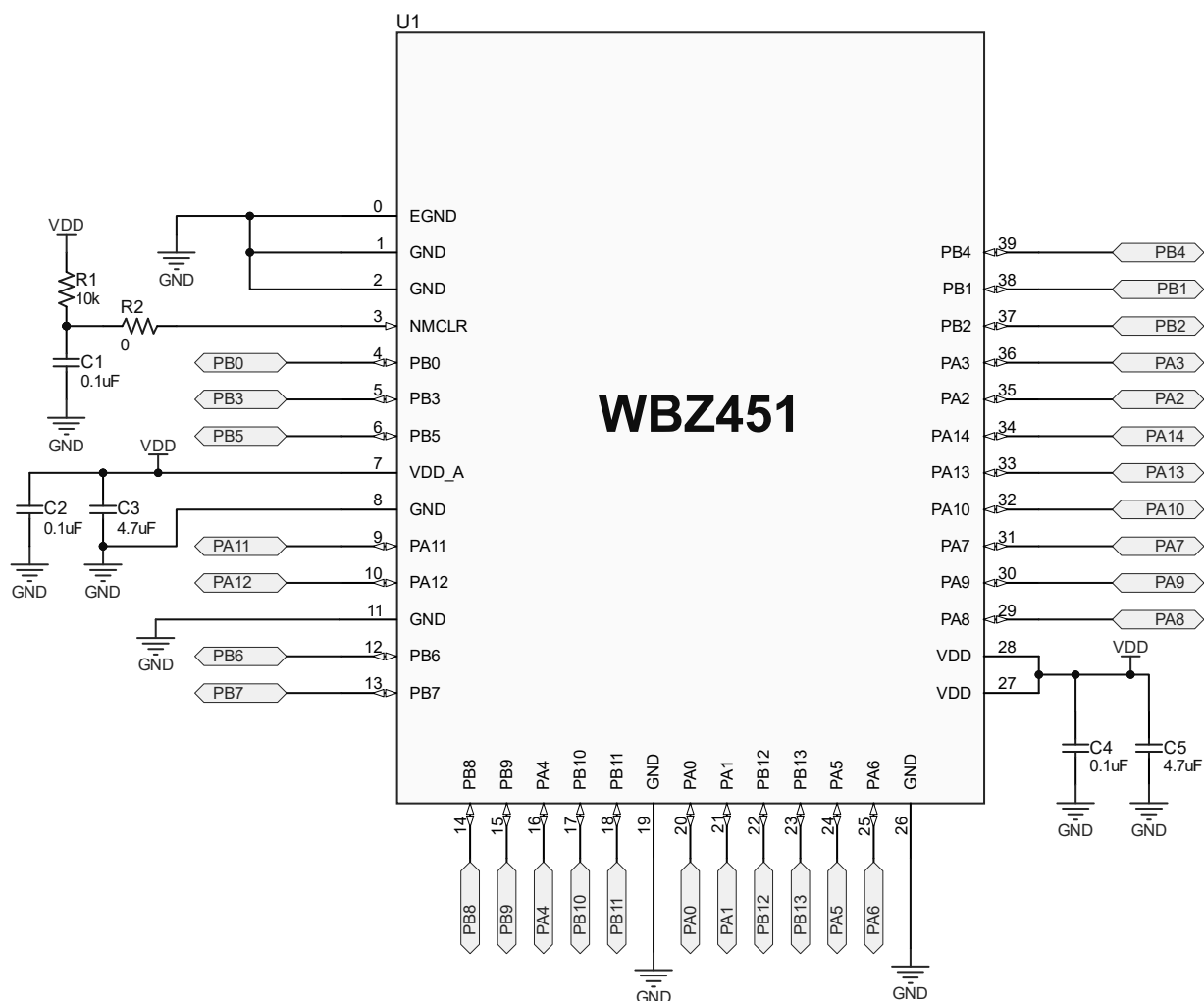


Figure 4-7. WBZ451 Module Basic Connection and Interface Diagram



4.2.1 Power Pins

It is recommended that a bulk and a decoupling capacitor be added at the input supply pin (VDD, VDD_A and GND pins) of the WBZ45 module.

- It is recommended to add a 4.7 μF and a 0.1 μF cap on both VDD and VDD_A pins.
- The value of the capacitors are based on typical application requirements and are the minimum recommended values. Depending on the application requirement (in other words, a noisy power line or other known noise sources), the values of capacitors can be adjusted to provide a clean supply to the module.
- All capacitors must be placed close to the Module Power supply pins.

Note:

The WBZ451H module can operate with a single power supply source (1.9V to 3.6V). The maximum power consumption of the WBZ451HPA module will be ~300 mA (WBZ451(200 mA) +SKY66114-11(100 mA)).

All the power supply de-coupling capacitors will be part of the module.

The module has a BOM option (for evaluation) to feed AVDD from an external voltage input on pin 12 (VDD_A) of the module.

When FB2 is mounted, the VDD from pin 32 and 33 powers the AVDD, FB3 mounted VDD_A from pin 12 powers the AVDD. FB2 and FB3 share a pad on the layout.

4.2.2 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides for two specific device functions:

- Device Reset
- Device programming and debugging

Pulling the $\overline{\text{MCLR}}$ pin low generates a device Reset. Module Basic Connection and Interface Diagram illustrates a typical $\overline{\text{MCLR}}$ circuit, see the *Module Basic Connection and Interface Diagram* in the *Basic Connection Requirement* from Related Links.

The module has sufficient filtering (0.1 μF) and pull-up (10k) on the Reset line. On a typical application, no extra filtering is required on this pin.

Related Links

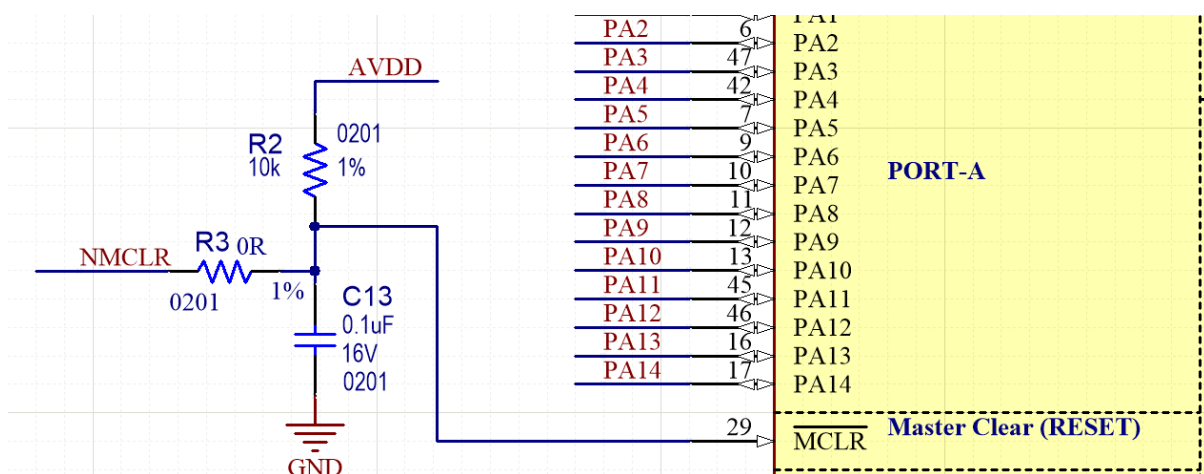
[4.2. Basic Connection Requirement](#)

4.2.3 Reset Requirements

The Reset line ($\overline{\text{MCLR}}$) will have the following circuit and will be available on the module LGA Pads.

The $\overline{\text{MCLR}}$ is pulled up to AVDD for routing convenience. Since they are of the same voltage levels.

Figure 4-8. Reset Requirements



4.2.4 SWD Lines

For SWD programming and debugging purposes, use the CM4_SWCLK, CM4_SWDIO and CM4_SWO pins. The recommendation is to use the CM4_SWCLK and CM4_SWDIO pins for the WBZ45 Module for SWD as the default configuration (CM4_SWO can be optional).

Keep the trace length between the SWD pins of the WBZ45 Module and the SWD header as short as possible. If the SWD connector is expected to experience an ESD event, a series resistor is recommended with the value in the range of a few tens of Ω s, not to exceed 100 Ω .

Note: Provide an option for adding an external pull-up on SWDIO.

4.2.5 Unused I/O Pins

It is recommended that unused I/O pins not be allowed to float as inputs. They can be configured as inputs and pulled up. Alternatively, depending on the application, they can be pulled down as well.

4.2.5.1 GPIO Pins/PPS Functions

Most of the WBZ45 Module pins can be configured as GPIOs pins or for PPS functionality. To find the functionality supported by each of these GPIOs, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links. The recommendation is to add a series resistor on the host board for all critical, high frequency pins and clocks for EMI considerations. The value of the series resistor depends on the actual pin configuration. The user must place these resistors close to the module. Example of Host Board on Top Layer illustrates the placement of the series resistor; see the *Example of Host Board on Top Layer* figure in the *WBZ45 Module Routing Guidelines* from Related Links.

Related Links

[6. I/O Ports and Peripheral Pin Select \(PPS\)](#)

[4.4. WBZ45 Module Routing Guidelines](#)

4.2.6 WBZ451H Module Interface Specifications

The WBZ451H module supports all the interfaces available except of below pins on the Chimera SoC. Refer to the [4.1. Pinout Diagram](#) for more details.

PB1 will be configured as By-Pass mode control (CPS) of the FEM (SKY66114-11), so that CPS will disable while the target power ≥ -1 dBm and enable while the target power < -1 dBm.

PA10 will sync with Tx signal and connects with both CTX and CHL for transmitting High-power mode.

PA9 will sync with Rx signal and connects with CRX for Receive mode.

CSD pull-high with fixed hardware control.

Table 4-1. Mode Control

Mode	Description	CSD (Pin 7)	CPS (Pin 15)	CRX (Pin 6)	CTX (Pin 2)	CHL (Pin 3)
0	All off (Sleep mode) ⁽¹⁾	0	X	X	X	X
1	Receive LNA mode	1	0	1	0	X
2	Transmit High-power mode	1	0	X	1	1
3	Transmit Ligh-power mode	1	0	X	1	0
4	Receive Bypass mode	1	1	1	0	X
5	Transmit Bypass mode	1	1	X	1	X
6	All off (Sleep mode)	1	X	0	0	X

1. All controls must be logic "0" to achieve the specified sleep current.

2. X = Don't care.

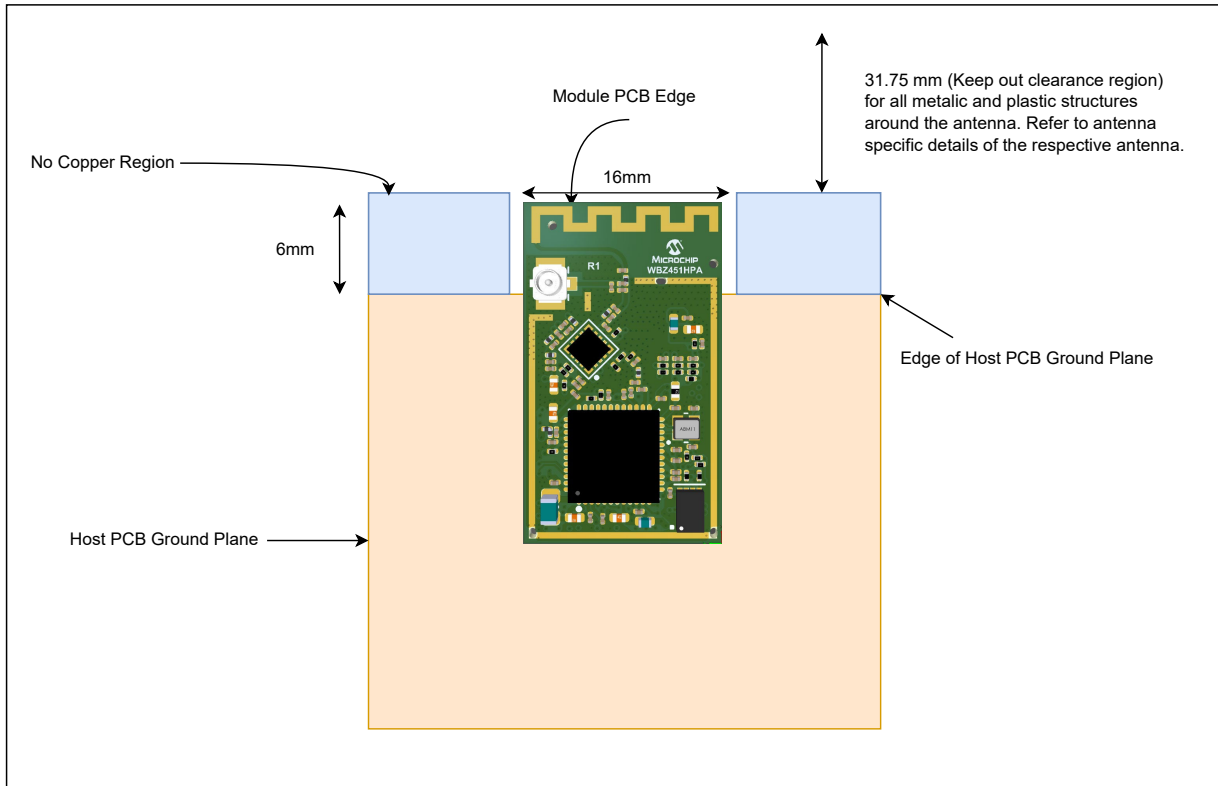
While device in Tx operation, the state will be in mode 2 and depends on target power can switch to mode 5 automatically; while device in Rx operation, the state will be in mode 1; while no RF operation, the state will be in mode 6 and stay in Sleep mode for power saving.

4.3 WBZ45 Module Placement Guidelines

- For any Bluetooth Low Energy/Zigbee product, the antenna placement affects the performance of the whole system. The antenna requires free space to radiate RF signals and it must not be surrounded by the ground plane. Thus, for the best PCB antenna performance, it is recommended that the WBZ45 module be placed at the edge of the host board.
- It is recommended that the WBZ45 module ground outline edge be aligned with the edge of the host board ground plane as shown in the following figure.
- A low-impedance ground plane for the WBZ45 module ensures the best radio performance (best range and lowest noise). The ground plane can be extended beyond the minimum recommendation as required for the host board EMC and noise reduction.
- For the best performance, keep metal structures and components (such as mechanical spacers, bump-on and so on) at least 31.75 mm away from the PCB trace antenna as illustrated in the following figure.
- It is recommended that the antenna on the WBZ45 module not be placed in direct contact with or in close proximity to plastic casing or objects. Keep a minimum clearance of 10 mm in all directions around the PCB antenna as shown in the following figure. Keeping metallic and plastic objects close to the antenna can detune the antenna and reduce the performance of the device.
- Exposed GND pads on the bottom of the WBZ45 module must be soldered to the host board (see the *Example of Host Board on Top Layer* figure in the *WBZ45 Module Routing Guidelines* from Related Links).
- A PCB cutout or a copper keepout is required under the RF test point (see *WBZ451 Module Packaging Information* from Related Links).
- Copper keepout areas are required on the top layer under voltage test points (see *WBZ451 Module Packaging Information* from Related Links).
- Alternatively, the entire region, except the exposed ground paddle, can be solder-masked.

The following figure illustrates the examples of WBZ45 module placement on a host board with a ground plane. Refer to the following figure for placement-specific guidance.

Figure 4-9. Module Placement Guidelines



Related Links

[4.4. WBZ45 Module Routing Guidelines](#)

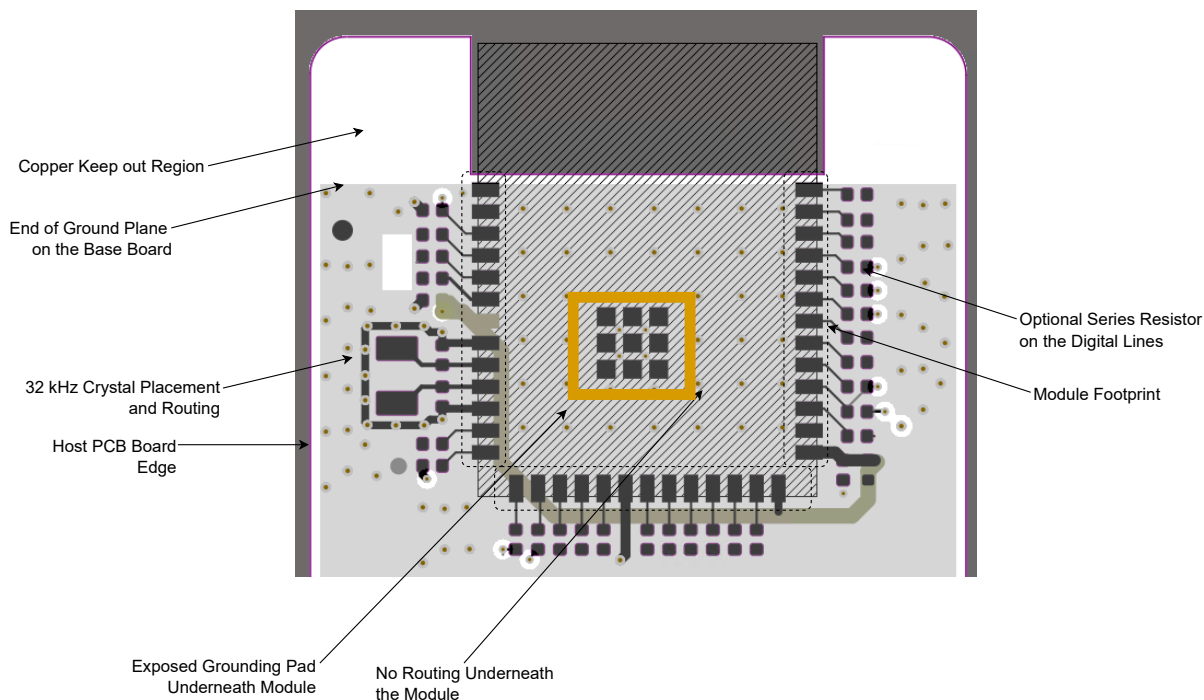
[44.4. WBZ450 Module Packaging Information](#)

[44.3. WBZ451 Module Packaging Information](#)

4.4 WBZ45 Module Routing Guidelines

- Use the multi-layer host board for routing signals on the inner layer and the bottom layer.
- The top layer (underneath the module) of the host board must be ground with as many GND vias as possible, (see the following figure).
- Avoid fan-out of the signals under the module or antenna area. Use a via to fan-out signals to the edge of the WBZ45 Module.
- For a better GND connection to the WBZ45 Module, solder the exposed GND pads of the WBZ45 Module on the host board.
- For the module GND pad, use a GND via of a minimum 10 mil (hole diameter) for good ground to all the layers and thermal conduction path.
- The recommendation is to have a series resistor on the host board for all GPIOs. Place these resistors close to the WBZ45 Module. The following figure illustrates the placement of the series resistor.
- Place the SOSC crystal (32.768 kHz) on the host board close to the WBZ45 Module and follow the shortest trace routing length with no vias (see the following figure).

Figure 4-10. Example of Host Board on Top Layer



4.5 WBZ45 Module RF Considerations

The product design, environment and application significantly affect the overall performance of the system. The product designer must ensure system-level shielding (if required) and verify the performance of the product features and applications.

The following are the guidelines to consider for optimal RF performance:

- Position the WBZ45 Module in a noise-free RF environment, keep it far away from high-frequency clock signals and any other sources of RF energy.
- Do not shield the antenna by any metal objects.
- The power supply must be clean and noise-free.
- Ensure that the width of the traces routed to GND, VDD rails are sufficiently large for handling peak TX current consumption.

Note: The WBZ45 Module includes RF shielding on top of the board as a standard feature.

WBZ451H RF Specifications

The 48 pin QFN LPA/MPA combined with external FEM design to achieve +20 dBm output power. WBZ451H module has the matching network for doing a combined match of the LPA and MPA path along with a harmonic filter followed by an antenna matching circuit and an antenna.

4.6 WBZ45 Module Antenna Considerations

4.6.1 PCB Antenna

For the WBZ45 Module, the PCB antenna is fabricated on the top copper layer and covered in a solder mask. The layers below the antenna do not have copper trace. It is recommended that the module be mounted on the edge of the host board and to have no PCB material below the antenna structure of the module and no copper traces or planes on the host board in that area.

The following table lists the technical specification of the PCB antenna when tested with the WBZ45 Module mounted on an Evaluation Board.

Table 4-2. PCB Antenna Specification for WBZ450 Module

Parameter	Specification
Operating frequency	2400 to 2480 MHz
Peak gain	4.14 dBi at 2440 MHz
Efficiency	50%

Table 4-3. PCB Antenna Specification for WBZ451 Module

Parameter	Specification
Operating frequency	2400 to 2480 MHz
Peak gain	2.36 dBi at 2420 MHz
Efficiency	50%

PCB Antenna Radiation Pattern

The following figures illustrate the antenna radiation patterns for WBZ450 and WBZ451 modules.

Figure 4-11. WBZ450 Antenna Radiation Pattern when $\Phi = 0^\circ$

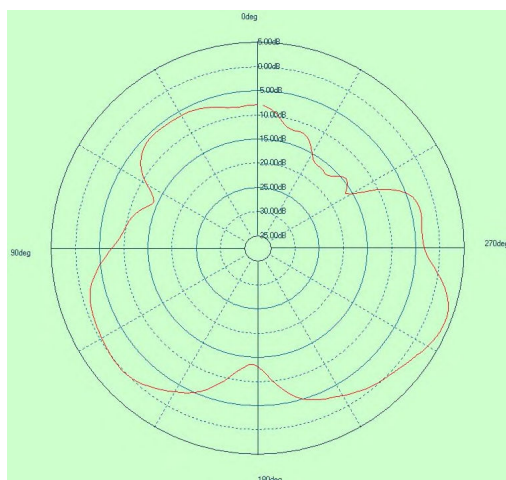


Figure 4-12. WBZ450 Antenna Radiation Pattern when $\Phi = 90^\circ$

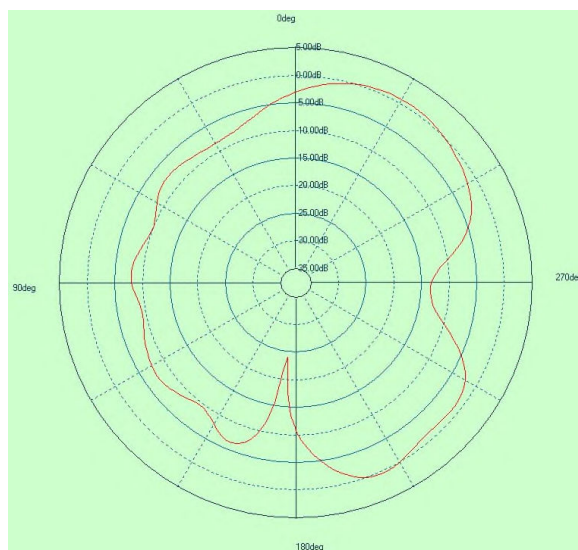


Figure 4-13. WBZ450 Antenna Radiation Pattern when $\Theta = 90^\circ$

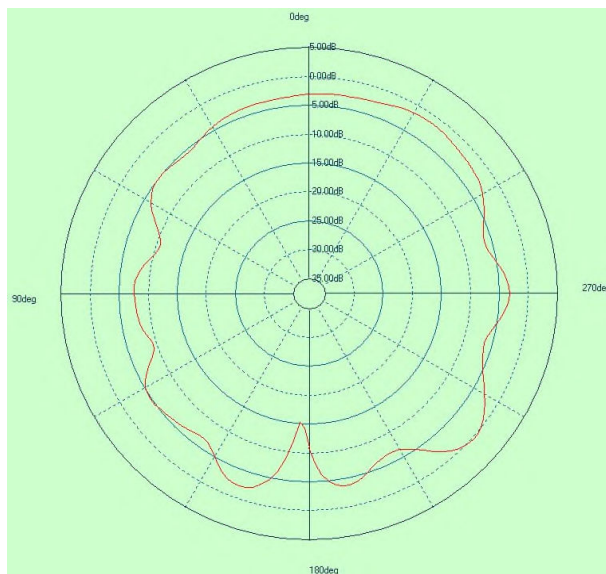


Figure 4-14. WBZ451 Antenna Radiation Pattern when $\Phi = 0^\circ$

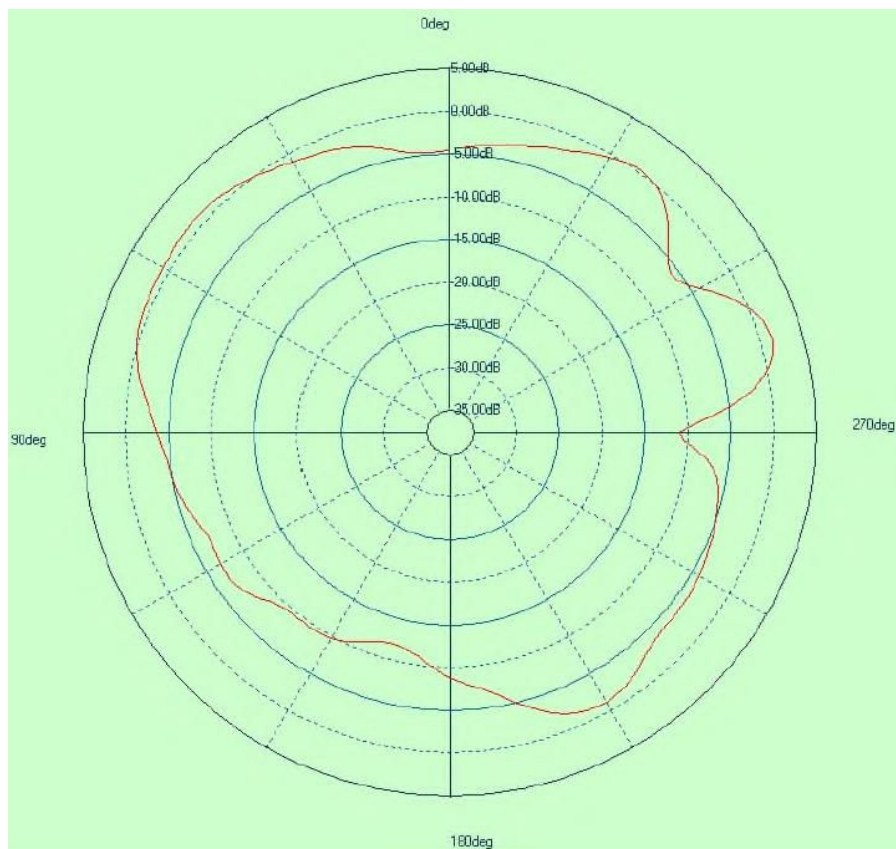


Figure 4-15. WBZ451 Antenna Radiation Pattern when $\Phi = 90^\circ$

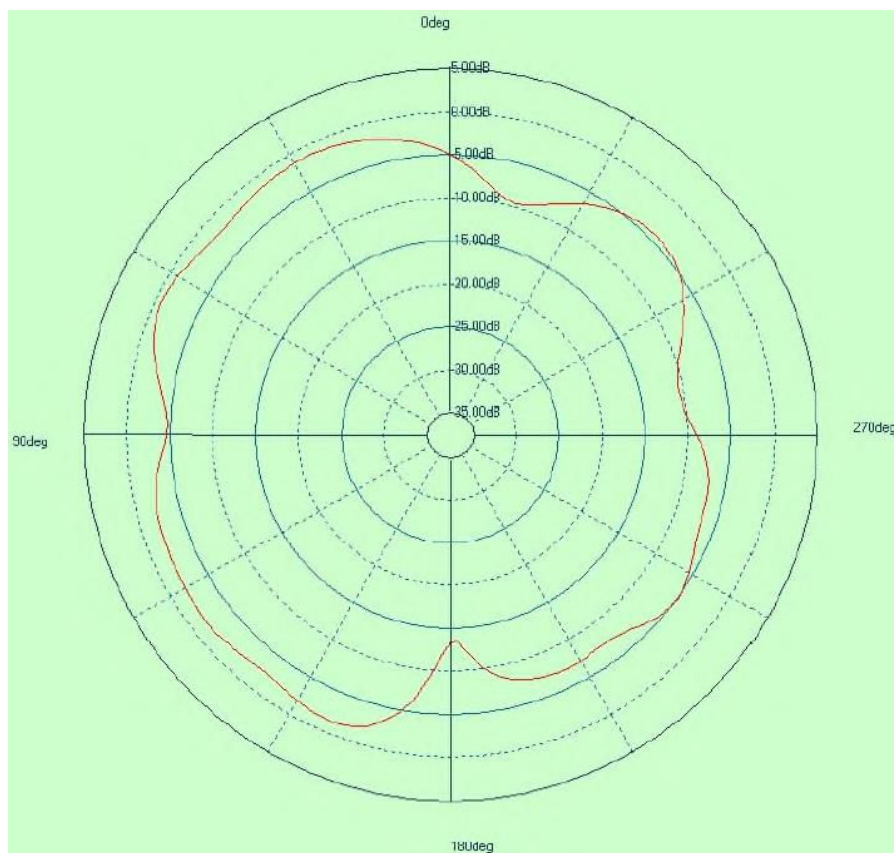


Figure 4-16. WBZ451 Antenna Radiation Pattern when Theta = 90°

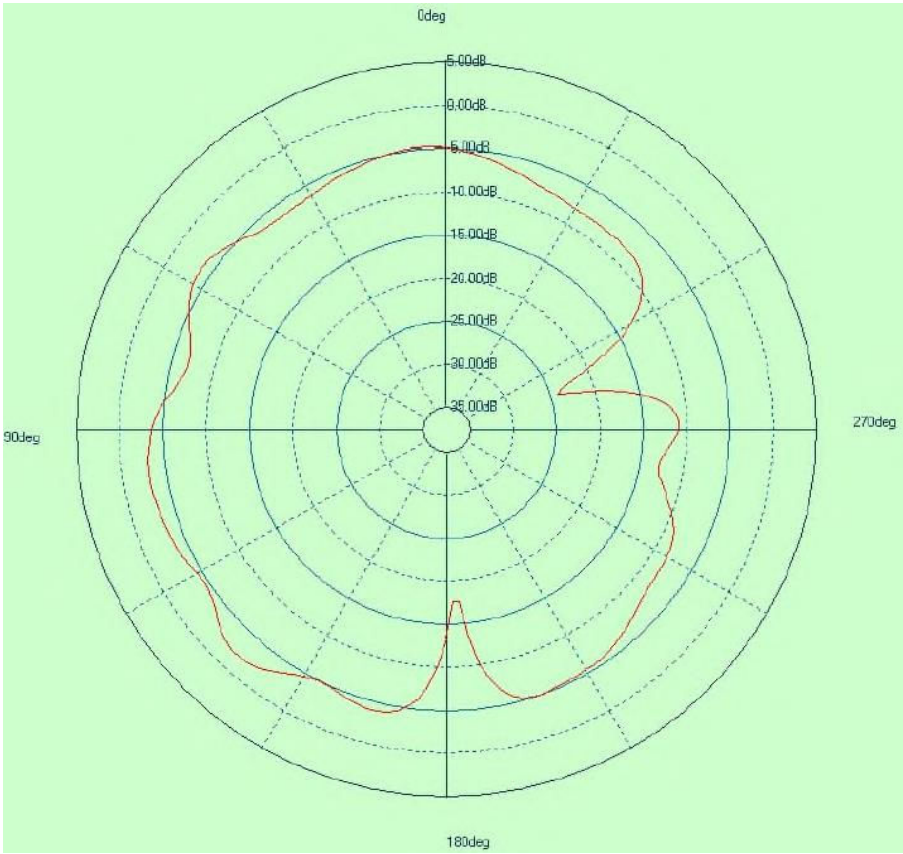


Figure 4-17. WBZ451H Antenna Radiation Pattern when Phi = 0°

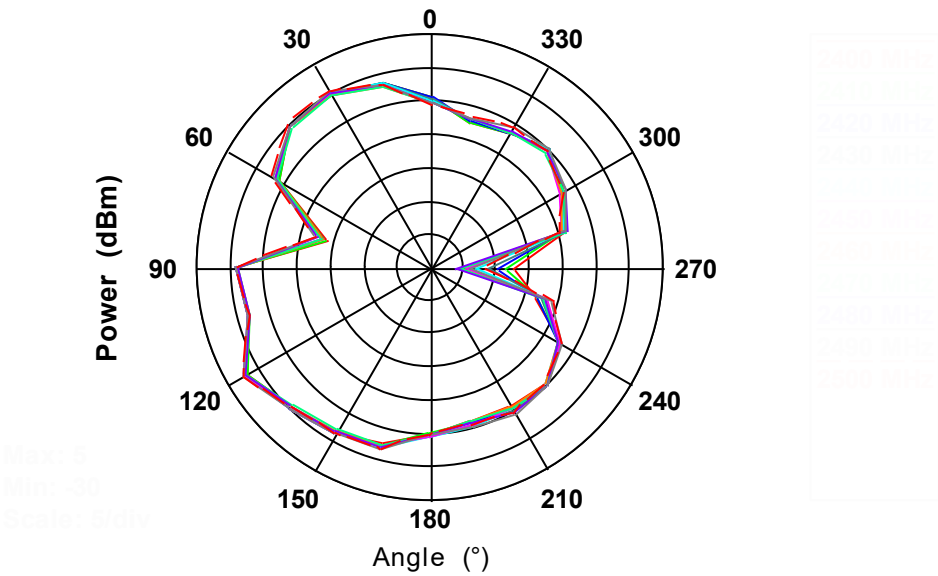


Figure 4-18. WBZ451H Antenna Radiation Pattern when $\Phi = 90^\circ$

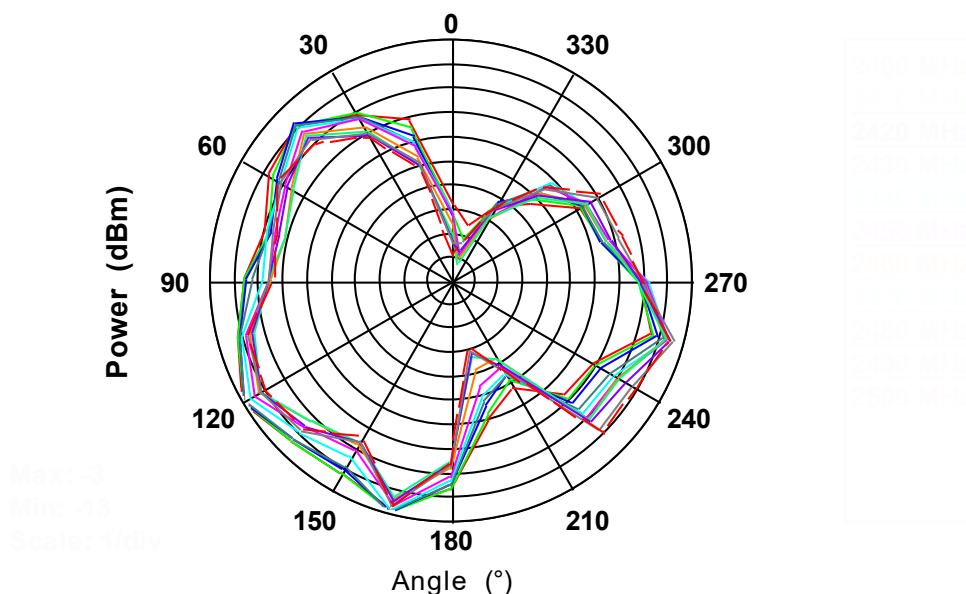
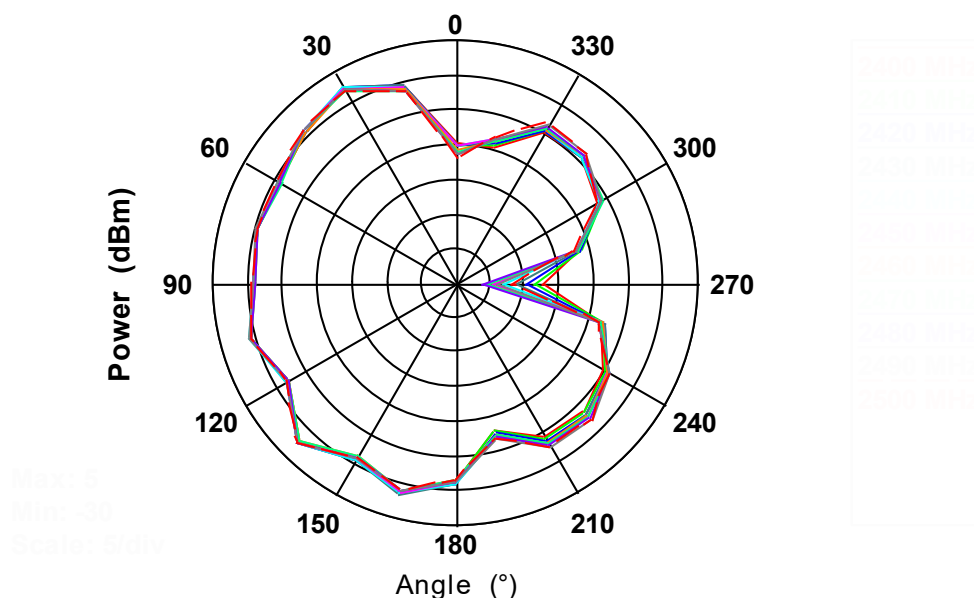


Figure 4-19. WBZ451H Antenna Radiation Pattern when $\Theta = 90^\circ$



4.6.2 External Antenna Placement Recommendations

The following recommendations must be applied for the placement of the antenna and its cable:

- The antenna cable must not be routed over circuits generating electrical noise on the host board or alongside or underneath the module. It is preferred that the cable is routed straight out of the module.
- The antenna must not be placed in direct contact or in close proximity of the plastic casing/objects (except when the selected antenna specifically recommends it).
- Do not enclose the antenna within a metal shield.

- Keep any components that may radiate noise, signals or harmonics within the 2.4-2.5 GHz frequency band away from the antenna and, if possible, shield those components. Any noise radiated from the host board in this frequency band degrades the sensitivity of the module.
- The antenna must be placed at a distance greater than 5 cm away from the module. The following figure illustrates the antenna keepout area where the antenna must not be placed.

These recommendations are based on an open-air measurement and do not take into account any metal shielding of the customer end product. When a metal enclosure is used, the antenna can be located closer to the WBZ45 Module.

Note: These are generic guidelines and it is recommended that customers check and fine-tune the antenna positioning in the final host product based on RF performance.

The following figure illustrates an indication on how to route the antenna cable depending on the location of the antenna with respect to the WBZ45 PCB; there are two possible options for the optimum routing of the cable.

Figure 4-20. WBZ45 Antenna Placement Guidelines

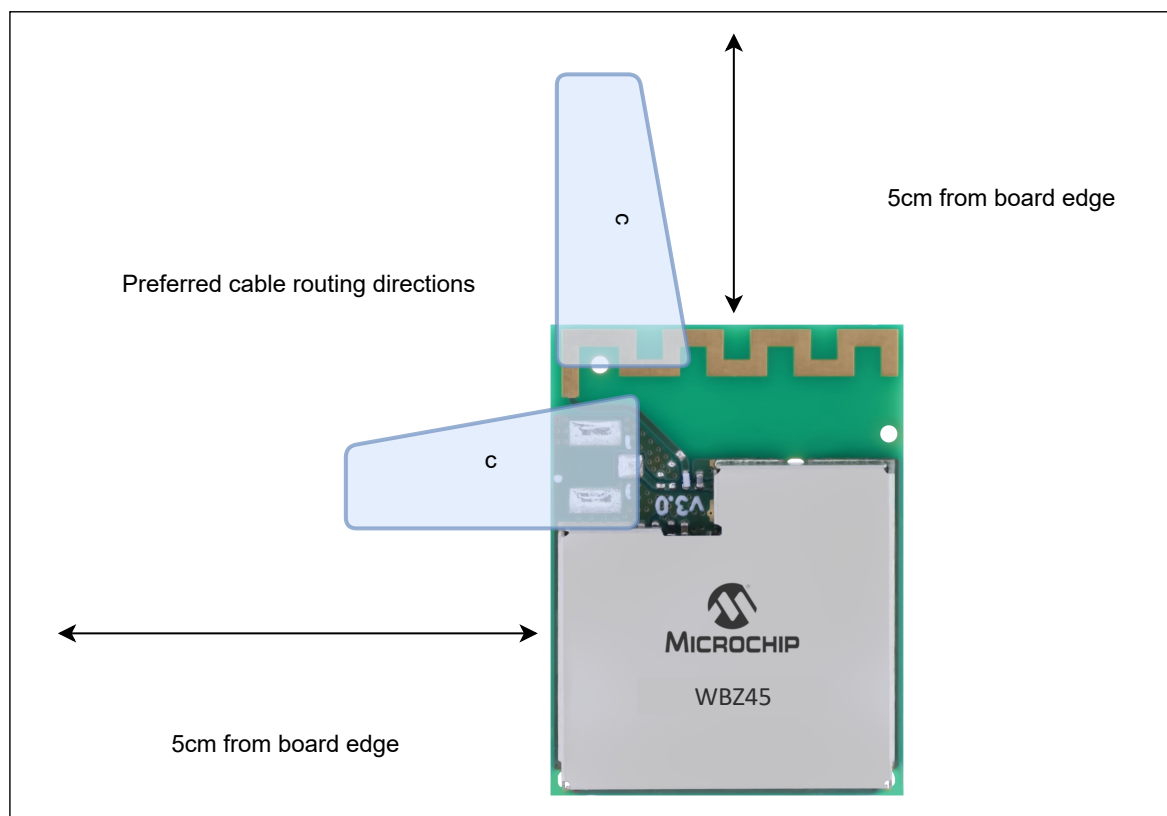


Table 4-4. List of Certified Antenna

Serial Number	Part Number	Vendor	Antenna Type	Gain	Comment
1	W3525B039	Pulse	PCB	2 dBi	Cable length 100 mm
2	001-0016	LSR	PIFA	2.5 dBi	Flex PIFA antenna
3	001-0001	LSR	Dipole	2 dBi	RPSMA connector*
4	1461530100	Molex	PCB	3 dBi	100 mm (Dual Band)
5	ANT-2.4-LPW-125	Linx Technologies	Dipole	2.8 dBi	125 mm
6	RFA-02-P05-D034	Alead	PCB	2 dBi	150 mm
7	RFA-02-P33-D034	Alead	PCB	2 dBi	150 mm

.....continued

Serial Number	Part Number	Vendor	Antenna Type	Gain	Comment
8	ABAR1504-S2450	ABRACON	PCB	2.28 dBi	250 mm
9	WBZ451H LGA	Microchip	PCB	4.08 dBi	On-board PCB antenna

4.7 WBZ45 Module Reflow Profile Information

The WBZ45 Module was assembled using the IPC/JEDEC J-STD-020 standard lead-free reflow profile. The WBZ45 Module can be soldered to the host board using standard leaded or lead-free solder reflow profiles. To avoid damaging the module, adhere to the following recommendations:

- For solder reflow recommendations, refer to the *AN233 Solder Reflow Recommendation Application Note* ([DS00233](#)).
- Do not exceed a peak temperature (TP) of 250°C.
- For specific reflow profile recommendations from the vendor, refer to the *Solder Paste Data Sheet*.
- Use no-clean flux solder paste.
- Do not wash as moisture can be trapped under the shield.
- Use only one flow. If the PCB requires multiple flows, apply the module on the final flow.

4.7.1 Cleaning

The exposed GND pad helps to self-align the module, avoiding pad misalignment. The recommendation is to use the no clean solder pastes. Ensure full drying of no-clean paste fluxes as a result of the reflow process. As per the recommendation by the solder paste vendor, this requires longer reflow profiles and/or peak temperatures toward the high end of the process window. The uncured flux residues can lead to corrosion and/or shorting in accelerated testing and possibly the field.

4.8 WBZ45 Module Assembly Considerations

The WBZ45 Module is assembled with an EMI shield to ensure compliance with EMI emission and immunity rules. The EMI shield is made of a tin-plated steel (SPTE) and is not hermetically sealed. Use the solutions such as IPA and similar solvents to clean this module. The user must never use cleaning solutions containing acid on the module.

4.8.1 Conformal Coating

The modules are not intended for use with a conformal coating, and the customer assumes all risks (such as the module reliability, performance degradation and so on) if a conformal coating is applied to the modules.

5. Pinout and Signal Descriptions List

The following table provides details on signal names classified by the peripherals along with the device pinout for each variant of the PIC32CX-BZ2 SoC and WBZ45 module.

Table 5-1. Pinout and Signal Descriptions List

PIC32CX-BZ2 SoC		WBZ45 Module			Pad Name	Peripherals									
24032	25048	450	451	WBZ451H		AC	ADC	EIC ⁽⁴⁾	GPIO (1, 2)	QSPI	RTCC	SERCOM	OSC	RF	DEBUG
		1, 5, 8, 18, 21, 27	1, 2, 8, 11, 19, 26	1, 2, 8, 11, 19, 26	GND	—	—	—	—	—	—	—	—	—	—
32	1				PMU_BK	—	—	—	—	—	—	—	—	—	—
1	2				VPMU_VDD	—	—	—	—	—	—	—	—	—	—
2	3				PMU_MLDO	—	—	—	—	—	—	—	—	—	—
	4		20	20	PA0	—	—	—	RA0	QSPI_DATA2	RTC_IN3	—	—	—	—
	5		21	21	PA1	AC_CMP1	—	—	RA1	QSPI_DATA3	RTC_IN2	—	—	—	—
	6		35	35	PA2	AC_CMP0	—	—	RA2	—	RTC_IN1	—	—	—	—
3	7	19	24	24	PA5	—	—	—	RA5	—	—	SERCOM0_PAD0	—	—	—
4	8	16, 17	27, 28	27, 28	VDD	—	—	—	—	—	—	—	—	—	—
5	9	20	25	25	PA6	AC_CMP1_ALT	—	—	RA6	—	—	SERCOM0_PAD1	—	—	—
6	10	25	31	31	PA7	—	—	—	RA7	—	—	SERCOM1_PAD0	—	—	TRACECLK
7	11	24	29	29	PA8	—	—	—	RA8	—	—	SERCOM1_PAD1	—	FECTRL0	—
8	12	22	30		PA9	—	—	—	RA9	—	RTC_IN0_ALT	SERCOM1_PAD2	—	FECTRL1	—
			30		CRX	—	—	—	—	—	—	—	—	—	—
9	13	23	32		PA10	—	—	—	RA10	—	RTC_OUT_ALT	SERCOM1_PAD3	—	FECTRL2	—
			32		CTX_CHL	—	—	—	—	—	—	—	—	—	—
	14		22	22	PB12	—	—	—	RB12	QSPI_DATA0	—	—	—	—	—
	15		23	23	PB13	—	—	—	RB13	QSPI_DATA1	RTC_EVENT	—	—	—	—
	16		33	33	PA13	—	—	—	RA13	—	—	SERCOM2_PAD0	—	COEXCTRL0	—
	17		34	34	PA14	—	—	—	RA14	—	—	SERCOM2_PAD1	—	COEXCTRL1	—
10	18				CLDO_OUT	—	—	—	—	—	—	—	—	—	—
11	19				BUCK_CLDO	—	—	—	—	—	—	—	—	—	—
12	20				EXTR ⁽⁹⁾	—	—	—	—	—	—	—	—	—	—
13	21				BUCK_BB	—	—	—	—	—	—	—	—	—	—
14	22				XO_N	—	—	—	—	—	—	—	XO-	—	—
15	23				XO_P	—	—	—	—	—	—	—	XO+	—	—
16	24				BUCK_PLL	—	—	—	—	—	—	—	—	—	—
17	25				BUCK_LPA	—	—	—	—	—	—	—	—	—	—
18	26				LPA_OUT	—	—	—	—	—	—	—	—	LPA	—
	27				MPA_OUT	—	—	—	—	—	—	—	—	MPA	—
	28				BUCK_MPA	—	—	—	—	—	—	—	—	—	—
19	29	2	3	3	NMCLR	—	—	—	—	—	—	—	—	—	—
20	30	3	4	4	PB0	AC_AIN2	AN4	—	RB0	—	—	—	—	COEXCTRL2	—

.....continued															
PIC32CX-BZ2 SoC		WBZ45 Module			Pad Name	Peripherals									
24032	25048	450	451	WBZ451H		AC	ADC	EIC ⁽⁴⁾	GPIO (1, 2)	QSPI	RTCC	SERCOM	OSC	RF	DEBUG
21	31	10	38		PB1	AC_AIN3	AN5	—	RB1	—	—	—	—	—	—
			38		CPS	—	—	—	—	—	—	—	—	—	—
	32		37	37	PB2	AC_AIN0	AN6	—	RB2	—	—	—	—	—	—
	33		5	5	PB3	AC_AIN1	AN7	—	RB3	—	—	—	—	—	—
22	34	11	39	39	PB4	—	AN0	INT0 ⁽¹¹⁾	RB4	—	—	—	—	FECTRL3	TRACEDATA3
23	35	9	6	6	PB5	—	AN1	—	RB5	—	—	—	—	FECTRL4	TRACEDATA0
24	36	4	7	7	AVDD	—	—	—	—	—	—	—	—	—	—
25	37	12	12	12	PB6	—	ANNO,AN2	—	RB6	—	—	—	—	FECTRL5	TRACEDATA1
26	38	15	13	13	PB7	LVDIN	AN3	—	RB7	—	—	—	—	—	TRACEDATA2, CM4_SWO
	39				VDD	—	—	—	—	—	—	—	—	—	—
27	40	14	15	15	PB9	—	—	—	RB9	—	—	—	—	—	CM4_SWDIO
28	41	13	14	14	PB8	—	—	—	RB8	—	—	—	—	—	CM4_SWCLK
29 ⁽⁶⁾	42	6	16	16	PA4	—	—	—	RA4	—	RTC_OUT	SERCOM0_PAD3	—	—	—
	43		17	17	PB10	—	—	—	RB10	QSPI_CS	—	—	—	—	—
	44		18	18	PB11	—	—	—	RB11	QSPI_SCK	—	—	—	—	—
29	45		9	9	PA11	—	—	—	RA11 ⁽⁵⁾	—	—	—	SOSCI	—	—
30	46		10	10	PA12	—	—	—	RA12 ⁽⁵⁾	—	—	—	SOSCO	—	—
30 ⁽⁶⁾	47	7	36	36	PA3	—	—	—	RA3	—	RTC_IN0	SERCOM0_PAD2	SCLKI	—	—
31	48				PMU_VDD	—	—	—	—	—	—	—	—	—	—
		26, 28, 29, 30			NC	—	—	—	—	—	—	—	—	—	—

Notes:

1. All GPIOs (RAn and RBn) can be used by remappable peripherals via PPS.
2. All GPIOs (RAn and RBn) can be used as I/O Change Notification (IOCA_n and IOCB_n) except RA11 and RA12. Please refer to Table 6-13 Port A Register Map for RA11 and RA12. CN register.
3. The metal paddle at the bottom of the device must be connected to system ground.
4. These peripherals have signals that are only available via the PPS remappable pins.
5. This pin can be used as Input only pin if not using SOSC.
6. For 24032 only, pin 29 and pin 30 act as GPIO PA4/PA3 respectively ONLY if CFGCON0.GPSOSCE = 0. If CFGCON0.GPSOSCE = 1, these pins are SOSC 32 kHz crystal inputs OR as digital input only RA11/RA12 respectively.
7. For 25048 only, pin 29 and pin 30 can be configured as PA11 and PA12 by setting up CFGCON2.SOSCSEL = 0.
8. External resistor used to set internal reference current of the SOC.
9. Disable trace data output for SWD or 4-wire trace output are shared. To disable trace data output, use CFGCON0.SWOEN=0. To disable 4-wire trace output, use CFGCON0.TROEN=0.
10. INT0 can be used as a wake-up source from Deep Sleep or Extreme Deep Sleep Low Power modes, as well as an ADC trigger source. The INT0 can be configured using Configuration Control Register 0 (CFGCON0). INT0 functionality on PB4 cannot be remapped using PPS. The software

SDK and operational stacks provided by Microchip handles the operation of INT0 as a wake-up source in Deep Sleep Low Power Mode.

11. These I/O pins are 5.5V tolerant: NMCLR, PA0, PA1, PA2, PA4, PA5, PA6, PA7, PA8, PA9, PA10, PA13, PA14, PB10, PB11, PB12, PB13. All other I/O pins are 3.3V tolerant.

6. I/O Ports and Peripheral Pin Select (PPS)

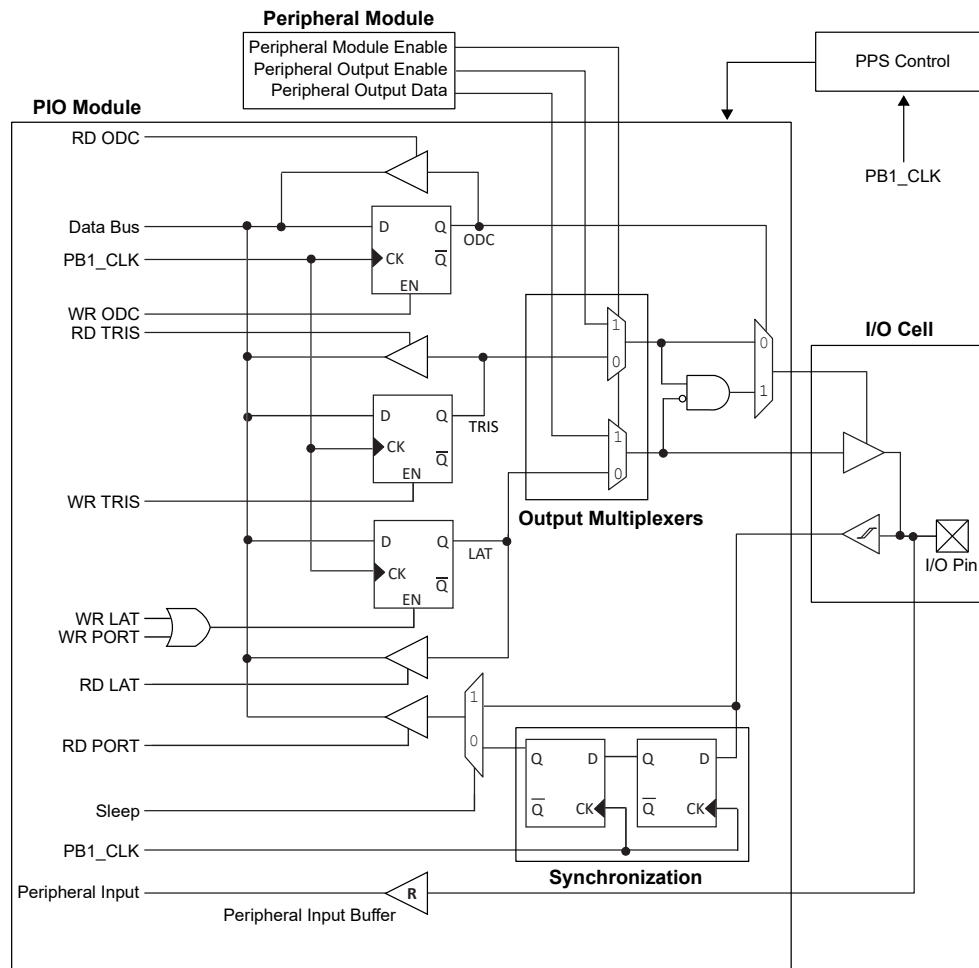
General purpose I/O (GPIO) pins allow the PIC32CX-BZ2 devices to monitor and control other devices. To add flexibility and functionality, some pins are multiplexed with alternate function(s). These functions depend on which peripheral features are on the device. In general, when a peripheral is functioning, that pin may not be used as a general purpose I/O pin. For more details on pin multiplexing, see *GPIO Pins/PPS Functions* from Related Links. There are default priorities for each GPIO pin as well. For details on priorities, see *Function Priority for Device Pins* from Related Links. It must be noted that 'fuse' values stored in NVR memory can be used to alter the power-on default function for certain GPIO pins. See *System Configuration Registers (CFG)* from Related Links.

Some of the key features of the I/O ports are:

- Individual output pin open-drain enable/disable
- Individual input pin weak pull-up and pull-down
- Monitor selective inputs and generate interrupt when change in pin state is detected
- Operation during Sleep and Idle modes
 - Fast bit manipulation using CLR, SET and INV registers
 - Slew rate control

The following figure illustrates a block diagram of a typical multiplexed I/O port.

Figure 6-1. Typical Multiplexed Port Structure Block Diagram



Related Links

[4.2.5.1. GPIO Pins/PPS Functions](#)

[6.3. Function Priority for Device Pins](#)

[18. System Configuration and Register Locking \(CFG\)](#)

6.1 Control Registers

Before reading and writing any I/O port, the desired pin(s) must be properly configured for the application. Each I/O port has nine registers directly associated with the operation of the port and one control register. Each I/O port pin has a corresponding bit in these registers. Throughout this section, the letter 'x', denotes any or all port module instances. For example, TRISx represents TRISA or TRISB. Any bit and its associated data and control registers that is not valid for a particular device will be disabled and will read as zeros.

6.1.1 Configuring Tri-State Functions (TRISx)

The TRISx registers configure the data direction flow through port I/O pins. The TRISx register bits determine whether a PORTx I/O pin is an input or an output:

- If a data direction bit is '1', the corresponding I/O port pin is an input.
- If a data direction bit is '0', the corresponding I/O port pin is an output.
- A read from a TRISx register reads the last value written to that register.

- All I/O port pins are defined as inputs after a Power-on Reset (POR).

6.1.2 Configuring Port Functions (PORTx)

The PORTx registers allow I/O pins to be accessed:

- A write to a PORTx register writes to the corresponding LATx register (PORTx data latch). Those I/O port pin(s) configured as outputs are updated.
- A write to a PORTx register is effectively the same as a write to a LATx register.
- A read from a PORTx register reads the synchronized signal applied to the port I/O pins.

6.1.3 Configuring Latch Functions (LATx)

The LATx registers (PORTx data latch) hold data written to port I/O pins:

- A write to a LATx register latches data to corresponding port I/O pins. Those I/O port pins configured as outputs are updated.
- A read from a LATx register reads the data held in the PORTx data latch, not from the port I/O pins.

6.1.4 Open-Drain Configuration (ODCx)

Each I/O pin can be individually configured for either normal digital output or open-drain output. This is controlled by the open-drain control register, ODCx, associated with each I/O pin. If the ODCx bit for an I/O pin is a '1', the pin acts as an open-drain output. If the ODCx bit for an I/O pin is a '0', the pin is configured for a normal digital output (the ODCx bit is valid only for output pins). After a Reset, the status of all the bits of the ODCx register is set to '0'.

The maximum open-drain voltage allowed is the same as the maximum V_{IH} specification. The ODCx register setting functions in all of the I/O modes, allowing the output to behave as an open-drain even if a peripheral is controlling the pin. Although, the user may achieve the same result by manipulating the corresponding LATx and TRISx bits, this procedure does not allow the peripheral to operate in the Open-Drain mode (except for the default operation of the I²C pins). I²C pins are already open-drain pins; therefore, the ODCx settings do not influence the I²C pins.

6.1.5 Configuring Analog and Digital Port Pins (ANSELx)

The ANSELx register controls the operation of the analog port pins. The port pins that are to function as analog inputs must have their corresponding ANSEL and TRISx bits set. To use port pins for I/O functionality with digital modules, such as Timers, SERCOMs and so on, the corresponding ANSELx bit must be cleared. The ANSELx register has a default value of 0xFFFF; therefore, all pins that share analog functions are, by default, analog and not digital.

If the TRISx bit is cleared (output) while the ANSELx bit is set, the digital output level (V_{OH} or V_{OL}) is converted by an analog peripheral, such as the ADC module or the comparator module. When the PORTx register is read, all pins configured as analog input channels are read as cleared (a low-level). Pins configured as digital inputs do not convert an analog input. Analog levels on any pin defined as a digital input (including the ANx pins) can cause the input buffer to consume current that exceeds the device specifications.

6.1.6 Input Change Notification (CN)

The Input Change Notification (CN) function of the I/O ports allows PIC32CX-BZ2 devices to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This feature can detect input change of states even in the Sleep mode, when the clocks are disabled.

The following control registers are associated with the CN functionality of each I/O port:

- Change Notice Pull-up Enable (CNPUEx)
- Change Notice Pull-down Enable (CNPDX)

- Change Notice Control (CNCONx)
- Change Notice Enable (CNENx/CNNEx)
- Change Notice Status (CNSTATx/CNFX) or the positive edge control

Each I/O pin also has a weak pull-up and a weak pull-down connected to it. The pull-ups act as a current source or sink source connected to the pin and eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups and pull-downs are enabled separately using the CNPUEx and the CNPDx registers, which contain the control bits for each of the pins. Setting any of the control bits enables the weak pull-ups and/or pull-downs for the corresponding pins.

Note: Pull-ups and pull-downs on change notification pins must always be disabled when the port pin is configured as a digital output

The CNCONx registers provide change notice control.

The CNENx/CNNEx registers contain the CN interrupt enable control bits for each of the input pins. Setting any of these bits enables a CN interrupt for the corresponding pins. CNENx enables a mismatch CN interrupt condition when EDGEDETECT is not set. When EDGEDETECT is set, CNNEx controls the negative edge while CNENx controls the positive. On devices that do not have EDGEDETECT, this CN logic acts as if EDGEDETECT is not set.

The CNSTATx/CNFX registers indicate whether a change occurred on the corresponding pin since the last read of the PORTx bit. The CNFX registers indicate the type of change that occurred.

6.1.7 Registers for Peripheral Pin Select

The Peripheral Pin Select [pin name]R register ([pin name]R) and the Peripheral Pin Select Output (RPnR) register (RPnR) provide the control bits for the peripheral pin select input and output. See [pin name]R and RPnR from Related Links.

Related Links

[6.2.7.4. \[pin name\]R](#)

[6.2.9.1. RPnR](#)

6.1.8 Slew Rate Control

Some I/O pins can be configured for various types of slew rate control on its associated port. This is controlled by the slew rate control bits in the SRCON1x and SRCON0x registers that are associated with each I/O port. The slew rate control is configured using the corresponding bit in each register, as shown in the following table.

As an example, writing 0x0001, 0x0000 to SRCON1A and SRCON0A, respectively, can enable slew rate control on the RA0 pin and sets the slew rate to the slow edge rate.

Note: Slew rate control bits must not be enabled with pad configured as an input.

Table 6-1. Slew Rate Control Bit Settings⁽¹⁾

SRCON1x	SRCON0x	Description
1	1	Slew rate control is enabled and is set to the slowest edge rate
1	0	Slew rate control is enabled and is set to the slow edge rate
0	1	Slew rate control is enabled and is set to the medium edge rate
0	0	Slew rate control is disabled and is set to the fastest edge rate
1. By default, all the port pins are set to the fastest edge rate.		

6.1.9 CLR, SET and INV Registers

Every I/O module register has corresponding SET, CLR and INV registers, which provide atomic bit manipulations. As the name of the registers imply, a value written to a SET, CLR or INV register

effectively performs the implied operation, but only on the corresponding base register and only bits specified as '1' are modified. Bits specified as '0' are not modified. For example,

- Writing 0x0001 to the TRISASET register sets only bit 0 in the base register TRISA
- Writing 0x0020 to the PORTBCLR register clears only bit 5 in the base register PORTB
- Writing 0x9000 to the LATAINV register inverts only bits 15 and 12 in the base register LATA

Reading the SET, CLR and INV registers returns an undefined value. To see the influences of a write operation to a SET, CLR or INV register, the base register must be read instead.

A typical method to toggle an I/O pin requires a read-modify-write operation performed on a PORTx register in the software. For example, a read from a PORTx register, mask and modify the desired output bit or bits, and write the resulting value back to the PORTx register. This method is vulnerable to a read-modify-write issue where the port value may change after it is read and before the modified data can be written back, thus, changing the previous state. This method also requires more instructions.

A more efficient and atomic method uses the PORTxINV register. A write to the PORTxINV register effectively performs a read-modify-write operation on the target base register, equivalent to the software operation described previously; however, it is done in the hardware. To toggle an I/O pin using this method, a '1' is written to the corresponding bit in the PORTxINV register. This operation reads the PORTx register, inverts only those bits specified as '1', and writes the resulting value to the LATx register, thus, toggling the corresponding I/O pins all in a single atomic instruction cycle. PORTAINV = 0x0001.

6.2 Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. The challenge is even greater on low pin-count devices. In an application where more than one peripheral needs to be assigned to a single pin, inconvenient workarounds in application code or a complete redesign may be the only option.

The PPS configuration provides an alternative to these choices by enabling peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, the users can better modify the device to their entire application, rather than trimming the application to fit the device.

This feature operates over a fixed subset of digital I/O pins. The users may independently map the input and/or output of most digital peripherals to these I/O pins. The PPS configuration is performed in the software and generally does not require the device to be reprogrammed. The hardware safeguards that prevent accidental or spurious changes to the peripheral mapping are included once the PPS configuration is established.

In PPS mode, Maximum peripheral clock frequency = Direct mode clock frequency/2.

Note: Direct Mode is a mode in which peripherals are running based on Function Priority for Pins and not using PPS.

6.2.1 Re-Mappable Pin Groupings

The re-mappable pins, as well as the available input and output functions, are divided into four groups. The re-mappable pins of group k may be assigned pin functions only from group k (k = 1,2,3,4). The pins used by each peripheral are spread across all four groups when possible to maximize flexibility.

6.2.2 Enabling Remap Pins

Each remap pin (RPx) must be enabled by disabling all higher priority pin functions on that pin before it can be used. Typically, all functions other than GPIO are considered higher priority than remap pins.

6.2.3 RP Register Protection

The <INPUT>R and RPxxR registers are implemented with two levels of protection:

- I/O Lock Feature – All PPS registers may only be written while CFGCON0.IOLOCK = 0; once the IOLOCK is set, the registers cannot be written.
- IOLOCK Protection – The state of the IOLOCK bit can only be changed once it is unlocked using the CFGCON0.CFGLOCK[1:0] register.

These features prevent the RP registers from being inadvertently written during normal operation because changing the pinout functionality may have detrimental system-level outcome.

6.2.4 Available Pins

The number of available pins is dependent on the particular device and its pin count. Pins that support the PPS feature include the designation “RPn” in their full pin designation, where:

- RP – Designates a remappable peripheral
- n – Remappable port number

6.2.5 Available Peripherals

The peripherals managed by the PPS are all digital-only peripherals. These include general serial communications (SERCOM), general purpose timer clock inputs, timer-related peripherals (input capture and output compare), interrupt-on-change inputs and reference clocks (input and output).

In comparison, some digital-only peripheral modules are never included in the PPS feature. This is because the peripheral's function requires special I/O circuitry on a specific port and cannot be easily connected to multiple pins. These modules include I²C among others. A similar requirement excludes all modules with analog inputs, such as the ADC.

A key difference between remappable and non-remappable peripherals is that remappable peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non-remappable peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

When a remappable peripheral is active on a given I/O pin, it takes priority over all other digital I/O and digital communication peripherals associated with the pin. Priority is given regardless of the type of peripheral that is mapped. Remappable peripherals never take priority over any analog functions associated with the pin.

6.2.6 Controlling PPS

The PPS features are controlled through two sets of SFRs: one to map peripheral inputs and another to map outputs. They are separately controlled; therefore, a particular peripheral's input and output (if the peripheral has both) can be placed on any selectable function pin without constraint.

The association of a peripheral to a peripheral-selectable pin is handled in two different ways, depending on whether an input or output is mapped.

6.2.7 Remappable Inputs

When configuring a PAD for a new peripheral functionality, the existing PPS configuration must be cleared prior to using the same PAD for the different peripheral functionality.

6.2.7.1 Enabling Remappable Peripheral Inputs

With PPS, each remappable input pin function (EXTINT0, SERCOM0_PAD3 and so on) is assigned to be driven from a specific device pin by programming the corresponding <INPUT>R[3:0] register (meaning, EXTINT0R[3:0], SCOM0P3R[3:0], and so on) with a value defined in the *Input Pin Selection Group 1*, *Input Pin Selection Group 2*, *Input Pin Selection Group 3*, *Input Pin Selection Group 4* tables and [*pin name*]R register. See these tables in the *Remappable Input Example* and [*pin name*]R from Related Links.

Assigning a remappable input pin function does not automatically enable the digital input buffer on the pin. The buffer must be enabled for each remap pin (RPx) by disabling all higher priority pin functions on that pin. Typically, all functions other than GPIO are considered higher priority than remap pins. See *Function Priority for Device Pins* for the list and priority of pin functions on each pin from Related Links.

The mapping is dynamic; therefore, in order to avoid glitching outputs, the user is responsible for turning off the appropriate peripherals before remapping the pin functions associated with that peripheral. On Reset, all inputs are mapped to a default value and all outputs are disabled; therefore, the mapping may safely be performed after any device Reset.

Related Links

[6.2.7.3. Remappable Input Example](#)

[6.2.7.4. \[pin name\]R](#)

6.2.7.2 Remappable Input Priority

Only a single pin can be selected for any of the remappable peripheral inputs; therefore, priority encoding is not needed for RP inputs.

Note: A remappable input function does not have any control over the output of the RPx pin.

In this way, it is possible to drive a remappable output function on an RPx pin and a completely different remappable input function on the same pin. This can be useful, for instance, in driving an EVSYS output back into a Timer clock or gating input by assigning both functions to the same remap pin.

Note: To allow flexibility on 32 or 48-pin devices, the same input functions are repeated in multiple groups. Therefore, in order to differentiate between them, the “Off” code is provided in the *Input Pin Selection Group 1*, *Input Pin Selection Group 2*, *Input Pin Selection Group 3* and *Input Pin Selection Group 4* tables. See these tables in the *Remappable Input Example* from Related Links.

The software must ensure that the unused group register offset of a (repeated) input function is programmed to 4'h0 for proper operation. Failure to do so will lead to unknown behavior.

Related Links

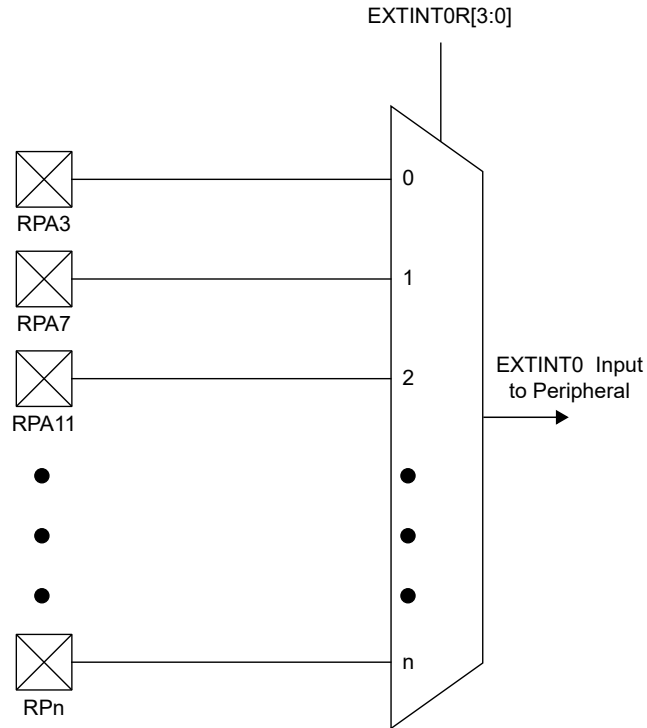
[6.2.7.3. Remappable Input Example](#)

6.2.7.3 Remappable Input Example

For example, the following figure illustrates the remappable pin selection for the EXTINT0 input. To remap the EXTINT0 input to a particular pin, the EXTINT0R remap register must be programmed. EXTINT0 is in group 1; therefore, it can be mapped to any pin that is in group 1 (RPA3, RPA7, RPA9, RPA11, RPB0, RPB4, RPB8 and so on).

To map it to RPB0, program the value 4 (4'b0100) into the EXINTR0R SFR register. See the following *Input Pin Selection Group 1* table.

Figure 6-2. EXTINT0 Remappable Pin Selection



Note: For input only, the PPS functionality does not have priority over the TRISx settings. Therefore, when configuring the RPn pin for input, the corresponding bit in the TRISx register must also be configured for input (set to '1').

Table 6-2. Input Pin Selection Group 1

Peripheral Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPn Pin Selection
EXTINT0	EXTINT0R	EXTINT0[3:0]	0000 = OFF
SERCOM0_PAD3	SCOM0P3R	SCOM0P3R[3:0]	0001 = RPA3
SERCOM1_PAD2	SCOM1P2R	SCOM1P2R[3:0]	0010 = RPA7
SERCOM2_PAD1	SCOM2P1R	SCOM2P1R[3:0]	0011 = RPA11
SERCOM3_PAD0	SCOM3P0R	SCOM3P0R[3:0]	0100 = RPB0
QSKK	QSKKR	QSKKR[3:0]	0101 = RPB4
QD1	QD1R	QD1R[3:0]	0110 = RPB8
REFI	REFIR	REFIR[3:0]	0111 = RPB12
CCLIN0	CCLIN0R	CCLIN0R[3:0]	1000 = RPA2
CCLIN3	CCLIN3R	CCLIN3R[3:0]	1001 = RPA6
TC0_WO0G1	TC0WO0G1R	TC0WO0G1R[3:0]	1010 = RPA10
TC1_WO0G1	TC1WO0G1R	TC1WO0G1R[3:0]	1011 = RPA14
TC2_WO0G1	TC2WO0G1R	TC2WO0G1R[3:0]	1100 = RPB3
TC3_WO0G1	TC3WO0G1R	TC3WO0G1R[3:0]	1101 = RPB7
			1110 = RPB11
			1111 = RPA9

Table 6-3. Input Pin Selection Group 2

Peripheral Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPN Pin Selection
EXTINT1	EXTINT1R	EXTINT1R[3:0]	0000 = OFF
SERCOM0_PAD0	SCOM0P0R	SCOM0P0R[3:0]	0001 = RPA4
SERCOM1_PAD3	SCOM1P3R	SCOM1P3R[3:0]	0010 = RPA8
SERCOM2_PAD2	SCOM2P2R	SCOM2P2R[3:0]	0011 = RPA12
SERCOM3_PAD1	SCOM3P1R	SCOM3P1R[3:0]	0100 = RPB1
QD2	QD2R	QD2R[3:0]	0101 = RPB5
CCLIN1	CCLIN1R	CCLIN1R[3:0]	0110 = RPB9
CCLIN4	CCLIN4R	CCLIN4R[3:0]	0111 = RPB13
TC0_WO0G2	TC0WO0G2R	TC0WO0G2R[3:0]	1000 = RPA3
TC1_WO1G2	TC1WO1G2R	TC1WO1G2R[3:0]	1001 = RPA7
TC2_WO1G2	TC2WO1G1R	TC2WO1G1R[3:0]	1010 = RPA11
TC3_WO1G2	TC3WO1G1R	TC3WO1G1R[3:0]	1011 = RPB0
			1100 = RPB4
			1101 = RPB8
			1110 = RPB12
			1111 = RPA0

Table 6-4. Input Pin Selection Group 3

Peripheral Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPN Pin Selection
EXTINT2	EXTINT2R	EXTINT2R[3:0]	0000 = OFF
SERCOM0_PAD1	SCOM0P1R	SCOM0P1R[3:0]	0001 = RPA5
SERCOM1_PAD0	SCOM1P0R	SCOM1P0R[3:0]	0010 = RPA9
SERCOM2_PAD3	SCOM2P3R	SCOM2P3R[3:0]	0011 = RPA13
SERCOM3_PAD2	SCOM3P2R	SCOM3P2R[3:0]	0100 = RPB2
QD3	QD3R	QD3R[3:0]	0101 = RPB6
CCLIN2	CCLIN2R	CCLIN2R[3:0]	0110 = RPB10
CCLIN5	CCLIN5R	CCLIN5R[3:0]	0111 = RPA0
TC0_WO1G3	TC0WO1G3R	TC0WO1G3R[3:0]	1000 = RPA4
TC2_WO0G3	TC2WO0G3R	TC2WO0G3R[3:0]	1001 = RPA8
TC3_WO0G3	TC3WO0G3R	TC3WO0G3R[3:0]	1010 = RPA12
			1011 = RPB1
			1100 = RPB5
			1101 = RPB9
			1110 = RPB13
			1111 = RPA1

Table 6-5. Input Pin Selection Group 4

Peripheral Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPN Pin Selection
EXTINT3	EXTINT3R	EXTINT3R[3:0]	0000 = OFF
NMI	NMIR	NMIR[3:0]	0001 = RPA6
SERCOM0_PAD2	SCOM0P2R	SCOM0P2R[3:0]	0010 = RPA10
SERCOM1_PAD1	SCOM1P1R	SCOM1P1R[3:0]	0011 = RPA14
SERCOM2_PAD0	SCOM2P0R	SCOM2P0R[3:0]	0100 = RPB3
SERCOM3_PAD3	SCOM3P3R	SCOM3P3R[3:0]	0101 = RPB7
QD0	QD0R	QD0R[3:0]	0110 = RPB11
TC0_WO1G4	TC0WO1G4R	TC0WO1G4R[3:0]	0111 = RPA1
TC2_WO1G4	TC2WO1G4R	TC2WO1G4R[3:0]	1000 = RPA5
TC3_WO1G4	TC3WO1G4R	TC3WO1G4R[3:0]	1001 = RPA9
			1010 = RPA13
			1011 = RPB2
			1100 = RPB6
			1101 = RPB10
			1110 = RPA8
			1111 = RPA2

6.2.7.4 Peripheral Pin Select Input Register

Name: *[pin name]R*
Offset: See the following Note
Reset: 0x00
Property: -

Notes:

- For the Offset address, see *Peripheral Pin Select Input Registers* table in the *I/O Ports Control Registers* from Related Links.
- Register values can only be changed if the IOLOCK Configuration bit (CFGCON0[13]) = 0.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					[pin name]R[3:0]			
Access					R/W-0	R/W-0	R/W-0	R/W-0
Reset					0	0	0	0

Bits 3:0 – [pin name]R[3:0] Peripheral Pin Select Input bits

Where *[pin name]* refers to the pins that are used to configure peripheral input mapping. See *Input Pin Selection Group 1*, *Input Pin Selection Group 2*, *Input Pin Selection Group 3* and *Input Pin Selection Group 4* tables in the *Remappable Input Example* for input pin selection values from Related Links.

Note: This field is only writable when CFGCON0.IOLOCK = 0.

Related Links

[6.2.7.3. Remappable Input Example](#)
[6.4. I/O Ports Control Registers](#)

6.2.8 Output Mapping

The remappable pin output assigns a peripheral output function to an output pin. Once the group for the output pin is identified, see the following table, which shows the peripheral output functions and its group.

Each remappable output can be programmed to an output function that is from its same output group number. As an example, if RPA0 is part of GROUP2, then it can be programmed to have any GROUP2 output function on its pin. Therefore, for a given output peripheral signal, the user must first choose which remappable pin to use, choose a Group number for that pin and, then, program the control registers for that pin. For example, RPA<0-10, 13, 14> G<1, 2, 3, 4> R or RPB<0-13> G<1, 2, 3, 4>R. See *Remappable Output Pin Configuration – Group1*, *Remappable Output Pin Configuration – Group2*, *Remappable Output Pin Configuration – Group3*, and *Remappable Output Pin Configuration – Group4* tables in the *Pin Output RP Registers* from Related Links.

The rules for which group belongs to which pin must be followed, such that multiple peripherals are not driving the same pin from different groups. For instance, pin RPA0 (PA0) as an output belongs to Group2 and Group3. If the peripheral driving the signal to RPA0 is coming from Group2, the software must ensure that all Group3 signals for RPA0 are disabled with an 'OFF' value in the corresponding RPA0G3R control register.

A null output is associated with the output register reset value of '0'. This is done to ensure that remappable outputs remain disconnected from all output pins, by default.

Table 6-6. PPS Output Groups

Group1	Group2	Group3	Group4
SERCOM0_PAD3	SERCOM0_PAD0	SERCOM0_PAD1	SERCOM0_PAD2
SERCOM0_PAD2	SERCOM0_PAD3	SERCOM0_PAD0	SERCOM0_PAD1
SERCOM0_PAD1	SERCOM0_PAD2	SERCOM0_PAD3	SERCOM0_PAD0
SERCOM1_PAD0	SERCOM1_PAD1	SERCOM1_PAD2	SERCOM1_PAD3
SERCOM1_PAD2 ⁽¹⁾	SERCOM1_PAD3 ⁽¹⁾	SERCOM1_PAD0	SERCOM1_PAD1
SERCOM1_PAD1	SERCOM1_PAD2 ⁽¹⁾	SERCOM1_PAD3	SERCOM1_PAD0
SERCOM2_PAD0	SERCOM2_PAD1	SERCOM2_PAD2	SERCOM2_PAD3
SERCOM2_PAD1	SERCOM2_PAD2	SERCOM2_PAD3	SERCOM2_PAD0
SERCOM3_PAD0	SERCOM3_PAD1	SERCOM3_PAD2	SERCOM3_PAD3
SERCOM3_PAD3	SERCOM3_PAD0	SERCOM3_PAD1	SERCOM3_PAD2
TCC0_WO0	TCC0_WO1	TCC0_WO2	TCC0_WO3
TCC0_WO4	TCC0_WO5	TCC0_WO0	TCC0_WO1
TCC0_WO2	TCC0_WO3	TCC0_WO4	TCC0_WO5
TCC1_WO0	TCC1_WO1	TCC1_WO2	TCC1_WO3
TCC1_WO4	TCC1_WO5	TCC1_WO0	TCC1_WO1
TCC1_WO2	TCC1_WO3	TCC1_WO4	TCC1_WO5
TCC2_WO0	TCC2_WO1	TCC2_WO0	TCC2_WO1
TC0_WO1	TC0_WO1	TC0_WO0	TC0_WO0
REFO1	REFO2	REFO3	REFO4
TC1_WO0	TC1_WO1	TC1_WO0	TC1_WO1
TC2_WO0	TC2_WO1	TC2_WO0	TC2_WO1
TC3_WO0	TC3_WO1	TC3_WO0	TC3_WO1
QSPI_SCK	QSPI_SCK	QSPI_SCK	QSPI_SCK
QSPI_CS	QSPI_CS	QSPI_CS	QSPI_CS
QSPI_DATA3	QSPI_DATA0	QSPI_DATA1	QSPI_DATA2
QSPI_DATA2	QSPI_DATA3	QSPI_DATA0	QSPI_DATA1
QSPI_DATA1	QSPI_DATA2	QSPI_DATA3	QSPI_DATA0
CCL_OUT0	CCL_OUT1	CCL_OUT0	CCL_OUT1
Note:			
1. Not applicable for WBZ451H users.			

Related Links

6.2.9. Pin Output RP Registers

6.2.9 Pin Output RP Registers

Register *RPnR* shows the RP Remap Register format for output functions. See *RPnR* from Related Links. Each RP pin has a 4-bit field that can be assigned to the desired output function. See the following tables for a complete list of output function values and associated register names.

The mapping is dynamic; therefore, to avoid glitching outputs, the user must ensure that the appropriate peripherals are turned off before remapping the functions. On Reset, all inputs are mapped to a default value and all outputs are disabled; therefore, the mapping must be performed after any device Reset.

Figure 6-3. Example Multiplexing of Remappable Output Signal for RPA0 (Map Output Function to Pin)

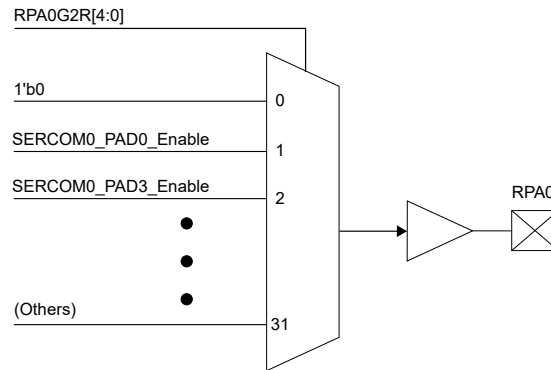


Table 6-7. Remappable Output Pin Configuration – Group1

Output Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPh Pin Selection
RPA2	RPA2G1R	RPA2G1R[4:0]	00000 = OFF
RPA3	RPA3G1R	RPA3G1R[4:0]	00001 = SERCOM0_PAD3
RPA5	RPA5G1R	RPA5G1R[4:0]	00010 = SERCOM0_PAD2
RPA6	RPA6G1R	RPA6G1R[4:0]	00011 = SERCOM0_PAD1
RPA7	RPA7G1R	RPA7G1R[4:0]	00100 = SERCOM1_PAD0
RPA9 ⁽¹⁾	RPA9G1R	RPA9G1R[4:0]	00101 = SERCOM1_PAD2
RPA10 ⁽¹⁾	RPA10G1R	RPA10G1R[4:0]	00110 = SERCOM1_PAD1
RPA14	RPA14G1R	RPA14G1R[4:0]	00111 = SERCOM2_PAD0
RPB0	RPB0G1R	RPB0G1R[4:0]	01000 = SERCOM2_PAD1
			01001 = SERCOM3_PAD0
			01010 = SERCOM3_PAD3
			01011 = TCC0_WO0
			01100 = TCC0_WO4
			01101 = TCC0_WO2
			01110 = TCC1_WO0
			01111 = TCC1_WO4
			10000 = TCC1_WO2
			10001 = TCC2_WO0
			10010 = TC0_WO1
			10011 = REFO1
			10100 = TC1_WO0
			10101 = TC2_WO0
			10110 = TC3_WO0
			10111 = QSCK
			11000 = QCS
			11001 = QD3
			11010 = QD2
			11011 = QD1
			11100 = CCLOUT0
			11101 = RESERVED
			11110 = RESERVED
			11111 = RESERVED
RPB3	RPB3G1R	RPB3G1R[4:0]	
RPB4	RPB4G1R	RPB4G1R[4:0]	
RPB7	RPB7G1R	RPB7G1R[4:0]	
RPB8	RPB8G1R	RPB8G1R[4:0]	
RPB11	RPB11G1R	RPB11G1R[4:0]	
RPB12	RPB12G1R	RPB12G1R[4:0]	
Note:			
1. Not applicable for WBZ451H users.			

Table 6-8. Remappable Output Pin Configuration – Group2

Output Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPin Pin Selection
RPA0	RPA0G2R	RPA0G2R[4:0]	00000 = OFF
RPA3	RPA3G2R	RPA3G2R[4:0]	00001 = SERCOM0_PAD0
RPA4	RPA4G2R	RPA4G2R[4:0]	00010 = SERCOM0_PAD3
RPA6	RPA6G2R	RPA6G2R[4:0]	00011 = SERCOM0_PAD2
RPA7	RPA7G2R	RPA7G2R[4:0]	00100 = SERCOM1_PAD1
RPA8	RPA8G2R	RPA8G2R[4:0]	00101 = SERCOM1_PAD3
RPA8	RPA8G2R	RPA8G2R[4:0]	00110 = SERCOM1_PAD2
RPB0	RPB0G2R	RPB0G2R[4:0]	00111 = SERCOM2_PAD1
RPB1 ⁽¹⁾	RPB1G2R	RPB1G2R[4:0]	01000 = SERCOM2_PAD2
RPB4	RPB4G2R	RPB4G2R[4:0]	01001 = SERCOM3_PAD1
RPB5	RPB5G2R	RPB5G2R[4:0]	01010 = SERCOM3_PAD0
RPB8	RPB8G2R	RPB8G2R[4:0]	01011 = TCC0_WO1
RPB9	RPB9G2R	RPB9G2R[4:0]	01100 = TCC0_WO5
RPB12	RPB12G2R	RPB12G2R[4:0]	01101 = TCC0_WO3
RPB13	RPB13G2R	RPB13G2R[4:0]	01110 = TCC1_WO1
			01111 = TCC1_WO5
			10000 = TCC1_WO3
			10001 = TCC2_WO1
			10010 = TC0_WO1
			10011 = REFO2
			10100 = TC1_WO1
			10101 = TC2_WO1
			10110 = TC3_WO1
			10111 = QSCK
			11000 = QCS
			11001 = QD0
			11010 = QD3
			11011 = QD2
			11100 = CCLOUT1
			11101 = RESERVED
			11110 = RESERVED
			11111 = RESERVED

Note:

1. Not applicable for WBZ451H users.

Table 6-9. Remappable Output Pin Configuration - Group3

Output Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPN Pin Selection
RPA0	RPA0G3R	RPA0G3R[4:0]	00000 = OFF
RPA1	RPA1G3R	RPA1G3R[4:0]	00001 = SERCOM0_PAD1 00010 = SERCOM0_PAD0
RPA3	RPA3G3R	RPA3G3R[4:0]	00011 = SERCOM0_PAD3
RPA4	RPA4G3R	RPA4G3R[4:0]	00100 = SERCOM1_PAD2
RPA5	RPA5G3R	RPA5G3R[4:0]	00101 = SERCOM1_PAD0
RPA8	RPA8G3R	RPA8G3R[4:0]	00110 = SERCOM1_PAD3
RPA9	RPA9G3R	RPA9G3R[4:0]	00111 = SERCOM2_PAD2
RPA13	RPA13G3R	RPA13G3R[4:0]	01000 = SERCOM2_PAD3
RPB1	RPB1G3R	RPB1GX3R[4:0]	01001 = SERCOM3_PAD2
RPB2	RPB2G3R	RPB2G3R[4:0]	01010 = SERCOM3_PAD1
RPB5	RPB5G3R	RPB5G3R[4:0]	01011 = TCC0_WO2
RPB6	RPB6G3R	RPB6G3R[4:0]	01100 = TCC0_WO0
RPB9	RPB9G3R	RPB9G3R[4:0]	01101 = TCC0_WO4
RPB10	RPB10G3R	RPB10G3R[4:0]	01110 = TCC1_WO2
RPB13	RPB13G3R	RPB13G3R[4:0]	01111 = TCC1_WO0 10000 = TCC1_WO4 10001 = TCC2_WO0 10010 = TC0_WO0 10011 = REFO3 10100 = TC1_WO0 10101 = TC2_WO0 10110 = TC3_WO0 10111 = QSCK 11000 = QCS 11001 = QD1 11010 = QD0 11011 = QD3 11100 = CCLOUT0 11101 = RESERVED 11110 = RESERVED 11111 = RESERVED

Table 6-10. Remappable Output Pin Configuration – Group4

Output Pin	[pin name]R SFR	[pin name]R Value to RPN Pin Selection
RPA1	RPA1G4R	00000 = OFF
RPA2	RPA2G4R	00001 = SERCOM0_PAD2
RPA4	RPA4G4R	00010 = SERCOM0_PAD1
RPA5	RPA5G4R	00011 = SERCOM0_PAD0
RPA6	RPA6G4R	00100 = SERCOM1_PAD3
RPA8	RPA8G4R	00101 = SERCOM1_PAD1
RPA9	RPA9G4R	00110 = SERCOM1_PAD0
RPA10	RPA10G4R	00111 = SERCOM2_PAD3
RPA13	RPA13G4R	01000 = SERCOM2_PAD0
RPA14	RPA14G4R	01001 = SERCOM3_PAD3
RPA14	RPA14G4R	01010 = SERCOM3_PAD2
RPB2	RPB2G4R	01011 = TCC0_WO3
RPB3	RPB3G4R	01100 = TCC0_WO1
RPB6	RPB6G4R	01101 = TCC0_WO5
RPB7	RPB7G4R	01110 = TCC1_WO3
RPB10	RPB10G4R	01111 = TCC1_WO1
RPB11	RPB11G4R	10000 = TCC1_WO5
		10001 = TCC2_WO1
		10010 = TC0_WO0
		10011 = REFO4
		10100 = TC1_WO1
		10101 = TC2_WO1
		10110 = TC3_WO1
		10111 = QSCK
		11000 = QCS
		11001 = QD2
		11010 = QD1
		11011 = QD0
		11100 = CCLOUT1
		11101 = RESERVED
		11110 = RESERVED
		11111 = RESERVED

Related Links

[6.2.9.1. RPNR](#)

6.2.9.1 Peripheral Pin Select Output Register

Name: RPnR
Offset: See the following Note
Reset: 0x0
Property: -

Notes:

1. For the Offset address, see the *Peripheral Pin Select Output Registers* table in the *I/O Ports Control Registers* from Related Links.
2. Register values can only be changed if the IOLOCK Configuration bit (CFGCON0.IOLOCK) = 0.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						RPnR[4:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – RPnR[4:0] Peripheral Pin Select Output Register

Output bits. For output pin selection values, see *Remappable Output Pin Configuration – Group1*, *Remappable Output Pin Configuration – Group2*, *Remappable Output Pin Configuration – Group3*, and *Remappable Output Pin Configuration – Group4* tables in the *Pin Output RP Registers* from Related Links.

Note: This field is only writable, when CFGCON0.IOLOCK = 0.

Related Links

[6.4. I/O Ports Control Registers](#)

6.3 Function Priority for Device Pins

The device pins have an associated priority order where functionality is exhibited on each pin. This priority order impacts the availability of PPS functionality. For example, if SERCOM0 is enabled with outputs chosen to be High Speed/Direct mode in the DEVCFG1 fuses (bit 17), pins PA3, PA4, PA5 and PA6 are given priority to be used as SERCOM0 pins instead of GPIO/PPS pins. See the following tables for the priority in which functions are brought out on each device pin.

Table 6-11. Priority for Device Pins PAn (n=0-14)

Pin Name ⁽¹⁾	Functions in Priority Order	Reference Peripheral	Pin Number (48-pin)	I/O	Type
pa0	QSPI_DATA2	QSPI	4	I/O	DIG/ST
	RTC_IN3	RTCC		I	ST/STMV
	RPA0	PPS		I/O	DIG/ST
	IOCA0	Change Notification		I	ST
	RA0	GPIO		I/O	DIG/ST
pa1 ⁽²⁾	QSPI_DATA3	QSPI	5	I/O	DIG/ST
	AC_CMP1	Analog Comparator		O	DIG
	RTC_IN2	RTCC		I	ST/STMV
	RPA1	PPS		I/O	DIG/ST
	IOCA1	Change Notification		I	ST
	RA1	GPIO		I/O	DIG/ST
pa2	AC_CMP0	Analog Comparator	6	O	DIG
	RTC_IN1	RTCC		I	ST/STMV
	RPA2	PPS		I/O	DIG/ST
	IOCA2	Change Notification		I	ST
	RA2	GPIO		I/O	DIG/ST
pa3	SCLKI	Secondary Oscillator - Digital	47	I	ST/STMV
	SERCOM0_PAD2	SERCOM0		I/O	DIG/ST
	RTC_IN0	RTCC		I	ST/STMV
	RPA3	PPS		I/O	DIG/ST
	IOCA3	Change Notification		I	ST
	RA3	GPIO		I/O	DIG/ST
pa4	SERCOM0_PAD3	SERCOM0	42	I/O	DIG/ST
	RTC_OUT	RTCC		O	DIGMV
	RPA4	PPS		I/O	DIG/ST
	IOCA4	Change Notification		I	ST
	RA4	GPIO		I/O	DIG/ST
pa5	SERCOM0_PAD0	SERCOM0	7	I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	RPA5	PPS		I/O	DIG/ST
	IOCA5	Change Notification		I	ST
	RA5	GPIO		I/O	DIG/ST
pa6	PGC2ENTRY	DEBUG	9	I	ST
	SERCOM0_PAD1	SERCOM0		I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	AC_CMP1_ALT	Analog Comparator		O	DIG
	RPA6	PPS		I/O	DIG/ST
	IOCA6	Change Notification		I	ST
	RA6	GPIO		I/O	DIG/ST
pa7	TRACECLK	DEBUG	10	I	DIG
	SERCOM1_PAD0	SERCOM1		I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	RPA7	PPS		I/O	DIG/ST
	IOCA7	Change Notification		I	ST
	RA7	GPIO		I/O	DIG/ST

.....continued

Pin Name ⁽¹⁾	Functions in Priority Order	Reference Peripheral	Pin Number (48-pin)	I/O	Type
pa8	PGD2ENTRY	DEBUG	11	I	ST
	FECTRL0	RF Radio		O	DIG
	SERCOM1_PAD1	SERCOM1		I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	RPA8	PPS		I/O	DIG/ST
	IOCA8	Change Notification		I	ST
	RA8	GPIO		I/O	DIG/ST
pa9	FECTRL1	RF Radio	12	O	DIG
	SERCOM1_PAD2	SERCOM1		I/O	DIG/ST
	RTC_IN0_ALT	RTCC		I	ST/STMV
	RPA9	PPS		I/O	DIG/ST
	IOCA9	Change Notification		I	ST
	RA9	GPIO		I/O	DIG/ST
pa10 ⁽²⁾	FECTRL2	RF Radio	13	O	DIG
	SERCOM1_PAD3	SERCOM1		I/O	DIG/ST
	RTC_OUT_ALT	RTCC		O	DIGMV
	RPA10	PPS		I/O	DIG/ST
	IOCA10	Change Notification		I	ST
	RA10	GPIO		I/O	DIG/ST
pa11	SOSCI	Secondary Oscillator	45	—	—
pa12	SOSCO	Secondary Oscillator	46	—	—
pa13	COEXCTRL0	RF Radio	16	I/O	DIG/ST
	SERCOM2_PAD0	SERCOM2		I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	RPA13	PPS		I/O	DIG/ST
	IOCA13	Change Notification		I	ST
	RA13	GPIO		I/O	DIG/ST
pa14	COEXCTRL1	RF Radio	17	I/O	DIG/ST
	SERCOM2_PAD1	SERCOM2		I/O	DIG/ST/ I2C ⁽¹⁾ /SMB
	RPA14	PPS		I/O	DIG/ST
	IOCA14	Change Notification		I	ST
	RA14	GPIO		I/O	DIG/ST

Notes:

1. SERCOMx_PADx supports SPI, UART, and I2C modes. SERCOM I2C mode is specifically mentioned in the "Type" column as it does not support Peripheral Pin Select (PPS) functionality.
2. Not applicable for WBZ451H users.

Table 6-12. Priority for Device Pins PBn (n=0-13)

Pin Name	Functions in Priority Order	Reference Peripheral	Pin Number (48-pin)	I/O	Type
pb0	COEXCTRL2	RF Radio	30	I/O	DIG/ST
	AN4	ADC		I	ANALOG
	AC_AIN2	Analog Comparator		I	ANALOG
	RPB0	PPS		I/O	DIG/ST
	IOCB0	Change Notification		I	ST
	RB0	GPIO		I/O	DIG/ST
pb1 ⁽¹⁾	AN5	ADC	31	I	ANALOG
	AC_AIN3	Analog Comparator		I	ANALOG
	RPB1	PPS		I/O	DIG/ST
	IOCB1	Change Notification		I	ST
	RB1	GPIO		I/O	DIG/ST
pb2	AN6	ADC	32	I	ANALOG
	AC_AIN0	Analog Comparator		I	ANALOG
	RPB2	PPS		I/O	DIG/ST
	IOCB2	Change Notification		I	ST
	RB2	GPIO		I/O	DIG/ST
pb3	AN7	ADC	33	I	ANALOG
	AC_AIN1	Analog Comparator		I	ANALOG
	RPB3	PPS		I/O	DIG/ST
	IOCB3	Change Notification		I	ST
	RB3	GPIO		I/O	DIG/ST
pb4	PGC1ENTRY	Debug	34	I	ST
	FECTRL3	RF Radio		O	DIG
	AN0	ADC		I	ANALOG
	TP4	DO NOT USE		—	—
	RPB4	PPS		I/O	DIG/ST
	INT0	Edge Interrupt		I	ST/STMV
	IOCB4	Change Notification		I	ST
	RB4	GPIO		I/O	DIG/ST
pb5	PGC4ENTRY	Debug	35	I	ST
	TRACEDAT0	Debug		O	DIG
	FECTRL4	RF Radio		O	DIG
	AN1	ADC		I	ANALOG
	TP5	DO NOT USE		—	—
	RPB5	PPS		I/O	DIG/ST
	IOCB5	Change Notification		I	ST
	RB5	GPIO		I/O	DIG/ST

.....continued

Pin Name	Functions in Priority Order	Reference Peripheral	Pin Number (48-pin)	I/O	Type
pb6	PGD1ENTRY	Debug	37	I	ST
	TRACEDAT1	Debug		O	DIG
	FECTRL5	RF Radio		O	DIG
	AN2	ADC		I	ANALOG
	ANN0	ADC (Differential)		I	ANALOG
	RPB6	PPS		I/O	DIG/ST
	IOCB6	Change Notification		I	ST
	RB6	GPIO		I/O	DIG/ST
pb7	PGD4ENTRY	Debug	38	I	ST
	CM4_SWO	Debug		O	DIG
	TRACEDAT2	Debug		O	DIG
	AN3	ADC		I	ANALOG
	LVDIN	LVD Voltage Reference		I	ANALOG
	RPB7	PPS		I/O	DIG/ST
	IOCB7	Change Notification		I	ST
	RB7	GPIO		I/O	DIG/ST
pb8	CM4_SWCLK	DEBUG	41	I	ST
	RPB8	PPS		I/O	DIG/ST
	IOCB8	Change Notification		I	ST
	RB8	GPIO		I/O	DIG/ST
pb9	CM4_SWDIO	DEBUG	40	I/O	DIG/ST
	RPB9	PPS		I/O	DIG/ST
	IOCB9	Change Notification		I	ST
	RB9	GPIO		I/O	DIG/ST
pb10	QSPI_CS	QSPI	43	O	DIG
	RPB10	PPS		I/O	DIG/ST
	IOCB10	Change Notification		I	ST
	RB10	GPIO		I/O	DIG/ST
pb11	QSPI_SCK	QSPI	44	I/O	DIG/ST
	RPB11	PPS		I/O	DIG/ST
	IOCB11	Change Notification		I	ST
	RB11	GPIO		I/O	DIG/ST
pb12	QSPI_DATA0	QSPI	14	I/O	DIG/ST
	RPB12	PPS		I/O	DIG/ST
	IOCB12	Change Notification		I	ST
	RB12	GPIO		I/O	DIG/ST
pb13	QSPI_DATA1	QSPI	15	I/O	DIG/ST
	RTC_EVENT	RTCC		O	DIGMV
	RPB13	PPS		I/O	DIG/ST
	IOCB13	Change Notification		I	ST
	RB13	GPIO		I/O	DIG/ST

Note:

1. Not applicable for WBZ451H users.

6.4 I/O Ports Control Registers

Notes: The following conventions are used in the following tables:

- x = Unknown value on Reset
- — = Unimplemented, read as '0'; Reset values are shown in hexadecimal
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- '1' = Bit is set
- '0' = Bit is cleared

Table 6-13. PortA Register Map

Virtual Address (0x4400-2200)	Register	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0010	TRISA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	TRISA14	TRISA13	TRISA12 ⁽¹⁾	TRISA11 ⁽¹⁾	TRISA10	TRISA9	TRISA8	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	0000
0020	PORTA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	RA14	RA13	RA12 ⁽²⁾	RA11 ⁽²⁾	RA10 ⁽³⁾	RA9 ⁽³⁾	RA8	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	0000
0030	LATA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	LATA14	LATA13	LATA12 ⁽²⁾	LATA11 ⁽²⁾	LATA10	LATA9	LATA8	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	0000
0040	ODCA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	ODCA14	ODCA13	—	—	ODCA10	ODCA9	ODCA8	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000
0050	CNPUA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CNPUA14	CNPUA13	—	—	CNPUA10	CNPUA9	CNPUA8	CNPUA7	CNPUA6	CNPUA5	CNPUA4	CNPUA3	CNPUA2	CNPUA1	CNPUA0	0000
0060	CNPDA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CNPDA14	CNPDA13	—	—	CNPDA10	CNPDA9	CNPDA8	CNPDA7	CNPDA6	CNPDA5	CNPDA4	CNPDA3	CNPDA2	CNPDA1	CNPDA0	0000
0070	CNCONA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	ON	FRZ	SIDL	—	EDGEDETECT	—	—	—	—	—	—	—	—	—	—	—	0000
0080	CNENA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CNENA14	CNENA13	—	—	CNENA10	CNENA9	CNENA8	CNENA7	CNENA6	CNENA5	CNENA4	CNENA3	CNENA2	CNENA1	CNENA0	0000
0090	CNSTATA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CN STATA14	CN STATA13	CN STATA12	CN STATA11	CN STATA10	CN STATA9	CN STATA8	CN STATA7	CN STATA6	CN STATA5	CN STATA4	CN STATA3	CN STATA2	CN STATA1	CN STATA0	0000
00A0	CNNEA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CNNEA14	CNNEA13	—	—	CNNEA10	CNNEA9	CNNEA8	CNNEA7	CNNEA6	CNNEA5	CNNEA4	CNNEA3	CNNEA2	CNNEA1	CNNEA0	0000
00B0	CNFA	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	CNFA14	CNFA13	—	—	CNFA10	CNFA9	CNFA8	CNFA7	CNFA6	CNFA5	CNFA4	CNFA3	CNFA2	CNFA1	CNFA0	0000
00C0	SRCON0A	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	SR014	SR013	—	—	SR010	SR09	SR08	SR07	SR06	SR05	SR04	SR03	—	SR01	SR00	0000
00D0	SRCON1A	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	SR114	SR113	—	—	SR110	SR19	SR18	SR17	SR16	SR15	SR14	SR13	—	SR11	SR10	0000

Notes:

1. All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.
2. Not applicable for PIC32CX1012BZ25048.
3. Not applicable for WBZ451H users.

Table 6-14. PortB Register Map

Virtual Address (0x4400_2300)	Register	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0100	ANSELB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	FFFF
0110	TRISB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000
0120	PORTB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1 ⁽²⁾	RB0	0000
0130	LATB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000
0140	ODCB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	ODCB13	ODCB12	ODCB11	ODCB10	ODCB9	ODCB8	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	0000
0150	CNPUB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CNPUB13	CNPUB12	CNPUB11	CNPUB10	CNPUB9	CNPUB8	CNPUB7	CNPUB6	CNPUB5	CNPUB4	CNPUB3	CNPUB2	CNPUB1	CNPUB0	0000
0160	CNPDB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CNPDB13	CNPDB12	CNPDB11	CNPDB10	CNPDB9	CNPDB8	CNPDB7	CNPDB6	CNPDB5	CNPDB4	CNPDB3	CNPDB2	CNPDB1	CNPDB0	0000
0170	CNCONB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	ON	FRZ	SIDL	—	EDGE DETECT	—	—	—	—	—	—	—	—	—	—	—	0000
0180	CNENB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CNENB13	CNENB12	CNENB11	CNENB10	CNENB9	CNENB8	CNENB7	CNENB6	CNENB5	CNENB4	CNENB3	CNENB2	CNENB1	CNENB0	0000
0190	CNSTATB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CN STATB13	CN STATB12	CN STATB11	CN STATB10	CN STATB9	CN STATB8	CN STATB7	CN STATB6	CN STATB5	CN STATB4	CN STATB3	CN STATB2	CN STATB1	CN STATB0	0000
01A0	CNNEB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CNNEB13	CNNEB12	CNNEB11	CNNEB10	CNNEB9	CNNEB8	CNNEB7	CNNEB6	CNNEB5	CNNEB4	CNNEB3	CNNEB2	CNNEB1	CNNEB0	0000
01B0	CNFB	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	CNFB13	CNFB12	CNFB11	CNFB10	CNFB9	CNFB8	CNFB7	CNFB6	CNFB5	CNFB4	CNFB3	CNFB2	CNFB1	CNFB0	0000
01C0	SRCON0B	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	SR013	SR012	SR011	SR010	—	—	—	—	—	—	—	—	—	—	0000
01D0	SRCON1B	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	SR113	SR112	SR111	SR110	—	—	—	—	—	—	—	—	—	—	0000

1. **Note:** All registers in this table have corresponding CLR, SET and INV registers at their virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.
2. Not applicable for WBZ451H users.

Table 6-15. Peripheral Pin Select Input Registers

Virtual Address (400_1000)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0000h	EXTINT0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	EXTINT0R[3:0]				0000
0004h	EXTINT1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	EXTINT1R[3:0]				0000
0008h	EXTINT2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	EXTINT2R[3:0]				0000
000Ch	EXTINT3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	EXTINT3R[3:0]				0000
003Ch	NMIR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	NMIR[3:0]				0000
0040h	SCOM0P0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM0P0R[3:0]				0000
0044h	SCOM0P1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM0P1R[3:0]				0000
0048h	SCOM0P2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM0P2R[3:0]				0000
004Ch	SCOM0P3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM0P3R[3:0]				0000
0050h	SCOM1P0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM1P0R[3:0]				0000
0054h	SCOM1P1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM1P1R[3:0]				0000
0058h	SCOM1P2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM1P2R[3:0]				0000
005Ch	SCOM1P3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM1P3R[3:0]				0000
0060h	SCOM2P0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM2P0R[3:0]				0000

.....continued																			
Virtual Address (4400_1000)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0064h	SCOM2P1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM2P1R[3:0]				0000
0068h	SCOM2P2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM2P2R[3:0]				0000
006Ch	SCOM2P3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM2P3R[3:0]				0000
0070h	SCOM3P0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM3P0R[3:0]				0000
0074h	SCOM3P1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM3P1R[3:0]				0000
0078h	SCOM3P2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM3P2R[3:0]				0000
007Ch	SCOM3P3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	SCOM3P3R[3:0]				0000
0080h	QSCKR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	QSCKR[3:0]				0000
0084h	QD0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	QD0R[3:0]				0000
0088h	QD1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	QD1R[3:0]				0000
008Ch	QD2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	QD2R[3:0]				0000
0090h	QD3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	QD3R[3:0]				0000
0094h	REFIR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	REFIR[3:0]				0000
0098h	CCLIN0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN0R[3:0]				0000

.....continued																			
Virtual Address (4400_1000)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
009Ch	CCLIN1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN1R[3:0]				0000
00A0h	CCLIN2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN2R[3:0]				0000
00A4h	CCLIN3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN3R[3:0]				0000
00A8h	CCLIN4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN4R[3:0]				0000
00ACh	CCLIN5R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	CCLIN5R[3:0]				0000
00B0h	TC0WO0G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC0WO0G1R[3:0]				0000
00B4h	TC0WO0G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC0WO0G2R[3:0]				0000
00B8h	TC0WO1G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC0WO1G3R[3:0]				0000
00BCh	TC0WO1G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC0WO1G4R[3:0]				0000
00C0h	TC1WO0G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC1WO0G1R[3:0]				0000
00C4h	TC1WO1G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC1WO0G2R[3:0]				0000
00C8h	TC2WO0G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC2WO0G1R[3:0]				0000
00CCh	TC2WO0G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC2WO0G3R[3:0]				0000
00D0h	TC2WO1G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC2WO1G2R[3:0]				0000

.....continued																			
Virtual Address (4400_1000)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
00D4h	TC2WO1G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC2WO1G4R [3:0]				
00D8h	TC3WO0G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC3WO0G1R [3:0]				
00DCh	TC3WO0G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC3WO0G3R[3:0]				
00E0h	TC3WO1G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC3WO1G2R [3:0]				
00E4h	TC3WO1G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	TC3WO1G4R[3:0]				0000

Table 6-16. Peripheral Pin Select Output Registers

Virtual Address (400_1000) (1)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0200h	RPA0G2R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA0G2R[4:0]				—	0000
0204h	RPA0G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA0G3R[4:0]				—	0000
0208h	RPA1G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA1G3R[4:0]				—	0000
020Ch	RPA1G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA1G4R[4:0]				—	0000
0210h	RPA2G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA2G1R[4:0]				—	0000
0214h	RPA2G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA2G4R [4:0]				—	0000
0218h	RPA3G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA3G1R[4:0]				—	0000
021Ch	RPA3G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA3G2R[4:0]				—	0000
0220h	RPA3G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA3G3R[4:0]				—	0000
0224h	RPA4G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA4G2R[4:0]				—	0000
0228h	RPA4G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA4G3R[4:0]				—	0000
022Ch	RPA4G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA4G4R[4:0]				—	0000
0230h	RPA5G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA5G1R[4:0]				—	0000
0234h	RPA5G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA5G3R[4:0]				—	0000

.....continued																			
Virtual Address (4400_1000) (1)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0238h	RPA5G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA5G4R[4:0]				—	0000
023Ch	RPA6G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA6G1R[4:0]				—	0000
0240h	RPA6G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA6G2R[4:0]				—	0000
0244h	RPA6G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA6G4R[4:0]				—	0000
0248h	RPA7G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA7G1R[4:0]				—	0000
024Ch	RPA7G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA7G2R[4:0]				—	0000
0250h	RPA8G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA8G2R[4:0]				—	0000
0254h	RPA8G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA8G3R[4:0]				—	0000
0258h	RPA8G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA8G4R[4:0]				—	0000
025Ch	RPA9G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA9G1R[4:0]				—	0000
0260h	RPA9G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA9G3R[4:0]				—	0000
0264h	RPA9G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA9G4R[4:0]				—	0000
0268h	RPA10G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA10G1R[4:0]				—	0000
026Ch	RPA10G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA10G4R[4:0]				—	0000

.....continued																			
Virtual Address (4400_1000) (1)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0278h	RPA13G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA13G3R[4:0]				—	0000
027Ch	RPA13G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA13G4R[4:0]				—	0000
0280h	RPA14G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA14G1R[4:0]				—	0000
0284h	RPA14G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPA14G4R[4:0]				—	0000
028Ch	RPB0G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB0G1R[4:0]				—	0000
0290h	RPB0G2R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB0G2R[4:0]				—	0000
0294h	RPB1G2R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB1G2R[4:0]				—	0000
0298h	RPB1G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB1G3R[4:0]				—	0000
029Ch	RPB2G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB2G3R[4:0]				—	0000
02A0h	RPB2G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB2G4R[4:0]				—	0000
02A4h	RPB3G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB3G1R[4:0]				—	0000
02A8h	RPB3G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB3G4R[4:0]				—	0000
02ACh	RPB4G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB4G1R[4:0]				—	0000
02B0h	RPB4G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB4G2R[4:0]				—	0000

.....continued

Virtual Address (4400_1000) (1)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
02B4h	RPB5G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB5G2R[4:0]				—	0000
02B8h	RPB5G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB5G3R[4:0]				—	0000
02BCh	RPB6G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB6G3R[4:0]				—	0000
02C0h	RPB6G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB6G4R[4:0]				—	0000
02C4h	RPB7G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB7G1R[4:0]				—	0000
02C8h	RPB7G4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB7G4R[4:0]				—	0000
02CCh	RPB8G1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB8G1R[4:0]				—	0000
02D0h	RPB8G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB8G2R[4:0]				—	0000
02D4h	RPB9G2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB9G2R[4:0]				—	0000
02D8h	RPB9G3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB9G3R[4:0]				—	0000
02DCh	RPB10G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB10G3R[4:0]				—	0000
02E0h	RPB10G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB10G4R[4:0]				—	0000
02E4h	RPB11G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB11G1R[4:0]				—	0000
02E8h	RPB11G4R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB11G4R[4:0]				—	0000

.....continued

Virtual Address (4400_1000) (1)	Register Name	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
02ECh	RPB12G1R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB12G1R[4:0]					0000
02F0h	RPB12G2R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB12G2R[4:0]					0000
02F4h	RPB13G2R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB13G2R[4:0]					0000
02F8h	RPB13G3R*	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	RPB13G3R[4:0]					0000

Related Links

[6.1.9. CLR, SET and INV Registers](#)

6.4.1 Change Notice Control for PORTx Register

Name: CNCONx
Offset: See the following Note
Reset: 0x0
Property: -

Note:

- For the Offset address, see the *PortA Register Map* and *PortB Register Map* tables in the *I/O Ports Control Registers* from Related Links.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ON	FRZ	SIDL		EDGEDETECT			
Access	R/W	R/W	R/W		R/W			
Reset	0	0	0		0			
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – ON Change Notice (CN) Control ON bit

- 1 = Change Notice is enabled
- 0 = Change Notice is disabled

Bit 14 – FRZ Freeze in the Debug mode bit

- 1 = Freezes the module operation when the emulator is in the Debug mode
- 0 = Continues the module operation when the emulator is in the Debug mode

Bit 13 – SIDL Stop in the Idle mode bit

- 1 = Discontinues the module operation when device enters the Idle mode
- 0 = Continues the module operation even in the Idle mode

Bit 11 – EDGEDETECT Change Notification Style bit

- 1 = Edge Style. Detects edge transitions (CNF_x used for CN Event).
- 0 = Mismatch Style. Detects change from last PORT_x read (CNSTAT_x used for CN Event).

Related Links

[6.4. I/O Ports Control Registers](#)

6.5 Operation in Power Saving Modes

6.5.1 I/O Port Operation in Sleep Mode

As the device enters Sleep mode, the system clock is disabled; however, the CN module continues to operate. If one of the enabled CN pins changes state, the corresponding Interrupt flag is set in NVIC, the device wakes from Sleep (or Idle) mode, then executes the CN interrupt service routine.

6.5.2 I/O Port Operation in Idle Mode

As the device enters Idle mode, the system clock sources remain functional. The SIDL bit (CNCONx[13]) selects whether the module is going to stop or continue to function in the Idle mode.

- If SIDL = 1, the module continues to sample Input CN I/O pins in the Idle mode; however, synchronization is disabled.
- If SIDL = 0, the module continues to synchronize and samples input CN I/O pins in the Idle mode.

6.6 Results of Various Resets

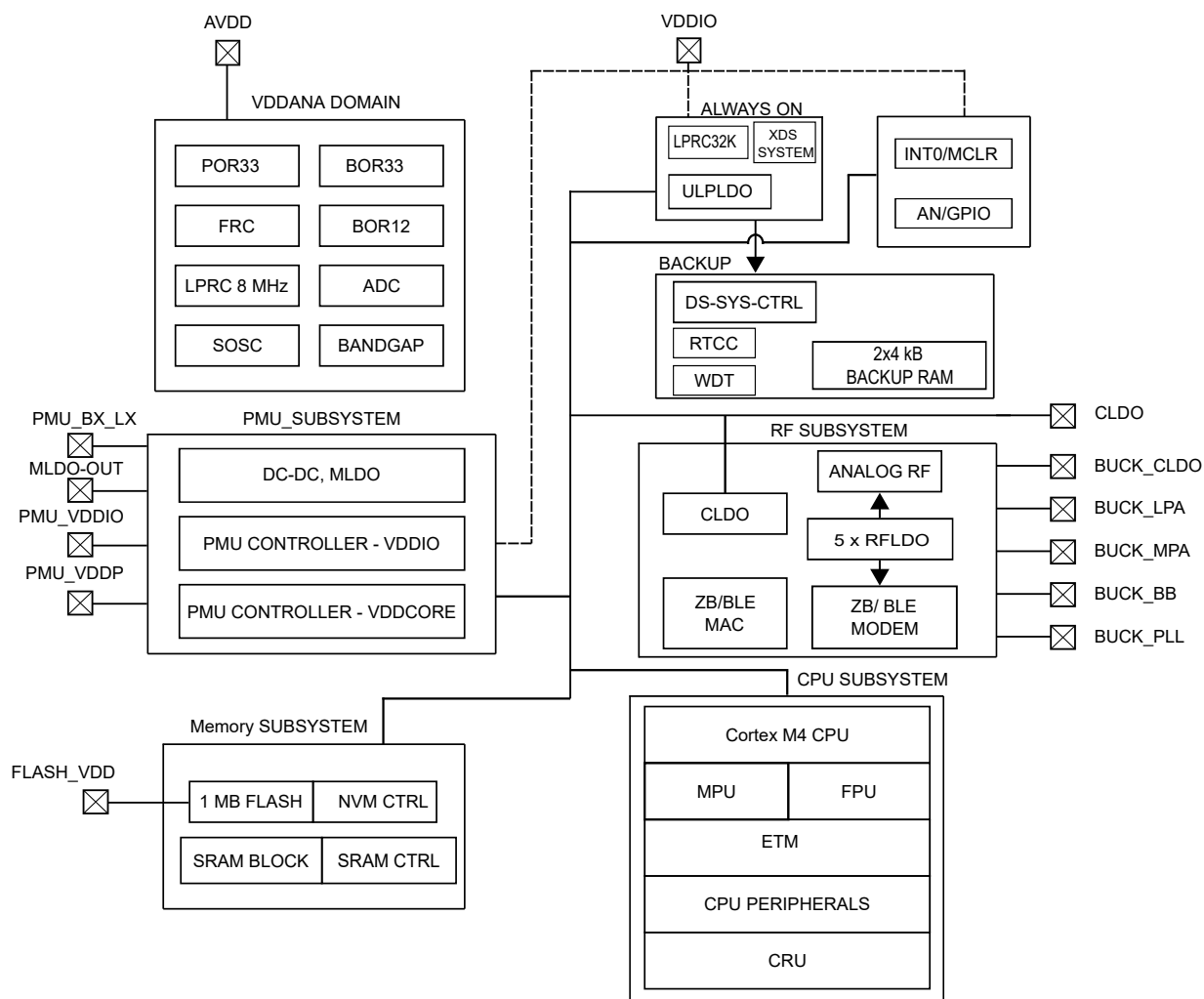
Table 6-17. Results of Resets Available

Reset Name	Description
Device Reset	All I/O registers are forced to their reset states upon a device Reset.
Power-on Reset (PoR)	All I/O registers are forced to their reset states upon a Power-on Reset (POR).
Watchdog Reset	All I/O registers are unchanged upon a Watchdog Reset.

7. Power Subsystem

7.1 Block Diagram

Figure 7-1. Power Subsystem Block Diagram



The power domains of PIC32CX-BZ2 and WBZ45 are as follows:

- VDD - 1.9V to 3.6V, Main Supply powering VDDIO, FLASH_VDD, AVDD, PMU_VDDIO, PMU_VDDP
 - VDDIO
 - 1.9V to 3.6, powering the AON, PMU-CTRL, AN/GPIO, INT0/MCLR, BKUP
 - FLASH_VDD
 - 1.9V to 3.6V, Filtered version of VDD (powering the Flash)
 - AVDD
 - 1.9V to 3.6V, Filtered version of VDD for system analog functionality
 - PMU_VDDIO
 - 1.9V to 3.6V, Filtered version of VDD (powering the PMU sub-system)
 - PMU_VDDP

- 1.9V to 3.6V, Filtered version of VDD (powering the PMU sub-system)
- GND
 - Common GND for digital, analog and RF sub-systems

Other power supply pins are as follows:

- CLDO
 - Output pin for the internal voltage regulator for decoupling, this pin must not be used as an external power supply source
 - CLDO output is 1.2V
 - CLDO powers the VDDCORE
 - VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals
- PMU_BK_LX
 - Switching node for the Buck converter
- PMU_MLDO_OUT
 - 1.35V output pin from the internal LDO. This is the shared output pin for both MLDO and the DC-DC converter
 - PMU_MLDO_OUT powers the LDO's in the Wireless subsystem (BUCK_LPA, BUCK_MPA, BUCK_PLL, BUCK_BB and BUCK_CLDO)

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

7.2 Voltage Regulators

The following voltage regulators are available in the power subsystem:

- MLDO – Linear voltage regulator generating 1.35V for powering CLDO and other RF LDOs operates from 1.9V to 3.6V
- DC-DC – Switching voltage regulator generating 1.35V operates from 2.4V to 3.6V
- ULP_VREG – Ultra low power voltage regulator for operation in back-up mode
- RF LDOs – Powering the different blocks of the RF subsystem
- CLDO – Powering the VDDCORE of PIC32CX-BZ2

7.3 Power Supply Modes

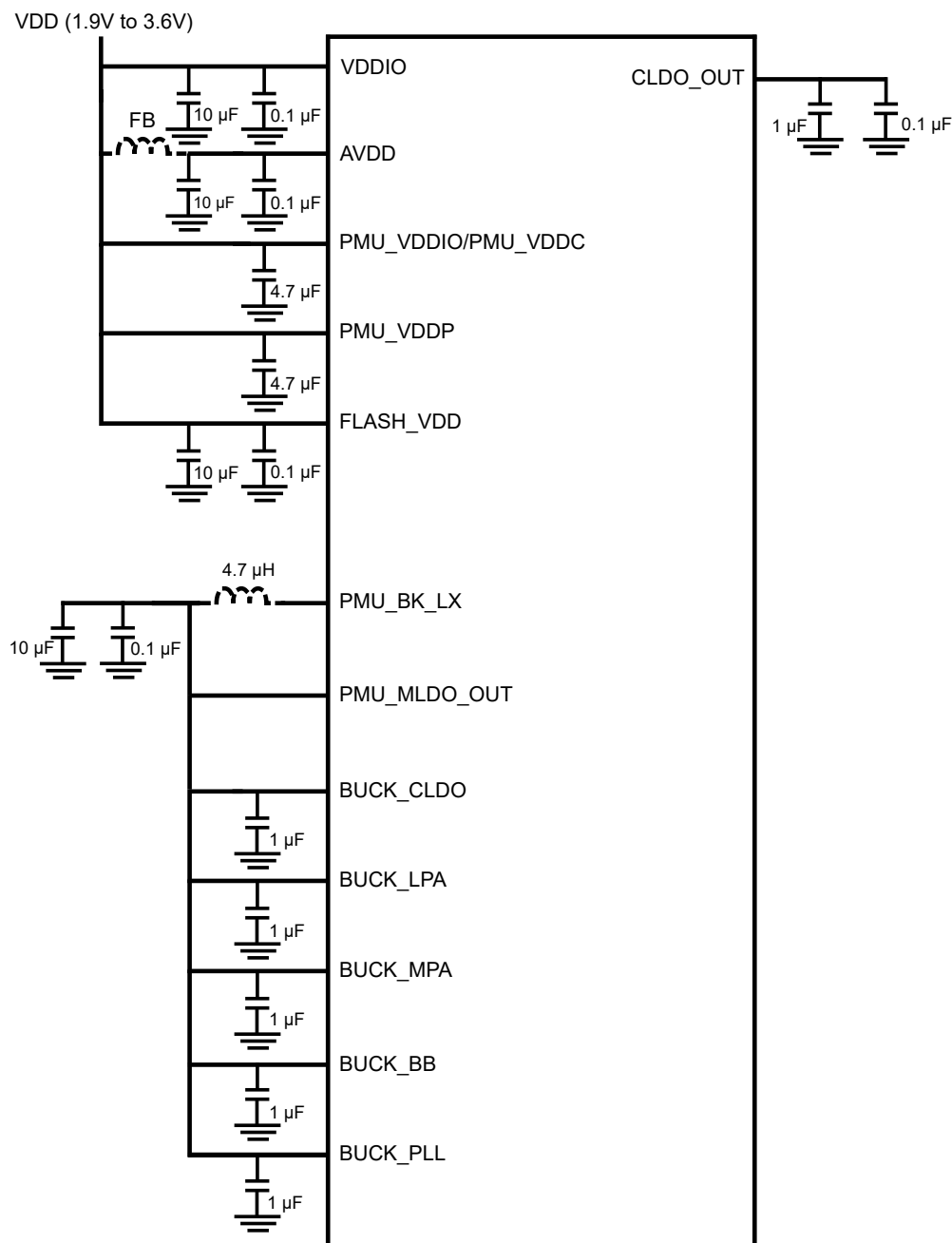
The PIC32CX-BZ2 supports a single power supply from 1.9V to 3.6V. The IO supply cannot be decoupled from the main supply. The same voltage must be applied to VDDx, PMU_VDDIO, VPMU_VDDC and AVDD with different levels of filtering. The internal voltage regulator has the following four different modes:

- Linear mode – This mode does not require any external inductor. This is the default mode when the CPU and peripherals are running. In this mode, the SoC is powered through the MLDO.
- Switching mode (Buck) – This is the most efficient mode when the CPU and peripherals are running. In this mode, the SoC is powered by the DC-DC converter.
- Low Power (PSK) mode – This is the mode used when the device is in Standby mode.
- Shutdown mode – When the device is in Backup mode, the internal regulator is turned off.

Selecting between the Switching mode and Linear mode can be done by software on-the-fly, but the power supply must be designed according to which mode is to be used.

7.4 Typical Power Supply Connection for SoC

Figure 7-2. Typical Power Supply Connection for SoC

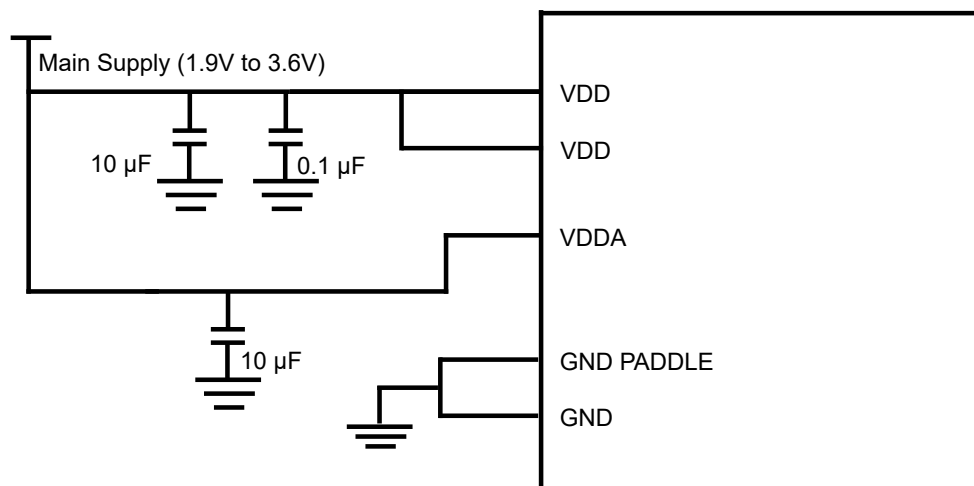


7.5 Typical Power Supply Connection for Module

The module requires only a single power supply on the VDD pins of the module.

- VDD can be from 1.9V to 3.6V

Figure 7-3. Module Schematics with VDD and Optional Bulk Capacitors



7.6 Power-Up Sequence

The characteristics of power-up sequence are as follows:

- The VDD/AVDD domains must rise at the same time
- On power-up, PIC32CX-BZ2 always start up on the MLDO mode
- After power-up in MLDO, the software can initiate the switch to Buck mode

7.6.1 Starting of Voltage Regulators

The characteristics of power-up voltage regulators are as follows:

- On power-up, the internal regulator starts in MLDO mode.
- After MLDO boots up, the CLDO gets initialized.
- Start the code execution.
- The RF system is maintained in Sleep mode during the power-up time.

7.6.2 Starting-up of Crystals

The characteristics of power-up crystals are as follows:

- The power-up of the SoC happens with the internal oscillators. After the power-up, the user software can request to switch on the SOSC and the XOSC crystals.

7.6.3 BOR and POR

The Brown-out Reset (BOR) monitors the VDD supply voltage. On detection of a brown-out condition, the BOR re-arms the POR. In this device, the minimum BOR trip point is the voltage below, which I/O is considered un-trusted; thus, it generates a Reset. There are three BOR in the PIC32CX-BZ2 and the WBZ45 Module:

- BOR3.3 to monitor VDD
- BOR1.2 to monitor VDDCORE
- ZPBOR monitors VDD during the Deep Sleep and Extreme Deep Sleep mode if enabled

8. Product Memory Mapping Overview

Figure 8-1. Product Mapping

System	Peripheral Bridge-A	Peripheral Bridge-B	Peripheral Bridge-C	Peripheral Bus-PIC	
0x00000000	Boot Flash (Alias)	0x40000000	PAC	0x44000000	Device Config Registers
0x00005000	Device Config/Boot (Alias)	0x40000400	FREM	0x44000500	WDT
0x00006000	OTP Page (Alias)	0x40000800	EIC	0x44000600	FC
0x00007000	Unimplemented	0x40000C00	SERCOM-0	0x44000800	PFW
0x00040000	Unimplemented	0x40001000	SERCOM-1	0x44000A00	CRU
0x00045000	Unimplemented	0x40001400	TC-0	0x44000C00	BOR_CMP
0x00047000	Unimplemented	0x40001800	TC-1	0x44000E00	DMT
0x00048000	Unimplemented	0x40001C00	TC-2	0x44001000	PPS
0x01000000	Flash	0x40002000	TC-3	0x44001400	ADCCON
0x01100000	Unimplemented	0x40002400	TCC-0	0x44001A00	ADCCFG
0x02000000	PUKCC	0x40002800	TCC-1	0x44001E00	ADCOUT
0x02012000	Unimplemented	0x40002C00	TCC-2	0x44002200	PORT A
0x03000000	CM4CCTCM	0x40003000	Reserved	0x44002300	PORT B
0x03001000	Unimplemented	0x40FFFFFF		0x44002400	Reserved
0x04000000	QSPI			0x44010000	Reserved
0x05000000	Unimplemented			0x44012000	ICD
0x20000000	System RAM			0x44012400	PCHE
0x40000000	Unimplemented			0x44012800	Reserved
0x41000000	PB-A			0x44013E00	PMU
0x42000000	PB-B			0x44014000	Backup RAM
0x43000000	PB-C			0x44016000	Reserved
0x44000000	Unimplemented			0x4401FFFF	
0x44020000	PB-(PIC)				
0x44020000	Unimplemented				
0xE0000000	CM4F System Registers				
0xFFFF3FFF					

Peripherals on Standard Peripheral Bus

Peripherals on Fast Peripheral Bus

Notes:

1. Access attempts to any unimplemented memory location generates a bus error.
2. The PUKCC space is fixed as non-cacheable and bufferable.
3. The QSPI space “cacheable and bufferable” attribute is controlled using CFGCON1.QSCHE_EN.

Table 8-1. Device Configuration Map

Register Name		Reset value from NVR memory?	Writable in User Mode?	Corresponding Write Lock Bit(s)	Purpose
System Configuration					
CFGCON0(L)	0x4400_0000	X	X	CFGLOCK[1:0]	Misc. System Configuration
CFGCON1(L)	0x4400_0010	X	X	CFGLOCK[1:0]	Misc. System Configuration
CFGCON2(L)	0x4400_0020	X	X	CFGLOCK[1:0]	Misc. System Configuration
CFGCON3	0x4400_0030	—	X	CFGLOCK[1:0]	Misc. System Configuration
CFGCON4(L)	0x4400_0040	X	X	CFGLOCK[1:0]	Misc. System Configuration
CFGPGQOS	0x4400_0050	—	X	CFGCON0.PGLOCK	Bus Matrix Permission Groups
CFGPCLKGEN1	0x4400_0060	—	X	CFGLOCK[1:0]	Peripheral Clock Gen Reg-1
CFGPCLKGEN2	0x4400_0070	—	X	CFGLOCK[1:0]	Peripheral Clock Gen Reg-2
CFGPCLKGEN3	0x4400_0080	—	X	CFGLOCK[1:0]	Peripheral Clock Gen Reg-3
ID ²	0x4400_0090	X	—	N/A- Read Only	32-bit Device ID
USER_ID(L)	0x4400_00A0	X	X	CFGLOCK[1:0]	16-bit User ID
SYSKEY	0x4400_00B0	—	X	Sequence	System Lock feature
PMD1	0x4400_00C0	—	X	CFGCON0.PMDLOCK	Peripheral Module Disable
PMD2	0x4400_00D0	—	X	CFGCON0.PMDLOCK	Peripheral Module Disable
PMD3	0x4400_00E0	—	X	CFGCON0.PMDLOCK	Peripheral Module Disable
Boot Configuration					
BCFG0	0x4400_0200	X	—	None	Pre-boot user configuration

Notes:

1. Registers marked with (L) are loadable from Flash, and they can be controlled by software after the boot with the correct unlock sequence.
2. ID Register is a JTAB ID register, maintained for legacy reasons. The actual external-facing ID register is DSU.DID register. See *DID* register in the *Device Service Unit (DSU)* from Related Links.

Table 8-2. CM4 System Components Register Offset Map

Peripheral		TA Clock	Size (Bytes)	Virtual Address		Physical Address	
ABRV.	Description			Start	End	Start	End
CM4 System Components accessible via CM4 PPB Bus (Only DAP and CM4 can access the registers)				Base Address 0xE000_0000		Base Address 0xE000_0000	
ITM	Inst. TM	SYS_CLK	4 KB	0000_0000	0000_0FFF	0000_0000	0000_0FFF
DWT	Data WT	SYS_CLK	4 KB	0000_1000	0000_1FFF	0000_1000	0000_1FFF
FPB	Flash PB	SYS_CLK	4 KB	0000_2000	0000_2FFF	0000_2000	0000_2FFF
RSVD	Reserved	—	4 KB*n	0000_3000	0000_DFFF	0000_3000	0000_DFFF
SCS	Sys. CS	SYS_CLK	4 KB	0000_E000	0000_EFFF	0000_E000	0000_EFFF
RSVD	Reserved	—	4 KB*n	0000_F000	0003_FFFF	0000_F000	0003_FFFF
TPIU	Trace PIU	SYS_CLK	4 KB	0004_0000	0004_0FFF	0004_0000	0004_0FFF
ETM	ETM	SYS_CLK	4 KB	0004_1000	0004_1FFF	0004_1000	0004_1FFF
ETB	ETB	SYS_CLK	4 KB	0004_2000	0004_2FFF	0004_2000	0004_2FFF
RSVD	Reserved	—	4 KB*n	0004_3000	000F_EFFF	0004_3000	000F_EFFF
CROM	CSight ROM	SYS_CLK	4 KB	000F_F000	000F_FFFF	000F_F000	000F_FFFF
RSVD	Reserved	—	4 KB*n	0010_0000	FFFF_FFFF	0010_0000	FFFF_FFFF

Notes:

1. All system and debug components carry a unique ID accessible via its own register space.
2. The DAP derives the base address of the components from CROM entry values.
3. Component Base address = CROM Base address + CROM Entry value.
4. Core sight ROM entries are not provided in this document.
5. Refer to *CM4F* documentation for details on each component register space (developer.arm.com/documentation/ddi0439/b/System-Control/Register-summary).

Related Links

[12.13.8. DID](#)

9. Prefetch Cache (PCHE)

9.1 Overview

The prefetch cache is a performance-enhancing module included in the PIC32CX-BZ2 devices, along with the L1 cache (Cortex M Cache Controller) to the Cortex-M4F CPU.

9.2 Features

The Prefetch module increases the system performance for most of the applications.

The Prefetch module includes the following features:

- Fully Associative Lines For:
 - Four lines for CPU instructions cache
 - Two lines for CPU data cache
 - Two lines for peripheral data cache
- 16-Byte Cache Lines and 128-Bits Parallel Memory Fetch
- Configurable Predictive Prefetch for CPU Instructions Cache
- Error Detection and Correction

9.3 Overview

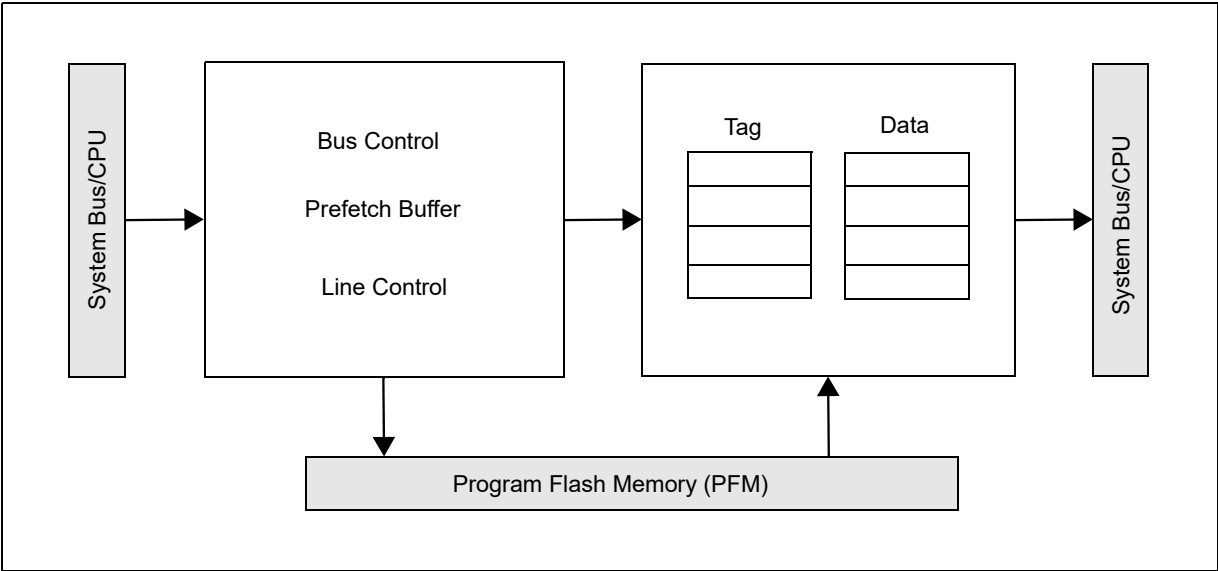
When running the prefetch module at high-clock rates, insert the Wait states into Program Flash Memory (PFM) read transactions to meet the access time of the PFM. The user can hide the Wait states to the core by prefetching and storing the instructions in a temporary holding area that the CPU can access quickly. Although, the data path to the CPU is 32 bits wide, the data path to the PFM is 128 bits wide. This wide data path provides the same bandwidth to the CPU as a 32-bit path running at four times the frequency.

The prefetch module holds a subset of PFM in temporary holding spaces known as lines. Each line contains a tag and data field. In general, the lines hold a copy of what is currently in memory to make instructions or data available to the CPU without the Wait states.

The CPU or a peripheral can request the data located in the PFM. If the requested data is not currently stored in the prefetch module line, a read is performed to the PFM at the correct address, and the data is supplied to the prefetch module and to the CPU or peripheral. If the requested data is stored in the prefetch module and is valid, the data is supplied to the CPU or peripheral without Wait states.

The following figure illustrates a block diagram of the prefetch module. Logically, the prefetch module fits between the system bus module and the PFM.

Figure 9-1. Prefetch Cache Block Diagram



9.3.1 Line Organization

The Prefetch module consists of two arrays, data and tag, each of which hold four lines. A data array consists of program instructions, program data or peripheral data. Address matches are based on the physical address, not the virtual address.

Each line in the tag array contains the following information:

- Tag – Physical address of the data held in the data line
- Valid bit

Each line in the data array, contains 16 bytes of data. Depending on the line, the data can be CPU instructions, CPU data or peripheral data.

The following figures illustrate the organization of a line.

Figure 9-2. Tag Line

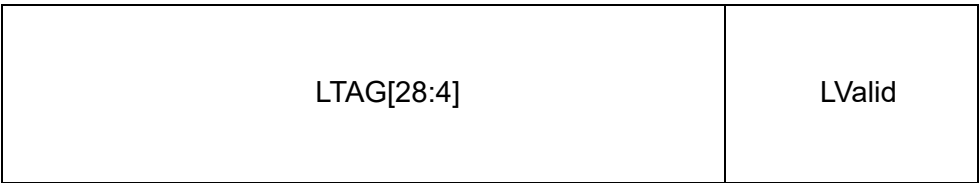
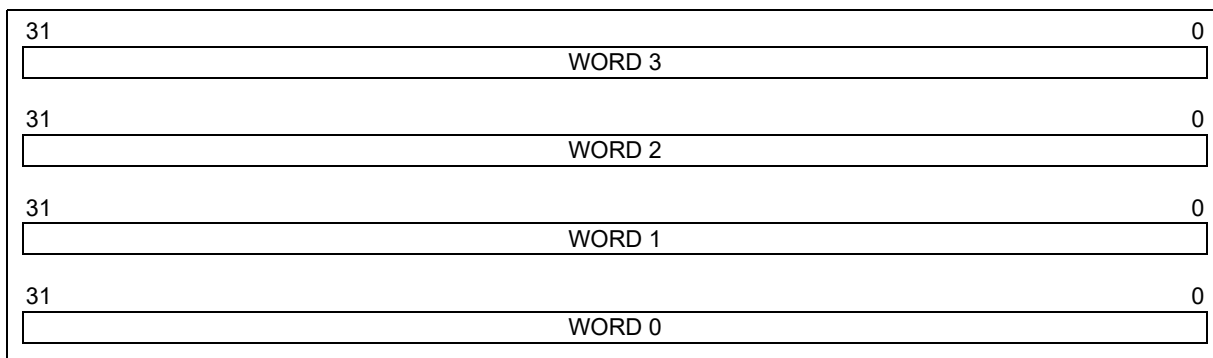


Figure 9-3. Data Line



9.4 Prefetch Behavior

The prefetch module complements an L1 CPU (CMCC) cache rather than replacing it. Four 128-bit (16-byte) lines hold instructions, two 128-bit (16-byte) lines hold CPU data and two 128-bit (16-byte) lines hold peripheral data from the PFM. The prefetch module uses the Wait state's value from the PFMWS[3:0] bits (CHECON[3:0]) and Address Wait state ADRWS bit (CHECON[8]) to determine how long it must wait for Flash access when it reads instructions or data from the PFM.

If the instructions or data already reside in the prefetch module line, the prefetch module returns the instruction or data in '0' Wait states. For CPU instructions, if predictive prefetch is enabled and the code is 100% linear, the prefetch module provides instructions back to the CPU with the Wait states only on the first instruction of the prefetch module line.

If the CPU accesses uncacheable addresses, it bypasses the cache. During the bypass, the prefetch module accesses the PFM for every instruction, incurring an address setup time defined by ADRWS and a Flash access time as defined by PFMWS bits. Therefore, the total Flash wait states is a sum of ADRWS and PFMWS. The Bypass mode is also forced for a cache if its associated I/D/A CHEEN bit (CHECON) is zero.

To allow caching for I and/or D caches, set the I and/or D *CHEEN bit to '1'. To enable a cache, set the ACHEEN bit to '1'.

9.5 Configurations

The CHECON register controls the general configurations available for accelerating the instruction and data accesses to the Flash memory system.

The Prefetch module implements the following general options:

- The PFMWS[3:0] bits (CHECON[3:0]) control the number of system clock cycles required to access the PFM. The total Flash Wait states is a sum of ADRWS and PFMWS.
- The PREFEN[1:0] bits (CHECON[5:4]) control the predictive and prefetched instruction, which allows the cache controller to fetch the next 16-byte aligned set of instructions.
- The PFMSECEN bit (CHECON[7]) controls the Prefetch module that generates an interrupt event on a specific count of single bit errors corrected by the Flash Error Correction Code (ECC).
- The ADRWS bit (CHECON[8]) controls the number of system clock cycles required for address setup to PFM.
- The CHEPERF bit (CHECON[12]) controls the gathering statistics of the CPU instruction cache.
- The ICHECOH bit (CHECON[16]) controls the auto invalidate for the CPU instruction cache.
- The DCHECOH bit (CHECON[17]) controls the auto invalidate for the CPU data cache.
- The ACHECOH bit (CHECON[18]) controls the auto invalidate for the peripheral data cache.

- The ICHEINV bit (CHECON[20]) controls the manual invalidate for the CPU instruction cache.
- The DCHEINV bit (CHECON[21]) controls the manual invalidate for the CPU data cache.
- The ACHEINV bit (CHECON[22]) controls the manual invalidate for the peripheral data cache.
- The ICHEEN bit (CHECON[24]) controls the CPU instruction cache enable.
- The DCHEEN bit (CHECON[25]) controls the CPU data cache enable.
- The ACHEEN bit (CHECON[26]) controls the peripheral data cache enable.

9.6 Predictive Prefetch Behavior

When the user configures the module for predictive prefetch, the prefetch module predicts the next line address, fetches the instruction and, then, stores it in the prefetch buffer. If the requested instruction is not in a prefetch module line and the read address matches the predicted address, the content of the prefetch buffer is loaded in the prefetch module line while simultaneously returning the critical word to the read initiator.

On enabling the predictive prefetch, the prefetch function starts predicting based on the first address read to the PFM. When the user places the first line in the prefetch module, the module increments the address to the next 16-byte aligned address and starts a PFM access.

The predictive prefetches, like all PFM read accesses, are never aborted. If a new address request does not match the predicted address, a new PFM access occurs after the current access finishes. The PREFEN [1:0] bits (CHECON[5:4]) can start a predictive prefetch. This allows the cache controller to speculatively fetch the next 16-byte aligned set of instructions. The predictive prefetch feature is available only for CPU instruction but not for CPU data and peripheral.

If the selected system clock speed is sufficiently low enough to access the Flash at zero Wait states, the predictive prefetch is detrimental and may be disabled.

9.7 Coherency Support

When a PFM programming event causes flash programming initiated by the Flash controller, the prefetch module invalidates all lines and the contents of the prefetch buffer. If a transaction is in progress, the invalidation occurs after completion. When programming or erasing a Flash page, a read of that Flash page causes the transaction to stall until the erase or program event completes.

The prefetch module provides two methods for coherency control:

- Auto invalidate via I/D/A CHECOH
- Manual invalidate via I/D/A CHEINV

The user can choose to auto invalidate the each/any cache on a PFM programming event by setting *CHECOH = 1. This is the safest option. However, the user has the option to never auto invalidate each/any cache by setting *CHECOH = 0.

In addition to using *CHECOH, use *CHEINV as an alternate invalidate method to invalidate each/any cache manually. If using *CHEINV to manually invalidate each/any cache due to a PFM programming event, stop all instruction/data fetches from the desired Flash, set *CHEINV, wait for it to clear and, then, start the programming sequence. When using *CHEINV to invalidate each/any cache for reasons other than programming, it can be set at any time but only takes effect after any pending transactions complete.

9.8 Effects of Reset

9.8.1 On Reset

Upon a device Reset, the following occurs:

- All lines are invalidated
- All tag bits are cleared

9.8.2 After Reset

The module operates as per the values in the CHECON register. See the *CHECON* register from Related Links.

Related Links

[9.12.1. CHECON](#)

9.9 Error Conditions

The prefetch module handles and reports information about two error types:

- ECC Double-bit Error Detected (DED)
- ECC Single-bit Error Corrected (SEC)

The user can enable and disable the ECC Error detection logic using the configuration bits, ECCCTL[1:0] (CFGCON0/DEVCFG0[29:28]).

The ECC logic increases the read access delay from the PFM. Depending on the frequency of the system clock, the Wait states can be different between ECC-enabled and ECC-disabled.

Note: ECC errors are captured for predictive prefetch reads of the PFM. However, do not report those errors until, and unless, the system starts using that data.

9.9.1 ECC Double-bit Error Detected (DED)

A read from the Flash memory that results in a PFM ECC DED causes the Prefetch module to return a bus exception error to the initiator. If that initiator is the CPU, it recognizes the bus exception error, prevents the instruction from executing, or read data from loading and generates an exception using the bus exception error vector.

When an ECC DED error occurs, the PFMDDED bit (CHESTAT[27]) is set. The exception handling code can, then, check this bit to determine whether the PFM ECC DED event is causing an exception or not. The exception handler clears this bit in software.

Note: CPU instructions or data prefetched from the PFM is always loaded into the Prefetch module, even if a DED error is generated. The Prefetch module line containing the DED data is tagged as valid until the line is replaced.

9.9.2 ECC Single Error Corrected (SEC)

A PFM ECC SEC event is not a critical error and, as such, is reported through an interrupt. The user has the option to enable or disable this interrupt through the PFMSECEN bit (CHECON[7]). The data in the Prefetch module is correct, and no further ECC events are generated for addresses that hit the data line as long as that data is in the Prefetch module.

Each read that returns from the PFM with an ECC SEC status causes the PFMSECCNT[7:0] bits (CHESTAT[7:0]) to decrement by one. If PFMSECCNT[7:0] is zero and a PFM ECC SEC event occurs, the PFMSEC bit (CHESTAT[26]) is set and an interrupt is generated. Therefore, the PFMSECCNT[7:0] bits must be set to the number of PFM ECC SEC events desired for an interrupt minus 1. For example, to generate an interrupt after five PFM ECC SEC events, PFMSECCNT[7:0] must be set to four ('00000100'). The Prefetch module does not reload the PFMSECCNT[7:0] bits when it reaches zero. Software must write the desired count each time it services the PFMSEC interrupt.

Software can generate an ECC SEC interrupt by setting the PFMSECEN bit, then setting the PFMSEC bit. If the PFMSEC bit is already set when PFMSECEN is set, the Prefetch module will also generate an ECC SEC interrupt. The ECC SEC interrupt persists as long as the PFMSECEN and PFMSEC bits remain set. See *PCACHE Interrupt* in the *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

[10.2. Nested Vector Interrupt Controller \(NVIC\)](#)

9.10 Operation in Power-saving Modes

9.10.1 Sleep Mode

When the device enters the Sleep mode, the Prefetch module is disabled and placed into a low-power state where no clocking occurs in the module.

9.10.2 Idle Mode

When the device enters the Idle mode, iCache, Prefetch and dCache clocks are internally gated-off, the aCache (peripheral data) clock remains functional for peripheral accesses and the CPU stops executing code. Any outstanding prefetch completes before the Prefetch module stops its clock through automatic clock gating.

9.10.3 Debug Mode

The behavior of the Prefetch module is unaltered in the Debug mode.

9.11 Register Summary (PCHE)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CHECON	7:0	PFMSECEN		PREFEN[1:0]		PFMWS[3:0]			
		15:8				CHEPERF				ADRWS
		23:16		ACHEINV	DCHEINV	ICHEINV		ACHECOH	DCHECOH	ICHECOH
		31:24						ACHEEN	DCHEEN	ICHEEN
0x04 ... 0x0F	Reserved									
0x10	CHESTAT	7:0	PFMSECCNT[7:0]							
		15:8								
		23:16								
		31:24					PFMEDD	PFMSEC		
0x14 ... 0x1F	Reserved									
0x20	CHEHIT	7:0	CHEHIT[7:0]							
		15:8	CHEHIT[15:8]							
		23:16	CHEHIT[23:16]							
		31:24	CHEHIT[31:24]							
0x24 ... 0x2F	Reserved									
0x30	CHEMIS	7:0	CHEMIS[7:0]							
		15:8	CHEMIS[15:8]							
		23:16	CHEMIS[23:16]							
		31:24	CHEMIS[31:24]							

9.12 Register Description

The CHECON and CHESTAT registers in the Register Summary table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.

The following are the description of the legends:

- R = Readable bit
- W = Writable bit
- S = Settable bit
- C = Clearable bit
- HC = Hardware Cleared
- HS = Hardware Set

Related Links

[6.1.9. CLR, SET and INV Registers](#)

9.12.1 CHECON - Prefetch Module Control Register

Name: CHECON
Offset: 0x00
Reset: 0x0700010F
Property: -

Bit	31	30	29	28	27	26	25	24
						ACHEEN	DCHEEN	ICHEEN
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	23	22	21	20	19	18	17	16
		ACHEINV	DCHEINV	ICHEINV		ACHECOH	DCHECOH	ICHECOH
Access		R/S/HC	R/S/HC	R/S/HC		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
				CHEPERF				ADRWS
Access				R/W				R
Reset				0				0

Bit	7	6	5	4	3	2	1	0
	PFMSECEEN		PREFEN[1:0]			PFMWS[3:0]		
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	1	1	1	1

Bit 26 – ACHEEN Peripheral Data Cache Enable bit

Value	Description
1	Caching is enabled
0	Caching is disabled (and all lines invalidated)

Bit 25 – DCHEEN Data Cache Enable bit

Value	Description
1	Caching is enabled
0	Caching is disabled (and all lines invalidated)

Bit 24 – ICHEEN Instruction Data Cache Enable bit

Value	Description
1	Caching is enabled
0	Caching is disabled (and all lines invalidated)

Bit 22 – ACHEINV Manual Invalidate Control for Peripheral Data Cache

Note: The hardware auto clears this bit when cache invalidate completes. Bits may clear at different times.

Value	Description
1	Force invalidate cache/invalidate busy
0	Cache invalidation follows ACHECOH/invalid complete

Bit 21 – DCHEINV Manual Invalidate Control for Data Cache

Note: The hardware auto clears this bit when cache invalidate completes. Bits may clear at different times.

Value	Description
1	Force invalidate cache/invalidate busy

Value	Description
0	Cache invalidation follows DCHECOH/invalid complete

Bit 20 – ICHEINV Manual Invalidate Control for Instruction Cache

Notes:

1. The Predictive Prefetch Buffer (PFB) is included with iCache invalidate.
2. The hardware auto clears this bit when cache invalidate completes. Bits may clear at different times.

Value	Description
1	Force invalidate cache/invalidate busy
0	Cache invalidation follows ICHECOH/invalid complete

Bit 18 – ACHECOH Auto Cache Coherency Control for Peripheral Data Cache

Note: ACHECOH must be stable before initiation of programming to ensure correct invalidation of data.

Value	Description
1	Auto invalidate cache on a programming event
0	No auto invalidated cache on a programming event

Bit 17 – DCHECOH Auto Cache Coherency Control for Data Cache

Note: DCHECOH must be stable before initiation of programming to ensure correct invalidation of data.

Value	Description
1	Auto invalidate cache on a programming event
0	No auto invalidated cache on a programming event

Bit 16 – ICHECOH Auto Cache Coherency Control for Instruction Cache

Note: ICHECOH must be stable before initiation of programming to ensure correct invalidation of data.

Value	Description
1	Auto invalidate cache on a programming event
0	No auto invalidated cache on a programming event

Bit 12 – CHEPERF Cache Performance Counters Enable

Note: Performance counters are reset on 0 to 1 transition of this bit.

Value	Description
1	Performance counters is enabled
0	Performance counters is disabled

Bit 8 – ADRWS Address Wait State

Address Wait state is hard wired to '0' for higher performance at the clock frequency.

Bit 7 – PFMSECEN Flash Single-bit Error Corrected (SEC) Interrupt Enable bit

Value	Description
1	Generate an interrupt when PFMSEC is set
0	Do not generate an interrupt when PFMSEC is set

Bits 5:4 – PREFEN[1:0] Instruction Predictive Prefetch Enable

Value	Description
01	Instruction predictive prefetch enabled for cacheable regions only
00	Instruction predictive prefetch disabled

Note: Other values are unavailable.

Bits 3:0 – PFMWS[3:0] PFM Access Time Defined in Terms of SYSCLK Wait States bits

Total Flash Wait states are ADRWS + PFMWS.

Value	Description
1111	Fifteen Wait states
1110	Fourteen Wait states
...	
0001	One Wait state
0000	Zero Wait state

Notes:

1. This is not the Wait state seen by the CPU.
2. For the Wait states to SYSCLK relationship, see *Electrical Characteristics* from Related Links.

Related Links[43. Electrical Characteristics](#)

9.12.2 CHESTAT - Prefetch Module Status Register

Name: CHESTAT
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
					PFMDDED	PFMSEC		
Access					HS, R/C	HS, R/W		
Reset					0	0		

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PFMSECCNT[7:0]							
Access	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W	HS, HC, R/W
Reset	0	0	0	0	0	0	0	0

Bit 27 – PFMDDED Flash Double-bit Error Detected (DED) Status bit
This bit is set in hardware and can only be cleared (set to '0') in software.

Value	Description
1	A DED error has occurred
0	A DED error has not occurred

Bit 26 – PFMSEC Flash Single-bit Error Corrected (SEC) Status bit
Note: The PCACHE interrupt event reports the error event to the CPU. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Value	Description
1	A SEC error occurred when PFMSECCNT[7:0] equals to zero
0	A SEC error has not occurred

Bits 7:0 – PFMSECCNT[7:0] Flash SEC Count bits
The count value decreases by '1' each time an SEC error occurs. Holds at zero. When an SEC error occurs when PFMSECCNT[7:0] is zero, the PFMSEC status bit is set. If PFMSECEN is also set, a prefetch module interrupt event is generated.

Related Links

[10.2. Nested Vector Interrupt Controller \(NVIC\)](#)

9.12.3 CHEHIT – Prefetch Module Hit Statistics Register

Name: CHEHIT
Offset: 0x20
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CHEHIT[31:24]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHEHIT[23:16]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHEHIT[15:8]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHEHIT[7:0]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHEHIT[31:0] Instruction Cache Hit Count bits

When CHECON.CHEPERF = 1, CHEHIT increments once per iCache or Predictive Prefetch Buffer (PFB) hit.

Note: CHEHIT is Reset on the '0' to '1' transition of CHECON.CHEPERF.

9.12.4 CHEMIS – Prefetch Module Miss Statistics Register

Name: CHEMIS
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CHEMIS[31:24]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHEMIS[23:16]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHEMIS[15:8]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHEMIS[7:0]							
Access	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHEMIS[31:0] Instruction Cache Miss Count bits

When CHECON.CHEPERF = 1, CHEMIS increments once per iCache or Predictive Prefetch Buffer (PFB) miss.

Note: CHEMIS is Reset on the '0' to '1' transition of CHECON.CHEPERF.

10. Processor and Architecture

10.1 Cortex M4 Processor

The ARM®Cortex™-M4 processor is a high performance 32-bit processor designed for the microcontroller market. It offers the following significant benefits to developers:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Efficient processor core, system, and memories
- Ultra low-power consumption with integrated sleep modes
- Platform security robustness, with integrated memory protection unit (MPU).

The implemented ARM Cortex-M4 is revision r0p1

For additional information, refer to <http://www.arm.com>

The Cortex-M4 processor is built on a high-performance processor core with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including IEEE 754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic, and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M4 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M4 processor implements a version of the Thumb instruction set based on Thumb®-2 technology, ensuring high code density and reduced program memory requirements. The Cortex-M4 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M4 processor closely integrates a configurable *Nested Vector Interrupt Controller* (NVIC), to deliver industry-leading interrupt performance. The NVIC includes a *Non-Maskable interrupt* (NMI), and provides up to 8 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of *Interrupt Service Routines* (ISRs), dramatically reducing interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tail-chain optimization also significantly reduces the overhead when switching from one ISR to another.

To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state.

10.1.1 System Level Interface

The Cortex-M4 processor provides multiple interfaces using AMBA technology to provide high-speed, low-latency memory accesses. It supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks, and thread-safe Boolean data handling.

The Cortex-M4 processor has a Memory Protection Unit (MPU) that provides fine grain memory control, enabling applications to utilize multiple privilege levels, separating and protecting code, data, and stack on a task-by-task basis. Such requirements are becoming critical in many embedded applications such as automotive.

10.1.2 Integrated Configurable Debug

The Cortex-M4 processor implements a complete hardware debug solution. This provides high system visibility of the processor and memory through a 2-pin *Serial Wire Debug* (SWD) port that is ideal for microcontrollers and other small package devices.

For system trace the processor integrates an *Instrumentation Trace Macrocell* (ITM) alongside data watchpoints and a profiling unit. The *Embedded Trace Macrocell* (ETM) delivers unrivaled instruction trace capture in an area far smaller than traditional trace units, enabling many low cost MCUs to implement full instruction trace for the first time.

To enable simple and cost-effective profiling of the system events these generate, a stream of software-generated messages, data trace, and profiling information is exported over three different ways:

- Output off chip using the TPIU, through a single pin, called *Serial Wire Viewer* (SWV). Limited to ITM system trace
- Output off chip using the TPIU, through a 4-bit pin interface. Bandwidth is limited
- Internally stored in RAM, using the CoreSight ETB. Bandwidth is then optimal but capacity is limited

The *Flash Patch and Breakpoint Unit* (FPB) provides up to 8 hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to 8 words in the program code in the Code memory region. This enables applications stored on a non-erasable, ROM-based microcontroller to be patched if a small programmable memory, for example Flash, is available in the device. During initialization, the application in ROM detects, from the programmable memory, whether a patch is required. If a patch is required, the application programs the FPB to remap a number of addresses. When those addresses are accessed, the accesses are redirected to a remap table specified in the FPB configuration, which means the program in the non-modifiable ROM can be patched.

10.1.3 Cortex-M4 Processor Features and Configuration

- Thumb® instruction set combines high code density with 32-bit performance
- IEEE 754-compliant single-precision Floating Point Unit (FPU)
- Integrated sleep modes for low power consumption
- Fast code execution permits slower processor clock or increases Sleep mode time
- Hardware division and fast digital-signal-processing orientated multiply accumulate
- Saturating arithmetic for signal processing
- Deterministic, high-performance interrupt handling for time-critical applications
- Memory Protection Unit (MPU) for safety-critical applications
- Extensive debug and trace capabilities: Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing and code profiling.

Features	Cortex-M4 Options	PIC32CX-BZ2 Configuration
Interrupts	1 to 240	40
Number of priority bits	3 to 8	3 = eight levels of priority
Data endianness	Little-endian or big-endian	Little-endian
SysTick Timer calibration value	—	0x80000000
MPU	Present or Not present	Present

.....continued

Features	Cortex-M4 Options	PIC32CX-BZ2 Configuration
Debug support level	0 = No debug. No DAP, breakpoints, watchpoints, Flash patch or halting debug 1 = Minimum debug. Two breakpoints, one watchpoint, no Flash patch 2 = Full debug minus DWT data matching 3 = Full debug plus DWT data matching	3 = Full debug plus DWT data matching
Trace support level	0 = No trace. No ETM, ITM or DWT triggers and counters 1 = Standard trace. ITM and DWT triggers and counters, but no ETM 2 = Full trace. Standard trace plus ETM 3 = Full trace plus HTM port	2 = Full trace. Standard trace plus ETM
JTAG	Present or Not present	Not present
Bit Banding	Present or Not present	Not present
FPU	Present or Not present	Present

10.1.4 Cortex-M4 Core Peripherals

Nested Vectored Interrupt Controller	The Nested Vector Interrupt Controller (NVIC) is an embedded interrupt controller that supports low latency interrupt processing.
System Control Block	The System Control Block (SCB) is the programmers model interface to the processor. It provides system implementation information and system control, including configuration, control and reporting of system exceptions. Refer to the <i>Cortex-M4 Technical Reference Manual</i> for more details (http://www.arm.com).
System Timer	The system timer, SysTick, is a 24-bit countdown timer. Use this as a Real-Time Operating System (RTOS) tick timer or as a simple counter. The SysTick timer runs on the processor clock and it does not decrement when the processor is halted for debugging. Refer to the <i>Cortex-M4 Technical Reference Manual</i> for more details (http://www.arm.com).
Memory Protection Unit	The Memory Protection Unit (MPU) improves system reliability by defining the memory attributes for different memory regions. It provides up to eight different regions and an optional predefined background region. Refer to the <i>Cortex-M4 Technical Reference Manual</i> for more details (http://www.arm.com).
Floating-Point Unit	The Floating Point Unit (FPU) provides IEEE 754-compliant operations on single-precision, 32-bit, floating-point values. Refer to the <i>Cortex-M4 Technical Reference Manual</i> for more details (http://www.arm.com).

10.1.5 Cortex-M4 Address Map

Address	Core Peripheral
0xE000E008-0xE000E00F	System control block
0xE000E010-0xE000E01F	System timer
0xE000E100-0xE000E4EF	Nested Vectored Interrupt Controller
0xE000ED00-0xE000ED3F	System control block
0xE000ED90-0xE000ED93	MPU Type Register
0xE000ED94-0xE000EDB8	Memory Protection Unit
0xE000EF00-0xE000EF03	Nested Vectored Interrupt Controller
0xE000EF30-0xE000EF44	Floating Point Unit

10.2 Nested Vector Interrupt Controller (NVIC)

10.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the PIC32CX-BZ2 family devices supports 40 interrupts with eight different priority levels. For more details, refer to the *Cortex-M4 Technical Reference Manual* (www.arm.com).

10.2.2 Interrupt Line Mapping

Each of the interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a '1' to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing '1' to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

Depending on their criticality, the interrupt requests for one peripheral are either ORed together on system level, generating one interrupt or directly connected to an NVIC interrupt lines. This is described in the table below.

An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Module	Source	Line
EIC NMI – External Interrupt Control	NMI	NMI
RTCC – Real-Time Counter and Calendar	CMP A 0	0
	CMP A 1	
	CMP A 2	
	CMP A 3	
	OVF A	
	PER A 0	
	PER A 1	
	PER A 2	
	PER A 3	
	PER A 4	
	PER A 5	
	PER A 6	
	PER A 7	
	TAMPER A	
EIC – External Interrupt Controller	EXTINT 0	1
	EXTINT 1	
	EXTINT 2	
	EXTINT 3	
FREQM – Frequency Meter	DONE	2
Flash Subsystem	Flash Controller	3
	PFW	
	PCACHE	
PORT-A	PortA Input Change Interrupt	4
PORT-B	PortB Input Change Interrupt	5

.....continued

Module	Source	Line
DMAC – Direct Memory Access Controller	SUSP 0..3	6
	TCMPL 0..3	
	TERR 0..3	
	SUSP 4..15	7
	TCMPL 4..15	
	TERR 4..15	
EVSYS – Event System Interface	EVD 0..3	8
	OVR 0..3	
	EVD 4..11	9
	OVR 4..11	
PAC – Peripheral Access Controller	ERR	10
RAM ECC	SINGLEE-0	11
	DualE-1	
SERCOM0 – Serial Communication Interface 0 ⁽¹⁾ Order: USART, I2CM, I2CS, SPI	0	12
	1	
	2	
	3	
	4	
	5	
	7	
SERCOM1 – Serial Communication Interface 1 ⁽¹⁾ Order: USART, I2CM, I2CS, SPI	0	13
	1	
	2	
	3	
	4	
	5	
	7	
SERCOM2 – Serial Communication Interface 2 ⁽¹⁾ Order: USART, I2CM, I2CS, SPI	0	14
	1	
	2	
	3	
	4	
	5	
	7	
SERCOM3 – Serial Communication Interface 3 ⁽¹⁾ Order: USART, I2CM, I2CS, SPI	0	15
	1	
	2	
	3	
	4	
	5	
	7	

.....continued

Module	Source	Line
TCC0 – Timer Counter Control 0	CNT A	16
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0	
	MC 1	
	MC 2	
	MC 3	
	MC 4	
	MC 5	
TCC1 – Timer Counter Control 1	CNT A	17
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0	
	MC 1	
	MC 2	
	MC 3	
	MC 4	
	MC 5	
TCC2 – Timer Counter Control 2	CNT A	18
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0	
	MC 1	

.....continued

Module	Source	Line
TC0 – Basic Timer Counter 0	ERR A	19
	MC 0	
	MC 1	
	OVF	
TC1 – Basic Timer Counter 1	ERR A	20
	MC 0	
	MC 1	
	OVF	
TC2 – Basic Timer Counter 2	ERR A	21
	MC 0	
	MC 1	
	OVF	
TC3 – Basic Timer Counter 3	ERR A	22
	MC 0	
	MC 1	
	OVF	
ADCTRL	ADC_GIRQ	23
	ADC_DIRQ0, ADC_DIRQ1	
	ADC_AIRQ0, ADC_AIRQ1	
	ADC_FLT	34
	ADC_EOS	35
	ADC_BGVR_RDY	36
AC – Analog Comparators	COMP 0	24
	COMP 1	
	WIN 0	
AES – Advanced Encryption Standard	ENCCMP	25
	GFMCMP	
TRNG – True Random Generator	ISO	26
ICM – Integrity Check Monitor	ICM	27
QSPI – Quad SPI interface	QSPI	29
Wireless Radio	ZB_INT0	30
	BT_INT0	31
	BT_INT1	32
	ARBITER	33
	CLKI_WAKEUP_NMI	37
	BT_LC	38
	BT_RC	39

Note:

1. The integer number specified in the source refers to the respective bit position in the INTFLAG register of the respective peripheral.

10.3 High-Speed Bus System

10.3.1 Features

High-Speed Bus Matrix has the following features:

- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different Hosts to different Clients
- 32-bit data bus
- Operation at a one-to-one clock frequency with the bus Hosts

10.3.2 Configuration

Figure 10-1. Host-Client Relations High-Speed Bus Matrix

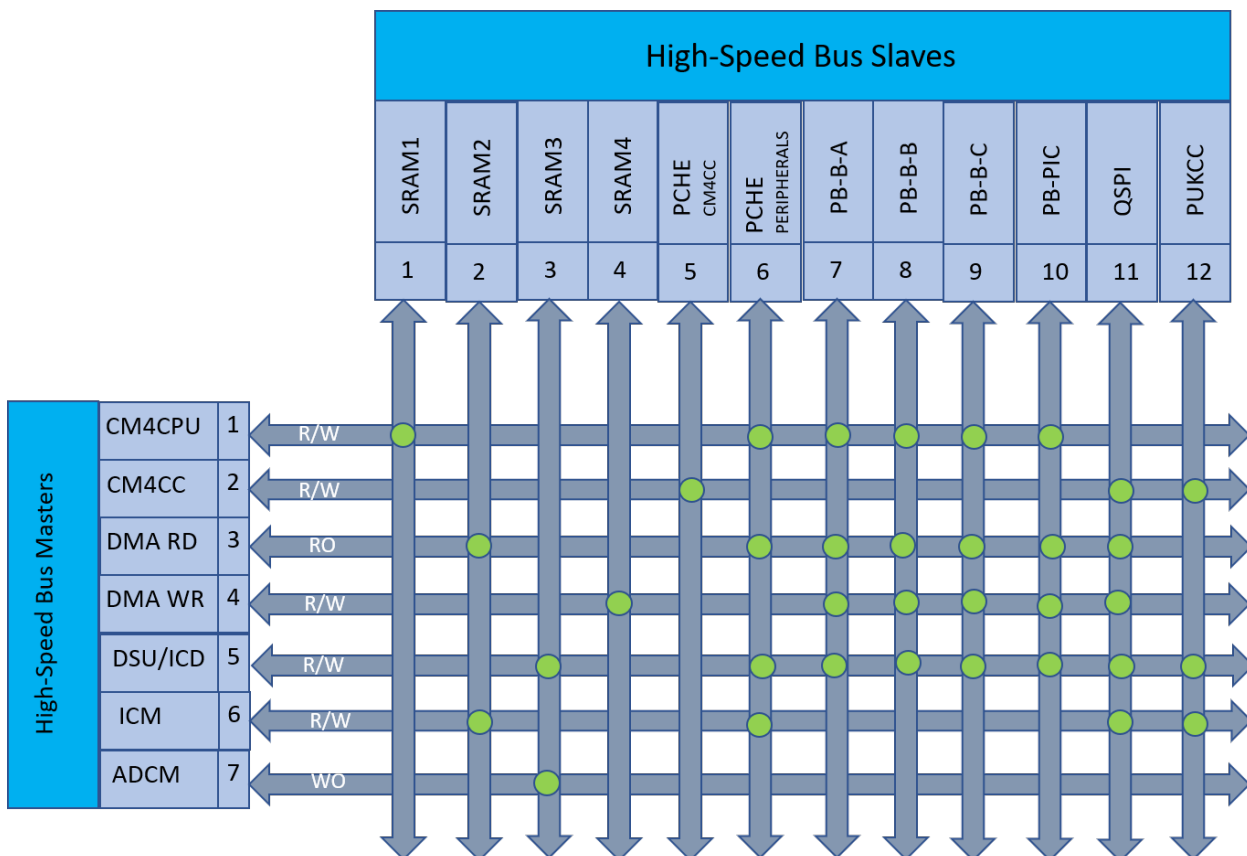


Table 10-1. High Speed Bus Matrix Host

High-Speed Bus Matrix Host	Host ID
CM4CPU - Cortex M4 Processor	1
CM4CC - Cortex-M Cache Controller	2
DMA RD - DMA-Read	3
DMA-WR - DMA-Write	4
DSU/ICD (private test mode only) - Device Service Unit/In-Chip Debugger	5
ICM - Integrity Check Monitor	6
ADCM - ADC Controller Module	7

Table 10-2. High-Speed Bus Matrix Client

High-Speed Bus Matrix Client	Client ID
SRAM1 - SRAM Port 1	1
SRAM2 - SRAM Port 2	2
SRAM3 - SRAM Port 3	3

.....continued

High-Speed Bus Matrix Client	Client ID
SRAM4 - SRAM Port 4	4
PCHE - Pre-fetch Cache of CM4CC	5
PCHE - Pre-fetch Cache of Peripherals	6
PB-B-A - Peripheral Bridge A	7
PB-B-B - Peripheral Bridge B	8
PB-B-C - Peripheral Bridge C	9
PB-PIC - Peripheral Bus PIC	10
QSPI - Quad SPI Interface	11
PUKCC - Public Key Cryptography Controller	12

11. Cortex M Cache Controller (CMCC)

11.1 Overview

The Cortex M Cache Controller provides an L1 cache to the Cortex M CPU. The CMCC sits transparently between the CPU and the cache leading to improved performance.

The CMCC interfaces with the CPU through the AHB and is connected to the APB bus interface for its configuration.

11.2 Features

- Physically addressed and physically tagged
- L1 data and instruction cache set to 4 KB
- L1 cache line size set to 16 Bytes
- L1 cache integrates 32-bit bus host interface
- Unified 4-Way set associative cache architecture
- Lock-Down feature, which allows cached to be locked per way
- Write through cache operations, read allocate
- Configurable as data and instruction Tightly Coupled Memory (TCM)
- Round Robin victim selection policy
- Event Monitoring, with one programmable 32-bit counter
- Cache Interface includes cache maintenance operations registers

11.3 Block Diagram

Figure 11-1. CMCC Block Diagram

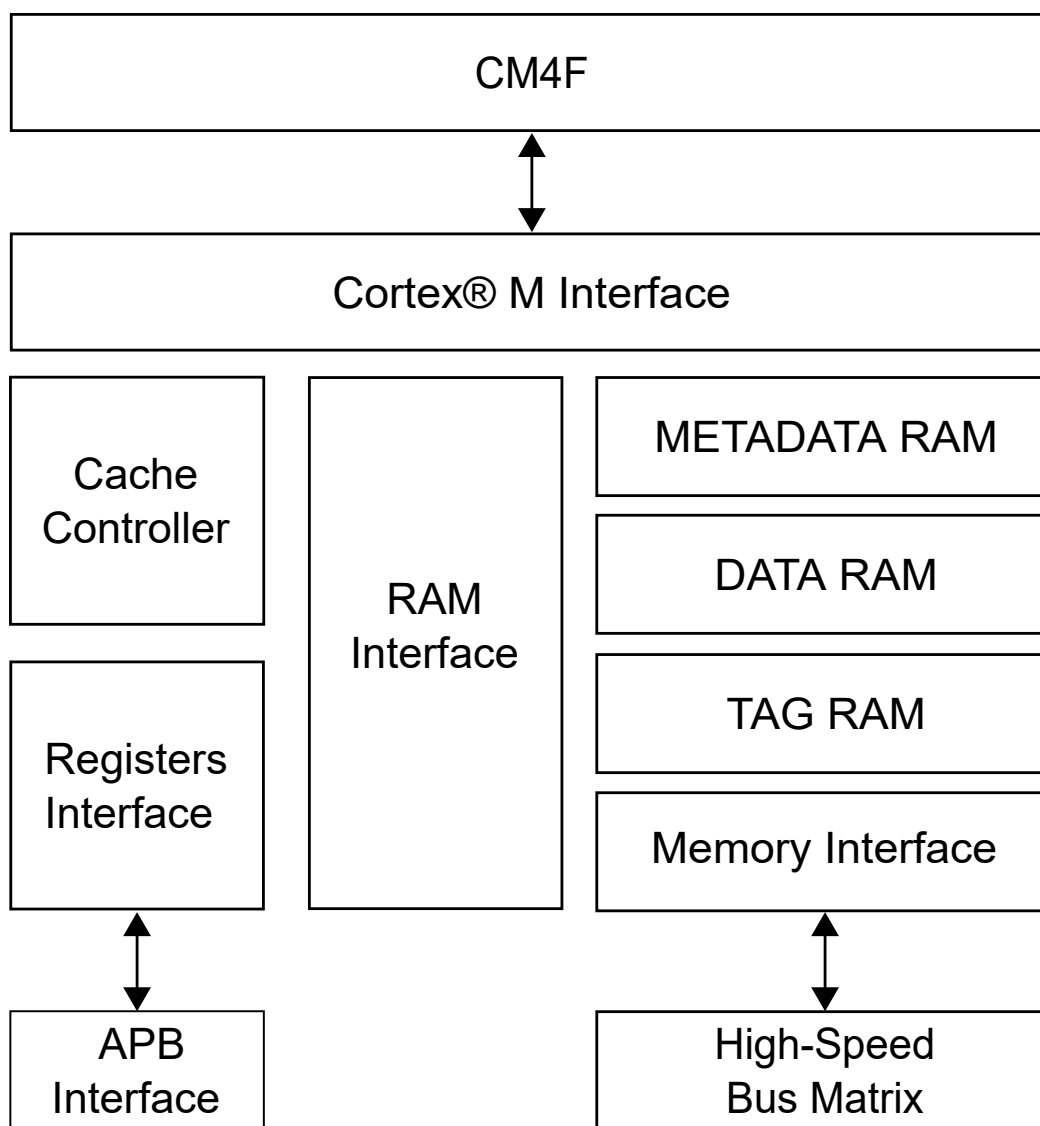
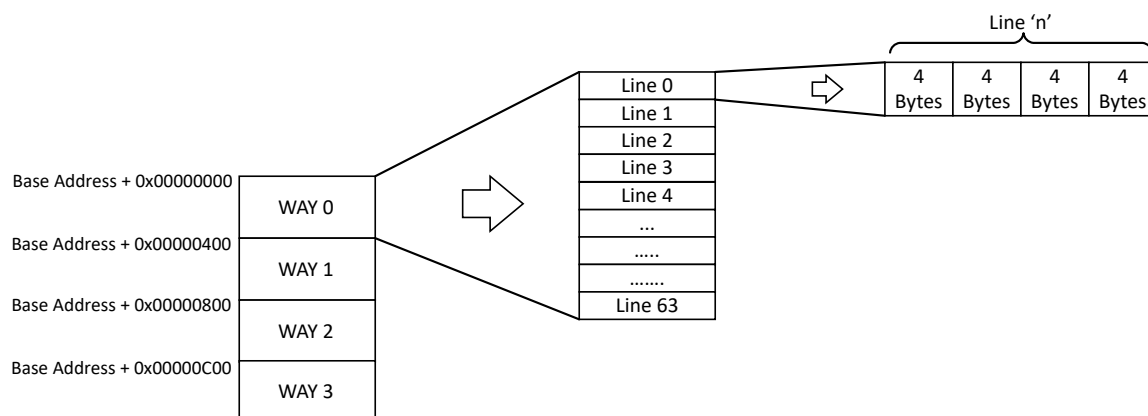


Figure 11-2. CMCC Organization



11.4 Signal Description

Not applicable.

11.5 Product Dependencies

Not applicable.

11.5.1 I/O Lines

Not applicable.

11.5.2 Power Management

The CMCC will continue to function as long as the CPU is not sleeping and the CMCC is enabled.

11.5.3 Clocks

Not applicable.

11.5.4 DMA

Not applicable.

11.5.5 Interrupts

Not applicable.

11.5.6 Events

Not applicable.

11.5.7 Debug Operation

When the CPU is halted in debug mode, the CMCC is halted. Any read access by the debugger in cached zones are not cached.

11.5.8 Register Access Protection

Not applicable.

11.5.9 Analog Connections

Not applicable.

11.6 Functional Description

11.6.1 Principle of Operation

11.6.2 Initialization and Normal Operation

On reset, the cache controller data entries are all invalidated, and the cache is disabled. The cache is transparent to processor operations. The cache controller is activated through the use of its configuration registers. The configuration interface is memory mapped in the APB bus.

Use the following sequence to enable the cache controller:

- Verify that the CMCC is disabled, reading the value of the SR.CSTS.
- Enable the CMCC by writing '1' in CTRL.CEN. The MODULE is disabled by writing a '0' in CTRL.CEN.

11.6.3 Change Cache Size

It is possible to change the cache size by writing to the Cache Size Configured By Software bits in the Cache Configuration register (CFG.CSIZESW).

Use the following sequence to change the cache size:

- Disable the CMCC controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN=0).
- Check the Cache Controller Status bit in the Cache Status register to verify that the CMCC is successfully disabled (SR.CSTS=0).
- Change CFG.CSIZESW to its new value.
- Enable the CMCC by writing CTRL.CEN=1.

11.6.4 Data Cache Disable

The Instructions alone can be cached by disabling the Data cache, as described in the following steps:

1. Disable the cache controller by writing a '0' to CTRL.CEN.
2. Check SR.CSTS to verify whether the CMCC is successfully disabled.
3. Write CFG.DCDIS = 1.
4. Enable the CMCC by writing CTRL.CEN = 1.

11.6.5 Instruction Cache Disable

The Data alone can be cached by disabling the Instruction cache, as described in the following steps:

1. Disable the cache controller by writing CTRL.CEN = 0.
2. Check SR.CSTS to verify that the CMCC is successfully disabled.
3. Write CFG.ICDIS = 1.
4. Enable the CMCC by writing CTRL.CEN = 1.

11.6.6 Cache Load and Lock

It is possible to lock a specific way for code optimization by writing the Lock Way register (LCKWAY.LCKWAY). The locked way will not be updated by the CMCC as part of cache operations.

The load and lock mechanism can be implemented to use cache memory in a deterministic way. Follow these steps to load and lock a way:

1. Disable cache controller by clearing the CTRL.CEN bit.
2. Invalidate the desired WAY line by line. This will reset the round robin algorithm of the invalidated line, that will become eligible for the next load operation.

3. Disable the Instruction cache, but keep the Data cache enabled.
4. Enable the cache by setting the CTRL.CEN bit.
5. Place the respective piece of code and/or data to the corresponding WAY due to simple LOAD operations. Loading the piece of code and/or data will force the cache to refill the previous invalidated line in the right way. No need to load all the bytes of the line, only the first byte. The cache will automatically refill the complete line.
6. Lock the specific WAY by setting LCKWAY.LCKWAY[3:0].
7. Re-enable the instruction cache. The locked WAY is now loaded and ready to operate. The remaining WAYS can be used as I-cache or D-cache as required.

11.6.7 Tightly Coupled Memory

Users can use a part of the cache as Tightly Coupled Memory (TCM). The cache size is determined by the Cache Size Configuration by Software bits in the Cache Configuration register (CFG.CSIZESW). The relation between cache and TCM is as given below:

TCM size = maximum Cache size – configured Cache size.

The TCM start address can be obtained from the product memory mapping. The cache memory starts first from the address followed by the TCM memory. Size of the Way is fixed and the number of ways varies according to the available size for the cache memory. See [8. Product Memory Mapping Overview](#).

Table 11-1. TCM Sizes

Max. Cache	Configured Cache	TCM Size
4 KB	4 KB	0 KB
4 KB	1 KB	3 KB
4 KB	2 KB	2 KB
4 KB	0 KB	4 KB

The TCM is also accessible in its maximum size when the CMCC is disabled. The TCM does not need to be locked in order to operate.

Note: Writing into the cache DATA RAM region through the CPU can overwrite the valid cache lines. This can result in data corruption when the cache controller is accessing the data for cache transactions. Access the DATA RAM region only after configuring it as TCM.

Related Links

[8. Product Memory Mapping Overview](#)

11.6.8 Cache Maintenance

11.6.8.1 Cache Invalidate by Line Operation

When an invalidate by line command is issued, the CMCC resets the valid bit information of the decoded cache line. As the line is no longer valid, the replacement counter points to that line.

- Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
- Check SR.CSTS to verify that the CMCC is successfully disabled.
- Perform an invalidate by line by writing the set {index,way} in the Cache Maintenance 1 register (MAINT1.INDEX, MAINT1.WAY).
- Enable the CMCC by writing a '1' to CTRL.CEN.

11.6.8.2 Cache Invalidate All Operation

Use the following sequence to invalidate all cache entries.

- Disable the cache controller by writing a zero to the Cache Enable bit in the Cache Control register (CTRL.CEN).
- Check SR.CSTS to verify that the CMCC is successfully disabled.
- Perform a full invalidate operation by writing a '1' to the Cache Controller Invalidate All bit in the Cache Maintenance 0 register (MAINT0.INVALL).
- Enable the CMCC by writing a '1' to CTRL.CEN.

11.6.9 Cache Performance Monitoring

The Cortex M cache controller includes a programmable monitor/32-bit counter. The monitor can be configured to count the number of clock cycles, the number of data hit or the number of instruction hit.

It is important to know that the Cortex-M4 processor prefetches instructions ahead of execution. It performs only 32-bit read access on the Instruction Bus, which means:

- One arm instruction is fetched per bus access
- Two thumb instructions are fetched per bus access

As a consequence, two thumb instructions (e.g., `NOP`) need one bus access, which results in the HIT counter incrementing by 1.

Use the following sequence to activate the counter:

- Configure the monitor counter by writing the MCFG.MODE.
 - CYCLE_COUNT is used to increment the counter along with the program counter, to count the number of cycles.
 - IHIT_COUNT is the instruction Hit counter, which increments the counter when there is a hit for the instruction in the cache.
 - DHIT_COUNT is the data Hit counter which increments the counter when there is a hit for the data in the cache.
- Enable the counter by writing a '1' to the Cache Controller Monitor Enable bit in the Cache Monitor Enable register (MEN.MENABLE).
- If required, reset the counter by writing a '1' to the Cache Controller Software Reset bit in the Cache Monitor Control register (MCTRL.SWRST).
- Check the value of the monitor counter by reading the MSR.EVENT_CNT bit field.

11.7 DEBUG Mode

In Debug mode, TAG and METADATA RAM blocks content is read/written through the AHB bus interface if the CMCC is disabled. When the CMCC is enabled, the TAG and METADATA RAM blocks are non readable.

Debug access has the same R/W properties as the CPU access for the DATA RAM block.

The TAG, METADATA and DATA RAM blocks' R/W properties are summarized in RAM Properties. See *RAM Properties* from the Related Links.

Use the following sequence to perform read access with the Debugger to the three RAM blocks:

- Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
- Check the Cache Controller Status bit in the Cache Status register (SR.CSTS) to verify that the CMCC is successfully disabled.
- Perform a read or write access through Debugger:
 - @ CMCC_AHB_ADDR for DATA RAM,
 - @ CMCC_AHB_ADDR_TAG for TAG RAM,

- @ CMCC_AHB_ADDR_MTDATA for METADATA RAM.
- If a write access has been performed in the TAG, METADATA, or DATA RAM in the cache section, an invalid operation must be performed before re-enabling the CMCC.

Related Links

[11.8. RAM Properties](#)

11.8 RAM Properties

The following table shows the different access properties of the three RAM blocks, according the different modes described in the previous chapters.

Table 11-2. Access to RAM

Access Condition	DATA RAM	TAG RAM	METADATARAM
CPU access when CMCC DISABLED	R/W	no R/W - hardfault	no R/W - hardfault
CPU access when CMCC ENABLED	CACHE section configured: R/W ⁽¹⁾ TCM section configured: R/W	no R/W - hardfault	no R/W - hardfault
Debugger access when CMCC DISABLED	R/W	R/W	R/W
Debugger access when CMCC ENABLED	CACHE section configured: R/W ⁽¹⁾ TCM section configured: R/W	no R/W	no R/W

Note:

1. A write operation in this zone can corrupt the coherency of the cache. An invalidate operation may be needed.

11.9 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TYPE	7:0	LCKDOWN	WAYNUM[1:0]		RRP	LRUP	RANDP	GCLK	AP
		15:8			CLSIZE[2:0]			CSIZE[2:0]		
		23:16								
		31:24								
0x04	CFG	7:0		CSIZESW[2:0]				DCDIS	ICDIS	
		15:8								
		23:16								
		31:24								
0x08	CTRL	7:0								CEN
		15:8								
		23:16								
		31:24								
0x0C	SR	7:0								CSTS
		15:8								
		23:16								
		31:24								
0x10	LCKWAY	7:0					LCKWAY[3:0]			
		15:8								
		23:16								
		31:24								
0x14 ... 0x1F	Reserved									
0x20	MAINT0	7:0								INVALL
		15:8								
		23:16								
		31:24								
0x24	MAINT1	7:0	INDEX[3:0]				INDEX[7:4]			
		15:8								
		23:16								
		31:24	WAY[3:0]							
0x28	MCFG	7:0						MODE[1:0]		
		15:8								
		23:16								
		31:24								
0x2C	MEN	7:0								MENABLE
		15:8								
		23:16								
		31:24								
0x30	MCTRL	7:0								SWRST
		15:8								
		23:16								
		31:24								
0x34	MSR	7:0	EVENT_CNT[7:0]							
		15:8	EVENT_CNT[15:8]							
		23:16	EVENT_CNT[23:16]							
		31:24	EVENT_CNT[31:24]							

11.10 Register Description

11.10.1 Cache Type

Name: TYPE
Offset: 0x00
Reset: 0x000012D2
Property: R

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			CLSIZE[2:0]			CSIZE[2:0]		
Access			R	R	R	R	R	R
Reset			0	1	0	0	1	0
Bit	7	6	5	4	3	2	1	0
	LCKDOWN	WAYNUM[1:0]		RRP	LRUP	RANDP	GCLK	AP
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	0	0	1	0

Bits 13:11 – CLSIZE[2:0] Cache Line Size

This field configures the Cache Line Size.

Value	Name	Description
0x2	CLSIZE_16B	Cache Line Size is 16 bytes
0x3–0x7	—	Reserved

Bits 10:8 – CSIZE[2:0] Cache Size

This bit field configures the cache size.

Value	Name	Description
0x0	CSIZE_1KB	Cache Size is 1 KB
0x1	CSIZE_2KB	Cache Size is 2 KB
0x2	CSIZE_4KB	Cache Size is 4 KB
0x3–0x7	—	Reserved

Bit 7 – LCKDOWN Lock Down Supported

Writing a '0' to this bit disables the Lock Down feature.

Writing a '1' to this bit enables the Lock Down feature.

Bits 6:5 – WAYNUM[1:0] Number of Way

This bit field configures the mapping of the cache.

Value	Name	Description
0x0	DMAPPED	Direct Mapped Cache
0x1	ARCH2WAY	2-WAY set associative
0x2	ARCH4WAY	4-WAY set associative
0x3	ARCH8WAY	8-WAY set associative

Bit 4 – RRP Round Robin Policy Supported

Writing a '0' to this bit disables Round Robin Policy.

Writing a '1' to this bit enables Round Robin Policy.

Bit 3 – LRUP Least Recently Used Policy Supported

Writing a '0' to this bit disables the Least Recently Used Policy Supported.

Writing a '1' to this bit enables the Least Recently Used Policy Supported.

Bit 2 – RANDP Random Selection Policy Supported

Writing a '0' to this bit disables the Random Selection Policy Supported.

Writing a '1' to this bit enables the Random Selection Policy Supported.

Bit 1 – GCLK Dynamic Clock Gating

Writing a '0' to this bit disables the Dynamic Clock Gating feature.

Writing a '1' to this bit enables the Dynamic Clock Gating feature.

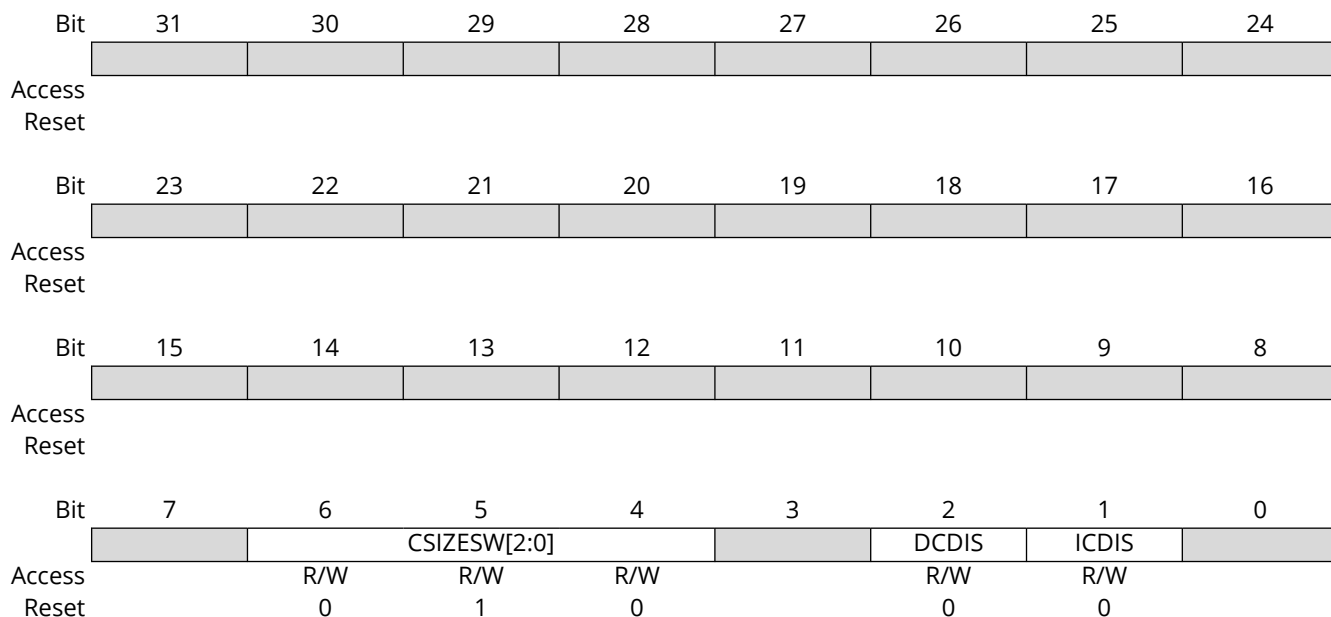
Bit 0 – AP Access Port Access Allowed

Writing a '0' to this bit disables the Access Port Access Allowed.

Writing a '1' to this bit enables the Access Port Access Allowed.

11.10.2 Cache Configuration

Name: CFG
Offset: 0x04
Reset: 0x00000020
Property: R/W



Bits 6:4 – CSIZESW[2:0] Cache Size Configured by Software

This field configures the cache size.

Value	Name	Description
0x0	CONF_CSIZ_1KB	The Cache Size is configured to 1KB
0x1	CONF_CSIZ_2KB	The Cache Size is configured to 2KB
0x2	CONF_CSIZ_4KB	The Cache Size is configured to 4KB
0x3	CONF_CSIZ_8KB	The Cache Size is configured to 8KB
0x4	CONF_CSIZ_16KB	The Cache Size is configured to 16KB
0x5	CONF_CSIZ_32KB	The Cache Size is configured to 32KB
0x6	CONF_CSIZ_64KB	The Cache Size is configured to 64KB
0x7		Reserved

Bit 2 – DCDIS Data Cache Disable

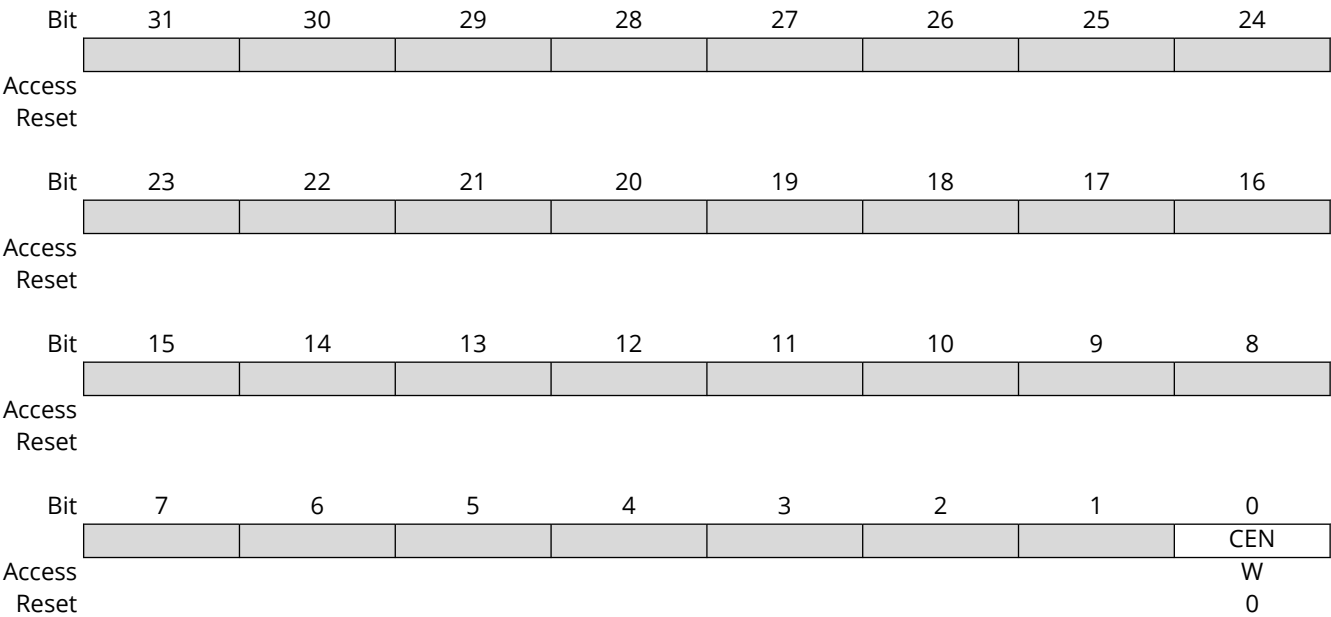
Writing a '0' to this bit enables data caching.
Writing a '1' to this bit disables data caching.

Bit 1 – ICDIS Instruction Cache Disable

Writing a '0' to this bit enables instruction caching.
Writing a '1' to this bit disables instruction caching.

11.10.3 Cache Control

Name: CTRL
Offset: 0x08
Reset: 0x00000000
Property: Write-only



Bit 0 – CEN Cache Controller Enable
Writing a '0' to this bit disables the CMCC.
Writing a '1' to this bit enables the CMCC.

11.10.4 Cache Status

Name: SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CSTS
Access								R
Reset								0

Bit 0 – CSTS Cache Controller Status
Writing a ‘0’ to this bit disables the CMCC.
Writing a ‘1’ to this bit enables the CMCC.

11.10.5 Cache Lock per Way

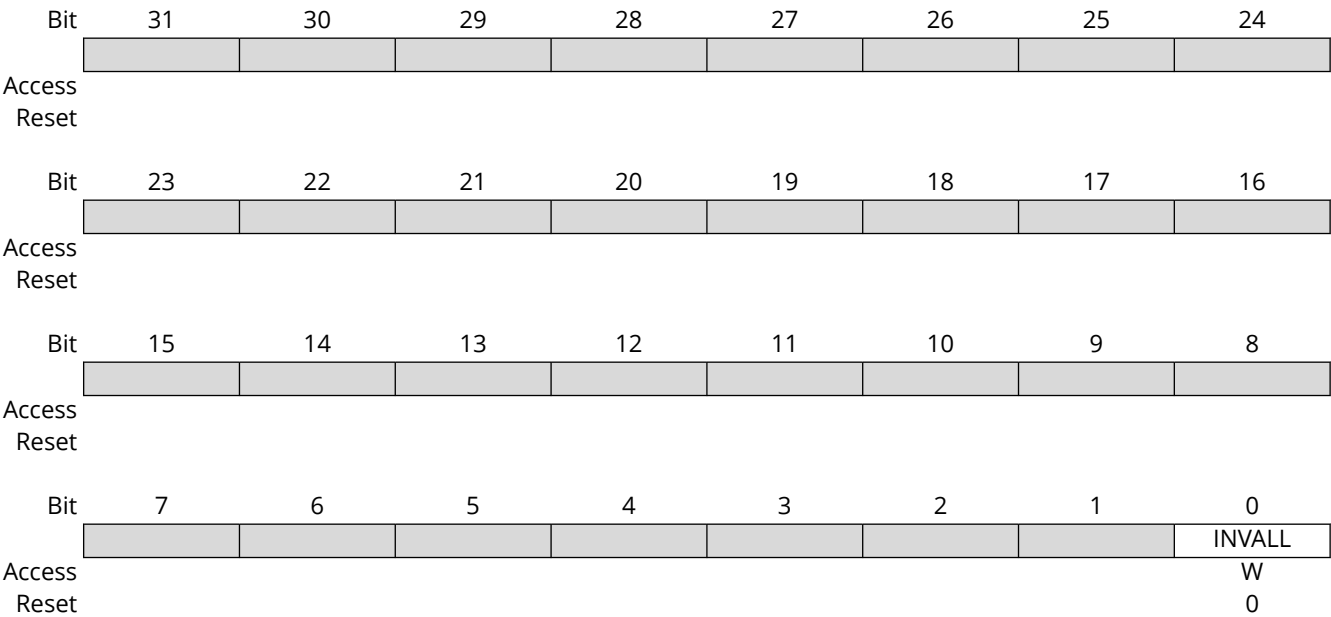
Name: LCKWAY
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					LCKWAY[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – LCKWAY[3:0] Lockdown Way Register
This field selects which way is locked.

11.10.6 Cache Maintenance 0

Name: MAINT0
Offset: 0x20
Reset: 0x00000000
Property: Write-only



Bit 0 – INVALL Cache Controller Invalidate All
Writing a '0' to this bit has no effect.
Writing a '1' to this bit invalidates all cache entries.

11.10.7 Cache Maintenance 1

Name: MAINT1
Offset: 0x24
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	WAY[3:0]							
Access	W	W	W	W				
Reset	0	0	0	0				
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					INDEX[7:4]			
Access					W	W	W	W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	INDEX[3:0]							
Access	W	W	W	W				
Reset	0	0	0	0				

Bits 31:28 – WAY[3:0] Invalidate Way

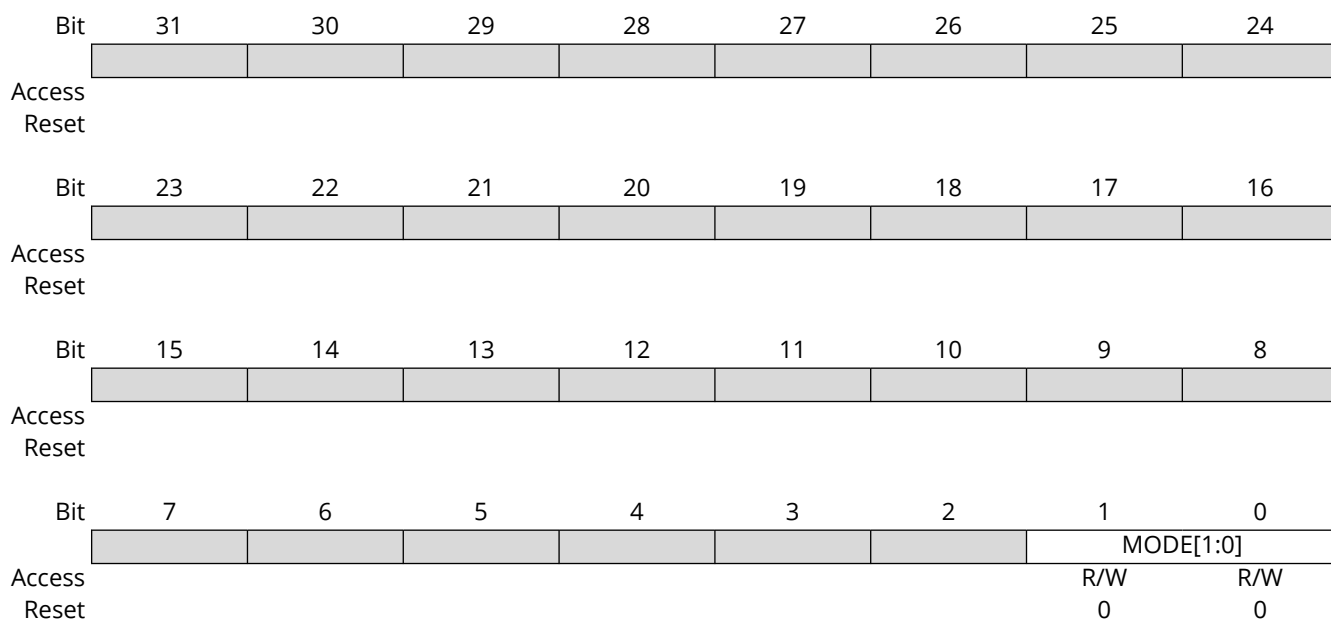
Value	Name	Description
0x0	WAY0	Way 0 is selection for index invalidation
0x1	WAY1	Way 1 is selection for index invalidation
0x2	WAY2	Way 2 is selection for index invalidation
0x3	WAY3	Way 3 is selection for index invalidation
0x4–0xF		Reserved

Bits 11:4 – INDEX[7:0] Invalidate Index

This field selects the index value for invalidation

11.10.8 Cache Monitor Configuration

Name: MCFG
Offset: 0x28
Reset: 0x00000000
Property: Read/Write



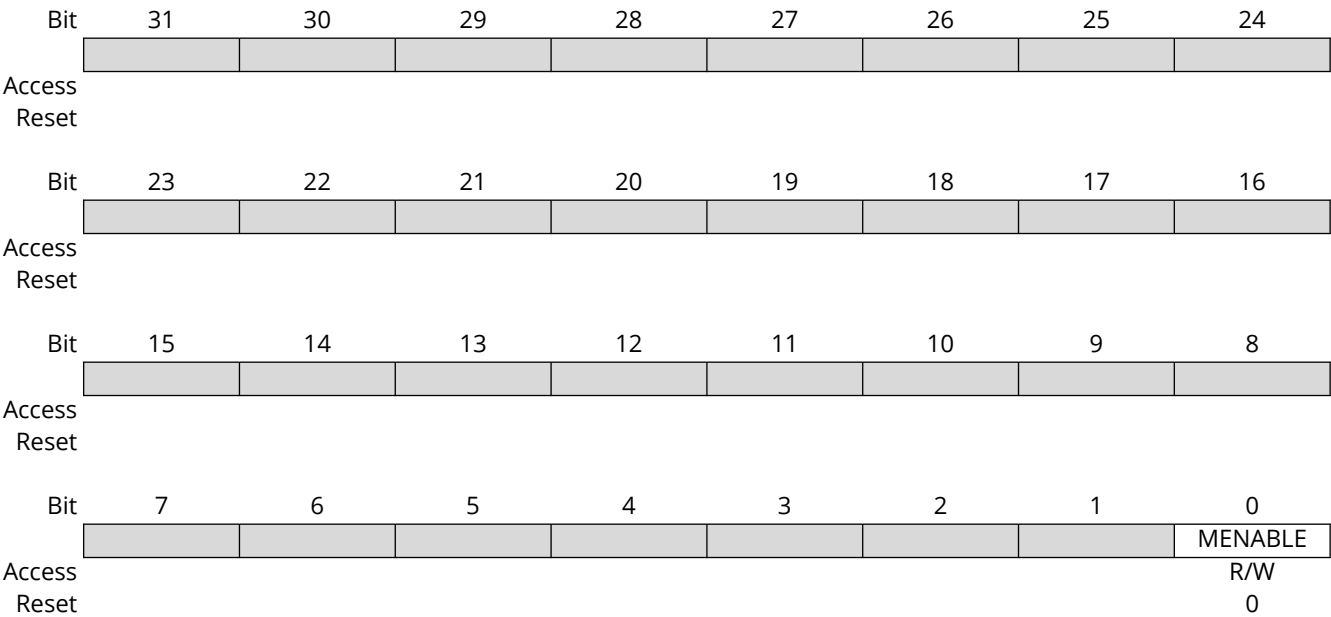
Bits 1:0 – MODE[1:0] Cache Controller Monitor Counter Mode

This field selects the type of data monitored.

Value	Name	Description
0x0	CYCLE_COUNT	Cycle counter
0x1	IHIT_COUNT	Instruction hit counter
0x2	DHIT_COUNT	Data hit counter
0x3		Reserved

11.10.9 Cache Monitor Enable

Name: MEN
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write



Bit 0 – MENABLE Cache Controller Monitor Enable
Writing a ‘0’ to this bit disables the monitor counter.
Writing a ‘1’ to this bit enables the monitor counter.

11.10.10 Cache Monitor Control

Name: MCTRL
Offset: 0x30
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								0

Bit 0 – SWRST Cache Controller Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the event counter register.

11.10.11 Cache Monitor Status

Name: MSR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EVENT_CNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EVENT_CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EVENT_CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EVENT_CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – EVENT_CNT[31:0] Monitor Event Counter
This field indicates the Monitor Event Counter value.

12. Device Service Unit (DSU)

12.1 Overview

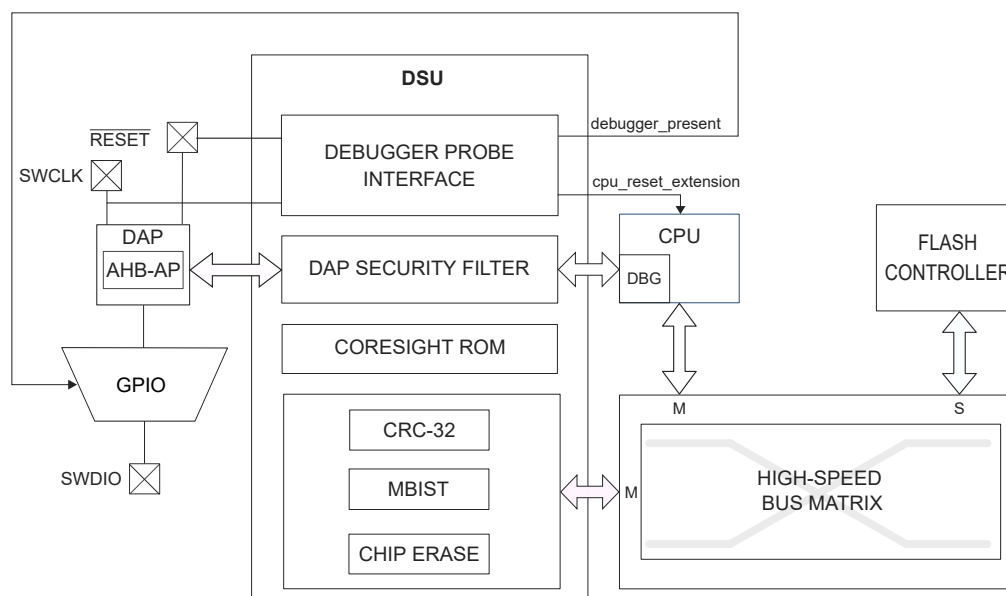
The Device Service Unit (DSU) provides a means of detecting debugger probes. It enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU Reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. For security reasons, some of the DSU features will be limited or unavailable when the device is protected by the Code Protect bit.

12.2 Features

- CPU Reset Extension
- Debugger Probe Detection (Cold- and Hot-Plugging)
- Chip-Erase Command and Status
- 32-Bit Cyclic Redundancy Check (CRC32) of any Memory Accessible Through the Bus Matrix
- ARM® CoreSight™ Compliant Device Identification
- Two Debug Communications Channels
- Debug Access Port Security Filter

12.3 Block Diagram

Figure 12-1. DSU Block Diagram



12.4 Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
RESET	Digital Input	External Reset pin

.....continued

Signal Name	Type	Description
SWCLK	Digital Input	SW clock pin
SWDIO	Digital I/O	SW bidirectional data pin

12.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

12.5.1 I/O Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU Reset phase (see *Debugger Probe Detection* from Related Links). The Hot-Plugging feature depends on the GPIO configuration. If the SWCLK pin function is changed in the port, the Hot-Plugging feature is not disabled. Hot-Plugging is disabled with the CFGCON0.HPLUGDIS bit, which is enabled by default. Therefore to use the SWCLK pin for GPIO functions, it must be disabled by setting CFGCON0.HPLUGDIS = 1.

Related Links

[12.6.3. Debugger Probe Detection](#)

12.5.2 Power Management

The DSU will continue to operate in any sleep mode where the selected source clock is running.

12.5.3 DMA

The DMA request lines are connected to the DMA Controller (DMAC). To use DMA requests with this peripheral, the DMAC must be configured first (see *Direct Memory Access Controller* from Related Links).

Related Links

[22. Direct Memory Access Controller \(DMAC\)](#)

12.5.4 Interrupts

Not applicable.

12.5.5 Events

Not applicable.

12.5.6 Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)

Note: Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

12.5.7 Analog Connections

Not applicable.

12.6 Debug Operation

12.6.1 Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

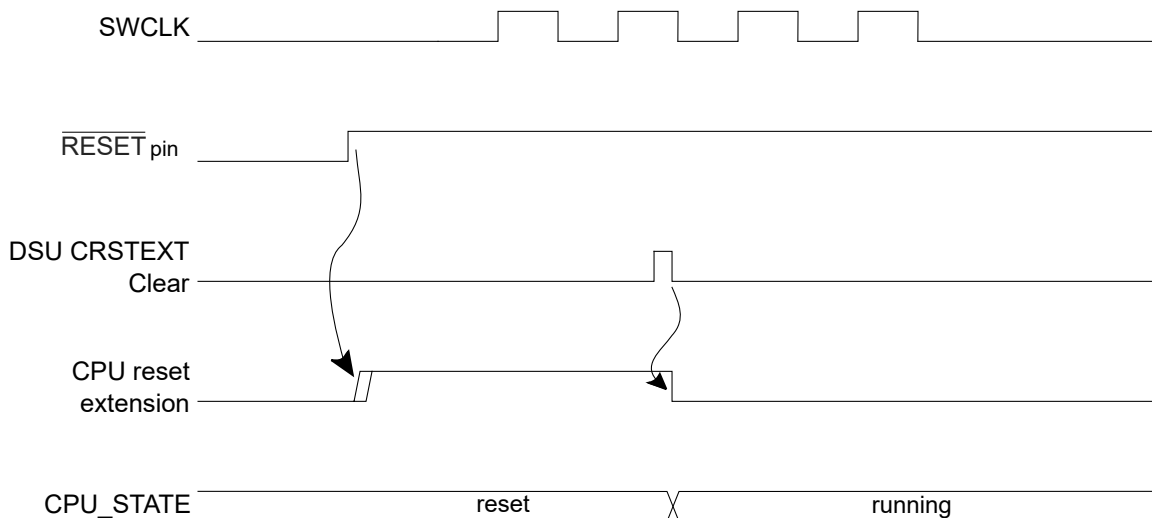
- CPU Reset extension
- Debugger probe detection

For more details on the ARM debug components, refer to the *ARM Debug Interface v5 Architecture Specification*.

12.6.2 CPU Reset Extension

"CPU Reset extension" refers to the extension of the Reset phase of the CPU core after the external Reset is released. This ensures that the CPU is not executing code at start-up while a debugger connects to the system. The debugger is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At start-up, SWCLK is internally pulled up to avoid false detection of a debugger if the SWCLK pin is left unconnected. When the CPU is held in the Reset extension phase, the CPU Reset extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will, then, be set to '0'. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU Reset extension when the device is protected by the Code Protect bit. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 12-2. Typical CPU Reset Extension Set and Clear Timing Diagram



12.6.3 Debugger Probe Detection

12.6.3.1 Cold Plugging

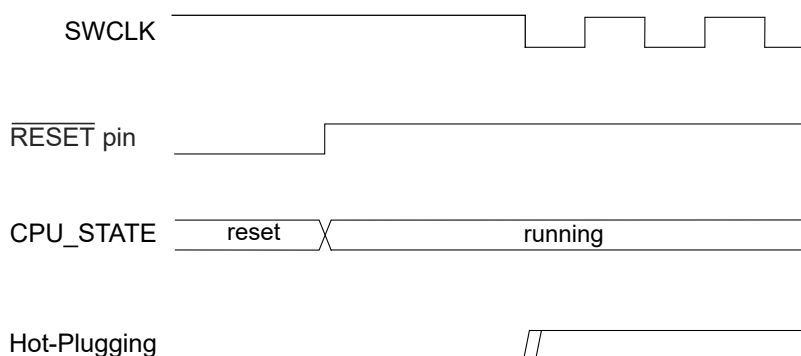
Cold-Plugging is the detection of a debugger when the system is in Reset. Cold-Plugging is detected when the CPU Reset extension is requested, as described above.

12.6.3.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in Reset. Hot-Plugging is not possible under Reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK pin function is changed in the port, the Hot-Plugging feature is not disabled. Hot-Plugging is disabled with the CFGCON0.HPLUGDIS bit, which is enabled by default. Therefore, to use the

SWCLK pin for GPIO functions, it must be Disabled by setting CFGCON0.HPLUGDIS=1. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 12-3. Hot-Plugging Detection Timing Diagram



The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the Code Protect bit.

This detection requires that pads are correctly powered. Thus, at cold start-up, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external Reset timing must be longer than the POR timing. If external Reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

12.7 Chip Erase

Chip erase consists of removing all sensitive information stored in the chip and clearing the Code Protect bit. Therefore, all volatile memories and the Flash memory (including the EEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must first reset the device in order to be detected. This ensures that internal registers are reset after the Protected state is removed. The chip erase operation is triggered by writing a '1' to the chip erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the chip erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The chip erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a chip erase after a Cold-Plugging procedure to ensure that the device is in a known and Safe state.

The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (see *Cold Plugging* from Related Links). The device then:
 - a. Detects the debugger probe.
 - b. Holds the CPU in Reset.
2. Issue the chip erase command by writing a '1' to CTRL.CE. The device then:
 - a. Clears the system volatile memories.
 - b. Erases the whole Flash array (excluding OTP).
 - c. Erases the Code Protect bit protection.
3. Check for completion by polling STATUSA.DONE (read as '1' when completed).

4. Reset the device to let the Flash Controller update the fuses.

Related Links

[12.6.3.1. Cold Plugging](#)

12.8 Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the Code Protect bit. The programming procedure is as follows:

1. At power-up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold (see *Power-on Reset (POR)* from Related Links for the characteristics). The system continues to be held in this Static state until the internally regulated supplies have reached a safe Operating state.
2. The PM starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal Resets are maintained due to the external Reset.
3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU remains in Reset due to the Cold-Plugging procedure; meanwhile, the rest of the system is released.
6. A chip erase is issued to ensure that the Flash is fully erased prior to programming.
7. Programming is available through the AHB-AP. Refer to the *PIC32CX-BZ2 Programming Specification* for more details.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$ or toggling power. Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent extending the CPU Reset.

Related Links

[13.15.3.2. Power-on Reset \(POR\)](#)

12.9 Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and this is accomplished by setting the Code Protect bit. This Protected state can be removed by issuing a chip erase (see *Chip Erase* from Related Links). When the device is protected, read/write accesses using the AHB-AP are limited to the DSU address range and DSU commands are restricted. When issuing a chip erase, sensitive information is erased from volatile memory and Flash.

The DSU implements a security filter that monitors the AHB transactions inside the DAP. If the device is protected, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (For more details, refer to the *ARM Debug Interface v5 Architecture Specification* on www.arm.com).

The DSU is intended to be accessed either:

- Internally from the CPU, without any limitation, even when the device is protected
- Externally from a debug adapter, with some restrictions when the device is protected

For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map has been mirrored at offset 0x100:

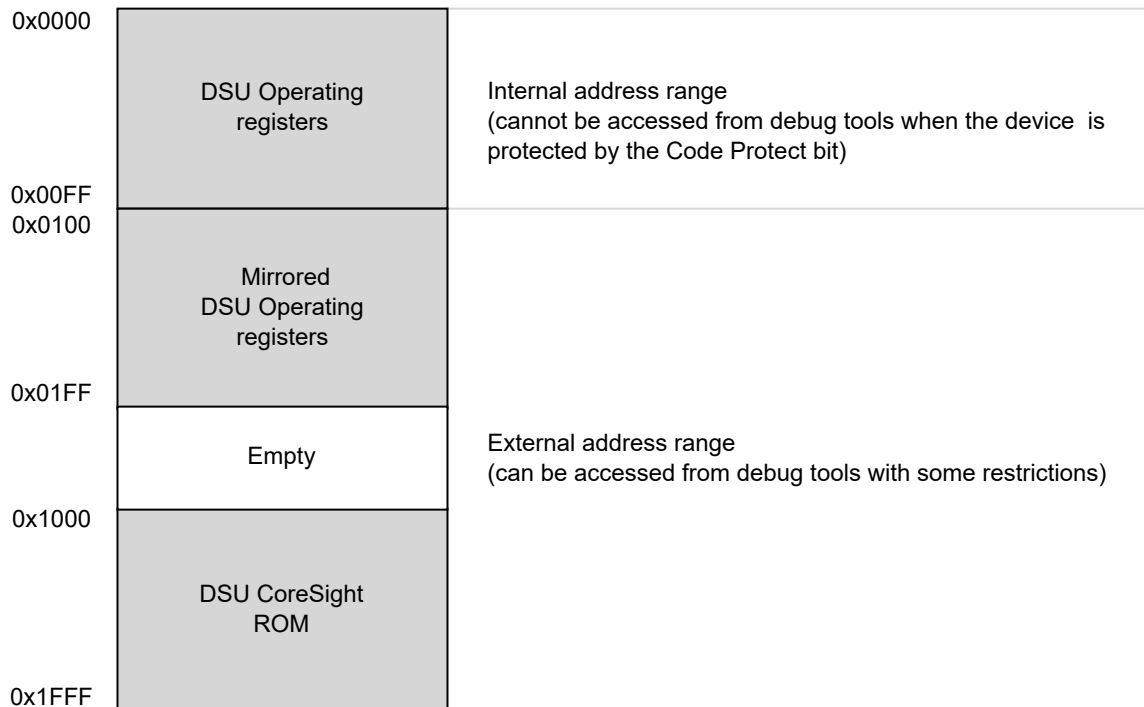
- The first 0x100 bytes form the internal address range

- The next 0x100 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU range 0x0100-0x2000.

The DSU Operating registers are located in the 0x0000-0x00FF area and remapped in 0x0100-0x01FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x0100-0x01FF, it is subject to security restrictions. For more information, refer to the following table.

Figure 12-4. APB Memory Mapping



Some features not activated by APB transactions are not available when the device is protected:

Table 12-1. Feature Availability Under Protection

Features	Availability When the Device is Protected
CPU Reset Extension	Yes
Clear CPU Reset Extension	No
Debugger Cold-Plugging	Yes
Debugger Hot-Plugging	No

Related Links

[12.7. Chip Erase](#)

12.10 Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

12.10.1 CoreSight Identification

A system-level ARM® CoreSight™ ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM

Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 12-5. Conceptual 64-bit Peripheral ID

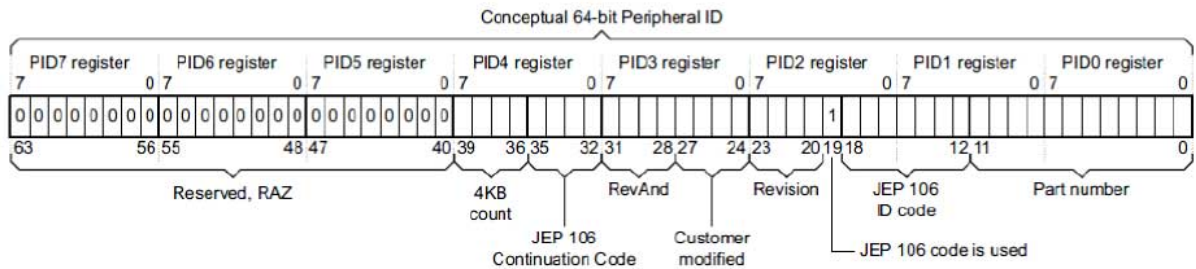


Table 12-2. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Microchip continuation code: 0x0	PID4
JEP-106 ID code	7	Microchip device ID: 0x1F	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
RevAnd	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID)	PID2

For more details, refer to the *ARM Debug Interface Version 5 Architecture Specification*.

12.10.2 Chip Identification Method

The DSU DID register identifies the device as shown in the following table:

Table 12-3. DSU DID Encoding

Field	Size	Value	Comments
Revision	4 bits	0x0	Immutable Field (0x0=Rev-A0)
Family	5 bits	0b00000	Family[4:0]
Series	6 bits	0b00000	Series[5:0]
Die	8 bits	0x9B	Immutable Field = Mask ID [7:0]
DEVSEL	8 bits	Flash Fuses	Determines variants of product {VSEL[7:0]} [0x8F 0x0F 0x0B]

12.11 Functional Description

12.11.1 Principle of Operation

The DSU provides memory services, such as CRC32 that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

12.11.2 Basic Operation

12.11.2.1 Initialization

The module is enabled by enabling its clocks, see *Clock and Reset Unit (CRU)* from Related Links. The DSU registers can be PAC write-protected, see *Peripheral Access Controller (PAC)* from Related Links.

Related Links

[13. Clock and Reset Unit \(CRU\)](#)

[26. Peripheral Access Controller \(PAC\)](#)

12.11.2.2 Operation From a Debug Adapter

Debug adapters must access the DSU registers in the external address range 0x100 – 0x2000. If the device is protected by the Code Protect bit, accessing the first 0x100 bytes causes the system to return an error. See *Intellectual Property Protection* from Related Links.

Related Links

[12.9. Intellectual Property Protection](#)

12.11.2.3 Operation From the CPU

There are no restrictions when accessing DSU registers from the CPU. However, the user must access DSU registers in the internal address range (0x0–0x100) to avoid external security restrictions. See *Intellectual Property Protection* from Related Links.

Related Links

[12.9. Intellectual Property Protection](#)

12.11.3 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is restricted; DATA, ADDR, and LENGTH values are forced (see below)

Table 12-4. AMOD Bit Descriptions when Operating CRC32

AMOD[1:0]	Short name	External range restrictions
0	ARRAY	CRC32 is restricted to the full Flash array area (EEPROM Emulation area not included) DATA forced to 0xFFFFFFFF before calculation (no seed)
1	EEPROM	CRC32 of the whole EEPROM Emulation area DATA forced to 0xFFFFFFFF before calculation (no seed)
2-3	Reserved	-

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

12.11.3.1 Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept noninverted if used as starting point for subsequent CRC32 calculations.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

12.11.3.2 Interpreting the Results

The user must monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

12.11.4 Debug Communication Channels

The Debug Communication Channels (DCC0 and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the Code Protect bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in Debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the Protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

12.11.5 System Services Availability when Accessed Externally and Device is Protected

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x000-0x100 range.

Table 12-5. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range and Device is Protected
Chip erase command and status	Yes
CRC32	Yes, only full array or full EEPROM
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
STATUSA.CRSTEXT clearing	No (STATUSA.PERR is set when attempting to do so)

12.12 DSU Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0				
0x00	CTRL	7:0				CE				SWRST				
0x01	STATUSA	7:0				PERR	FAIL	BERR	CRSTEXT	DONE				
0x02	STATUSB	7:0			CELCK	HPE	DCCD1	DCCD0	DBGPRES	PROT				
0x03	Reserved													
0x04	ADDR	7:0	ADDR[5:0]						AMOD[1:0]					
		15:8	ADDR[13:6]											
		23:16	ADDR[21:14]											
		31:24	ADDR[29:22]											
0x08	LENGTH	7:0	LENGTH[5:0]											
		15:8	LENGTH[13:6]											
		23:16	LENGTH[21:14]											
		31:24	LENGTH[29:22]											
0x0C	DATA	7:0	DATA[7:0]											
		15:8	DATA[15:8]											
		23:16	DATA[23:16]											
		31:24	DATA[31:24]											
0x10	DCC0	7:0	DATA[7:0]											
		15:8	DATA[15:8]											
		23:16	DATA[23:16]											
		31:24	DATA[31:24]											
0x14	DCC1	7:0	DATA[7:0]											
		15:8	DATA[15:8]											
		23:16	DATA[23:16]											
		31:24	DATA[31:24]											
0x18	DID	7:0	DEVSEL[7:0]											
		15:8	DIE[7:0]											
		23:16	FAMILY[0]		SERIES[5:0]									
		31:24	REVISION[3:0]				FAMILY[4:1]							
0x1C ... 0x0FFF	Reserved													
0x1000	ENTRY0	7:0							FMT	EPRES				
		15:8	ADDOFF[3:0]											
		23:16	ADDOFF[11:4]											
		31:24	ADDOFF[19:12]											
0x1004	ENTRY1	7:0						FMT	EPRES					
		15:8	ADDOFF[3:0]											
		23:16	ADDOFF[11:4]											
		31:24	ADDOFF[19:12]											
0x1008	END	7:0	END[7:0]											
		15:8	END[15:8]											
		23:16	END[23:16]											
		31:24	END[31:24]											
0x100C ... 0x1FCB	Reserved													
0x1FCC	MEMTYPE	7:0								SMEMP				
		15:8												
		23:16												
		31:24												
0x1FD0	PID4	7:0	FKBC[3:0]				JEPCC[3:0]							
		15:8												
		23:16												
		31:24												
0x1FD4 ... 0x1FDF	Reserved													

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1FE0	PID0	7:0	PARTNBL[7:0]							
		15:8								
		23:16								
		31:24								
0x1FE4	PID1	7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
		15:8								
		23:16								
		31:24								
0x1FE8	PID2	7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
		15:8								
		23:16								
		31:24								
0x1FEC	PID3	7:0	REVAND[3:0]				CUSMOD[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF0	CID0	7:0	PREAMBLEB0[7:0]							
		15:8								
		23:16								
		31:24								
0x1FF4	CID1	7:0	CCLASS[3:0]				PREAMBLE[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF8	CID2	7:0	PREAMBLEB2[7:0]							
		15:8								
		23:16								
		31:24								
0x1FFC	CID3	7:0	PREAMBLEB3[7:0]							
		15:8								
		23:16								
		31:24								

12.13 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. See *Register Access Protection* from Related Links.

Related Links

[12.5.6. Register Access Protection](#)

12.13.1 Control

Name: CTRL
Offset: 0x0000
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CE				SWRST
Access				W				W
Reset				0				0

Bit 4 – CE Chip-Erase
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit starts the Chip-Erase operation.

Bit 0 – SWRST Software Reset
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit resets the module.

12.13.2 Status A

Name: STATUSA
Offset: 0x0001
Reset: 0x00
Property: PAC Write Protection

Bit	7	6	5	4	3	2	1	0
				PERR	FAIL	BERR	CRSTEXT	DONE
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – PERR Protection Error

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Protection Error bit.
This bit is set when a command that is not allowed in Protected state is issued.

Bit 3 – FAIL Failure

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Failure bit.
This bit is set when a DSU operation failure is detected.

Bit 2 – BERR Bus Error

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Bus Error bit.
This bit is set when a bus error is detected.

Bit 1 – CRSTEXT CPU Reset Phase Extension

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CPU Reset Phase Extension bit.
This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU Reset phase.

Bit 0 – DONE Done

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Done bit.
This bit is set when a DSU operation is completed.

12.13.3 Status B

Name: STATUSB
Offset: 0x0002
Reset: 0x0x
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			CELCK	HPE	DCCD1	DCCD0	DBGPRES	PROT
Access			R	R	R	R	R	R
Reset			0	0	0	0	x	x

Bit 5 – CELCK Chip Erase Locked

Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when Chip Erase is locked.
This bit is cleared when Chip Erase is unlocked.

Bit 4 – HPE Hot-Plugging Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when Hot-Plugging is enabled.
This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

Bits 2, 3 – DCCD Debug Communication Channel x Dirty

Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when DCC is written.
This bit is cleared when DCC is read.

Bit 1 – DBGPRES Debugger Present

Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when a debugger probe is detected.
This bit is never cleared.

Bit 0 – PROT Protected

Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set at power-up when the device is protected.
This bit is never cleared.

12.13.4 Address

Name: ADDR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]						AMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:2 – ADDR[29:0] Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0] Access Mode
The functionality of these bits is dependent on the operation mode.
Bit description when operating CRC32 (see *32-bit Cyclic Redundancy Check (CRC32)* from Related Links).

Related Links

[12.11.3. 32-bit Cyclic Redundancy Check CRC32](#)

12.13.5 Length

Name: LENGTH
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0] Length
Length in words needed for memory operations.

12.13.6 Data

Name: DATA
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Memory operation initial value or result value.

12.13.7 Debug Communication Channel x

Name: DCC
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

12.13.8 Device Identification

Name: DID
Offset: 0x0018
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	REVISION[3:0]				FAMILY[4:1]			
Access	R	R	R	R	R	R	R	R
Reset	p	p	p	p	f	f	f	f
Bit	23	22	21	20	19	18	17	16
	FAMILY[0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	f		s	s	s	s	s	s
Bit	15	14	13	12	11	10	9	8
	DIE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	d	d	d	d	d	d	d	d
Bit	7	6	5	4	3	2	1	0
	DEVSEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – REVISION[3:0] Processor

The value of this field identifies the die revision number. 0x0=rev.A0.

Bits 27:23 – FAMILY[4:0] Product Family

The value of this field corresponds to the product family part of the ordering code. For this device, the value of this field is 0x0.

Bits 21:16 – SERIES[5:0] Product Series

The value of this field corresponds to the product series part of the ordering code. For this device, the value of this field is 0x01, corresponding to a product with the Cortex-M0+ processor with DMA and USB features.

Bits 15:8 – DIE[7:0] Die Number

Identifies the die family.

Bits 7:0 – DEVSEL[7:0] Device Selection

This bit field identifies a device within a product family and product series. The value corresponds to the Flash memory density, pin count and device variant parts of the ordering code.

12.13.9 CoreSight ROM Table Entry x

Name: ENTRY
Offset: $0x1000 + n \cdot 0x04$ [$n=0..1$]
Reset: 0xxxxxx00x
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset

The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format

Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present

This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

12.13.10 CoreSight ROM Table End

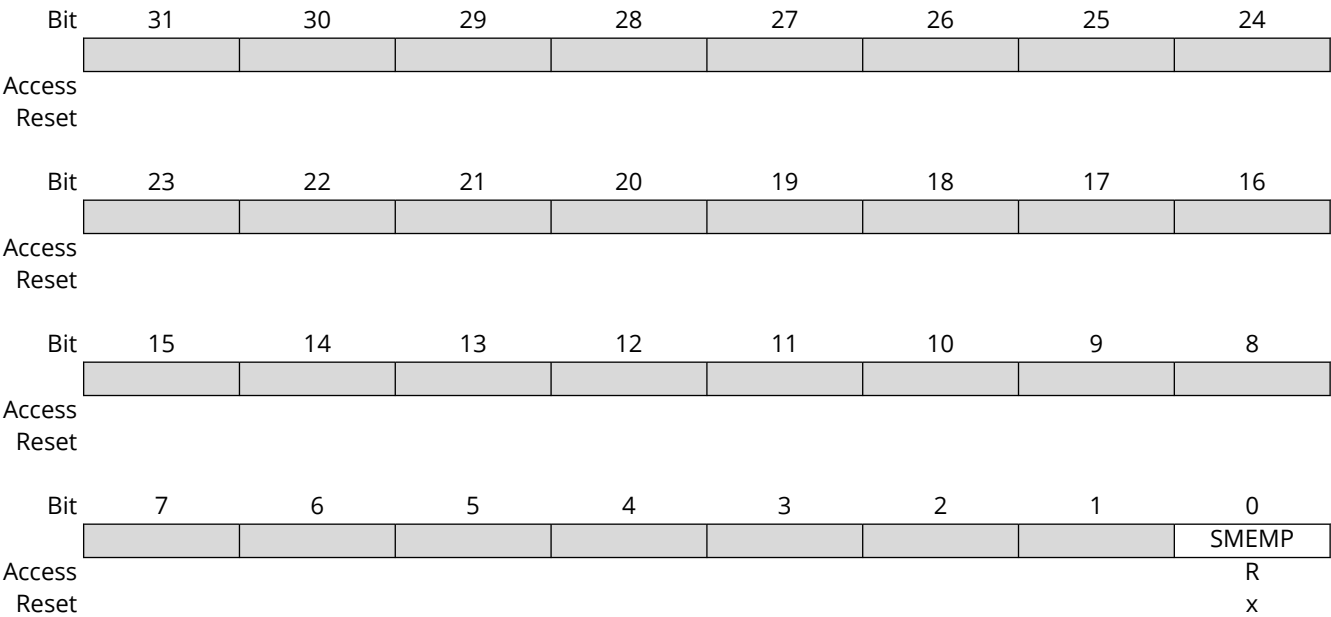
Name: END
Offset: 0x1008
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0] End Marker
Indicates the end of the CoreSight ROM table entries.

12.13.11 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: 0x0000000X
Property: -



Bit 0 – SMEMP System Memory Present
This bit indicates whether system memory is present on the bus that connects to the ROM table.
This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

12.13.12 Peripheral Identification 4

Name: PID4
Offset: 0x1FD0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0] 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0] JEP-106 Continuation Code
These bits will always return zero when read.

12.13.13 Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x000000D0
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PARTNBL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PARTNBL[7:0] Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

12.13.14 Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x000000FC
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0] Low Part of the JEP-106 Identity Code
These bits will always return 0xF when read (JEP-106 identity code is 0x1F).

Bits 3:0 – PARTNBH[3:0] Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

12.13.15 Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x00000009
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bits 7:4 – REVISION[3:0] Revision Number

Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU JEP-106 Identity Code is Used

This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0] JEP-106 Identity Code High

These bits will always return 0x1 when read, (JEP-106 identity code is 0x1F).

12.13.16 Peripheral Identification 3

Name: PID3
Offset: 0x1FEC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CUSMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0] Revision Number
These bits will always return 0x0 when read.

Bits 3:0 – CUSMOD[3:0] ARM CUSMOD
These bits will always return 0x0 when read.

12.13.17 Component Identification 0

Name: CID0
Offset: 0x1FF0
Reset: 0x0000000D
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PREAMBLEB0[7:0] Preamble Byte 0
These bits will always return 0x0000000D when read.

12.13.18 Component Identification 1

Name: CID1
Offset: 0x1FF4
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PREAMBLE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0] Component Class

These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (For more details, refer to the *ARM Debug Interface v5 Architecture Specification* at <http://www.arm.com>).

Bits 3:0 – PREAMBLE[3:0] Preamble

These bits will always return 0x0 when read.

12.13.19 Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x00000005
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PREAMBLEB2[7:0] Preamble Byte 2
These bits will always return 0x00000005 when read.

12.13.20 Component Identification 3

Name: CID3
Offset: 0x1FFC
Reset: 0x000000B1
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0] Preamble Byte 3
These bits will always return 0x000000B1 when read.

13. Clock and Reset Unit (CRU)

13.1 Overview

The PIC32CX-BZ2 Clock System provides both clocking and reset functions. This chapter describes the clocking functionality and summarizes the clock distribution and terminology in the PIC32CX-BZ2 device. For more details on configuration, see the respective peripherals' descriptions. Clock control is handled by the CRU to provide system clocks and interface peripheral clocks. The CRU controls switching and synchronization of clock sources.

For details on the Reset functionality, see *Resets* from Related Links.

Related Links

[13.15. Resets](#)

13.2 Features

The Clock and Reset Unit has the following features:

- Supports the following as system clock sources:
 - 16 MHz Primary Crystal Oscillator (POSC)
 - 8 MHz Fast RC Oscillator (FRC)
 - 32 kHz Low Power RC Oscillator (LPRC)
 - 32.768 kHz Secondary Crystal Oscillator (SOSC)
 - 96 MHz System PLL (RFPLL)
- Provides control registers for all PLLs
- Provides for glitch-free clock switching between various clock sources
- Post dividers on processor clock generator to slow down system clock for power save
- A fail safe clock monitor that detects clock failure and provides automatic switching to the FRC
- Provides control registers for user interface of clocks and resets
- Provides configuration bits for oscillator selection and calibration of on-chip oscillators
- Provides control registers to generate a reference clock output
- Provide resets for the system
- Provides NMI interrupts for the system
- Multiple PB clock dividers
- One system clock, SYS_CLK, from which almost all clocks used throughout the system are derived
- Three peripheral clocks, created by independent integer dividers of the SYS_CLK:
 - PB1_CLK: PB-PIC and PB-Bridge-A bus
 - PB2_CLK: PB-Bridge-B and PB-Bridge-C
 - PB3_CLK: DS/XDS bus clock
- Six reference output clocks (REFO1 – REFO6) with the following clock sources:
 - System clock (SYS_CLK)
 - PB1 bus clock (PB1_CLK)
 - 16 MHz Primary Crystal Oscillator (POSC)
 - 8 MHz Fast RC Oscillator (FRC)
 - 32 kHz Low Power RC Oscillator (LPRC)
 - 32.768 kHz Secondary Crystal Oscillator (SOSC)

- 96 MHz System PLL (RFPLL)
 - 64 MHz System PLL (RFPLL PGM MHz)
 - REFI pin
 - Sleep control, supporting Req/Ack signaling with the bus matrix to determine that no transactions are in flight when initiating sleep
- JTAG TCK clock control

13.3 Block Diagram

The CRU, along with the PMD, provides gated clock output for all peripheral buses. The following figure illustrates the CRU block diagram.

Figure 13-1. Clock and Reset Unit Block Diagram

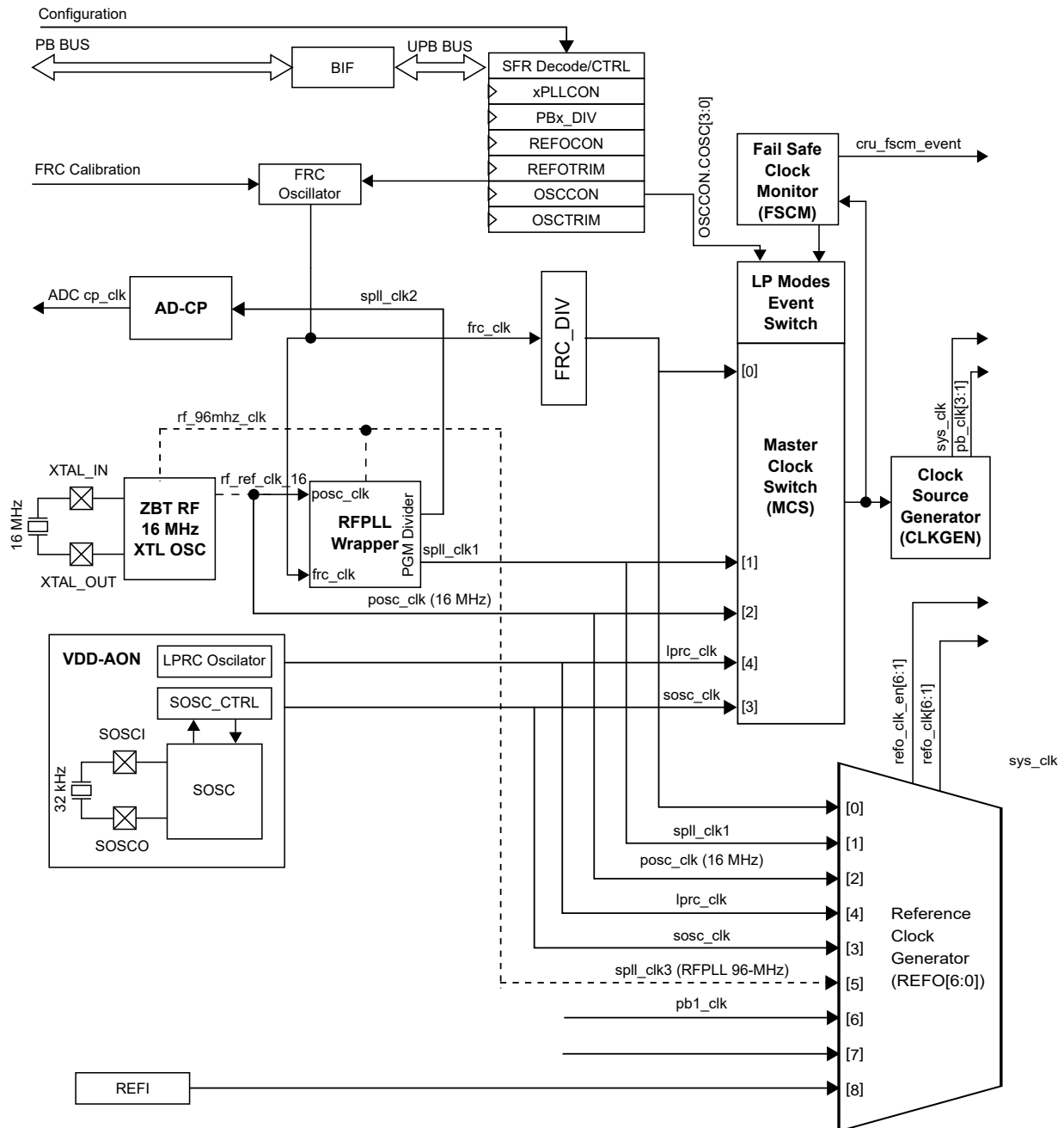


Figure 13-2. RFPLL Wrapper

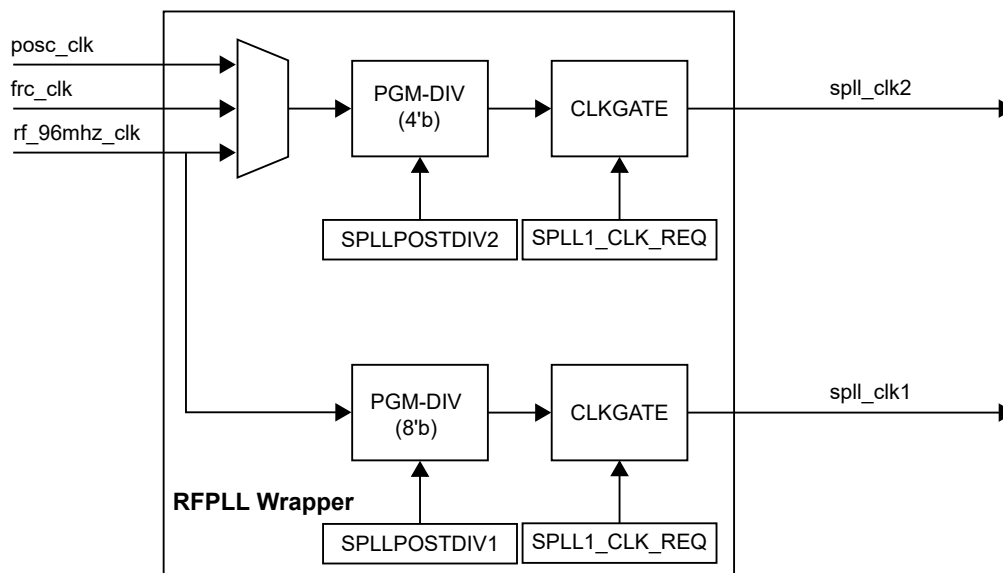
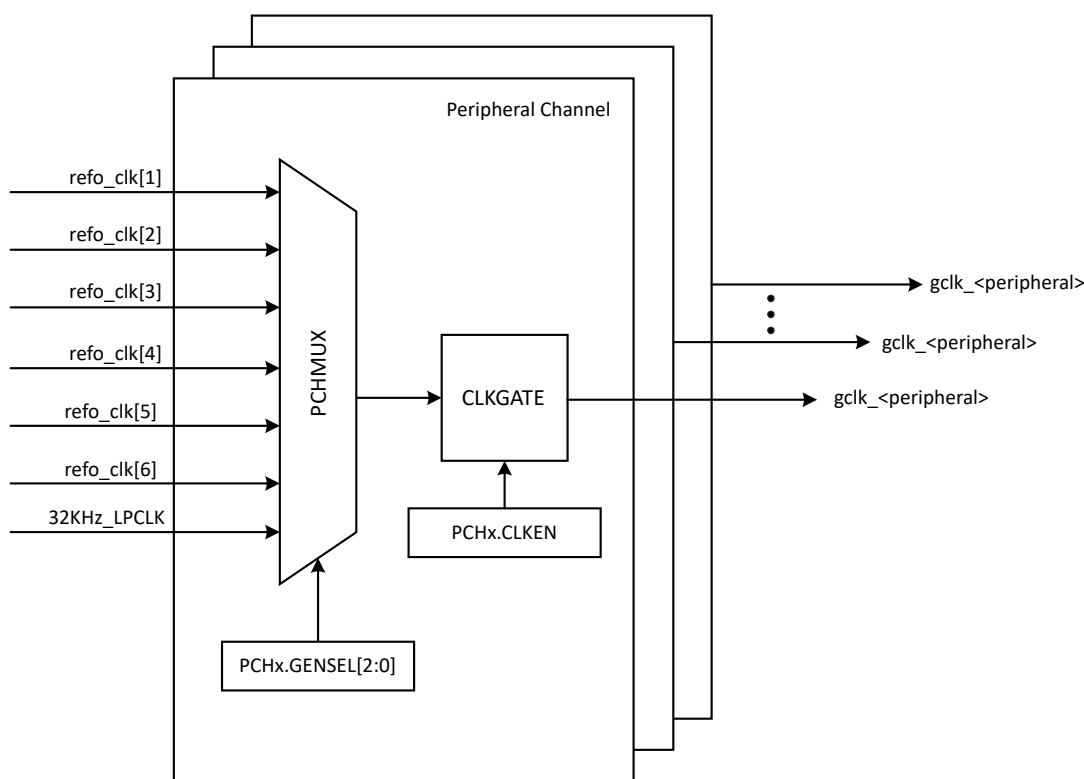


Figure 13-3. Peripheral Clock Generation



The CRU Master Clock Switch selects the input clock, which needs to be fed to the CLKGEN Synchronous Clock Generator. The CLKGEN generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB and AHB) and the synchronous (to the CPU) user interfaces of the peripherals. It contains prescalers for the CPU and bus clocks.

13.4 System and Peripheral Clock Generation (CLKGEN)

This sub-module generates the system clocks needed for the device from a single source clock. In addition, this module also shuts down these clocks during the Sleep mode.

There are two types of clocks generated by this block called core clocks and peripheral clocks. The system clock (SYS_CLK) is typically used by the CPU. It supports components, such as memory subsystems, and fast peripherals. The peripheral bus clocks (pb_clk) are used to clock slow peripheral devices attached to the pb_bus. The pb_clk[n] outputs are based on the SYS_CLK frequency with a fixed divisor. The divisor is determined by the value of the PBxDIV registers.

The system and peripheral clocks are stopped when in the Sleep mode. The clocks are restarted by disabling the sleep enable.

13.4.1 Sleep Mode

The Sleep mode is entered when DSCON[DSEN] is clear and the OSCCON[SLPEN] bit is set and the CPU executes a WFI instruction. This causes the device clocks to be held low ('0').

13.4.2 Sleep Mode Entry

Entry into the Sleep mode from any other mode does not require a clock switch. This is due to the fact that once in the Sleep mode, no clocks are needed.

Note: If software writes to the CRU SFRs before going into the Sleep mode, it is recommended a read be done of the SFRs to Flash the write before executing the WAIT instruction that initiates the Sleep mode.

13.4.3 Sleep Mode Exit

Unless the Two-Speed Start-up is enabled, there are no clock-switching events when exiting the Sleep mode. With Two-Speed Start-up disabled, the Sleep mode is effectively extended until the selected clock is ready. Once the clock is ready, the device enters the RUN mode.

If Two-Speed Start-up is enabled, the device will come out of Sleep running with the FRC as the clock source and perform an automatic clock switch to the selected clock source.

If Two-Speed Start-up is enabled, the RUN mode is entered immediately using the Two-Speed Start-up clock source. A clock switch to the selected clock source occurs once the source is ready.

Note: Two-Speed Start-up is not permanently enabled and is software programmable.

13.4.4 Sleep Mode with Delayed Exit

In some of the low power Sleep modes, some of the on-chip voltage regulators may be turned off in the system. If this is the case, the Sleep mode cannot be exited immediately on a wake-up event. The wake from Sleep will be delayed until the system is ready to have clocks turned on again.

13.5 Idle Mode

The Idle mode is entered when the OSCCON[SLPEN] bit is low and the CPU executes a WFI instruction. Only the CPU's internal clock is stopped in this mode.

Note: When exiting from the Sleep mode, the CRU will transition the system to the Idle mode before transitioning to the Run mode. This transition to the Idle mode is used to support the Dream mode and always occurs regardless of the CRU transitioning to the Dream mode or Run mode. Therefore, when exiting the Sleep mode, both the RCON bits [SLEEP] and [IDLE] will be set.

13.6 Dream Mode

When the DRMEN bit in the OSCCON register is set, it allows the DMA controller to switch between the Idle mode and the Sleep mode. When the OSCCON.SLPEN bit is set and the OSCCON.DRMEN bit is set, the CRU monitors the DMAC to make sure all transfers are complete before going into the Sleep mode.

If `OSCCON.SLPEN = 1`, `OSCCON.DRMEN = 1`, and peripheral clock requests are active, the CRU goes into the Idle mode until all peripheral clock requests are non-active; at which time the CRU goes into the Sleep mode.

If `OSCCON.SLPEN` is not set, the `DRMEN` bit has no affect as the DMA clocks are still running in the Idle mode.

If the CRU recognizes a wake/interrupt event whose priority will wake the DMA but not the CPU, the CRU transitions to the Idle mode. Therefore, the DMA can perform the needed operations and, when the DMA is finished, the CRU will go back to the Sleep mode. During this time, the CPU is still asleep.

If the wake event is such that the CPU must handle the event, the whole system will exit the Sleep mode and transition back to the Run mode.

13.7 FRCDIV

The FRC can be divided and used as a system clock. The user controls the divider setting using the `OSCCON.FRCDIV[2:0]` register bits. The divisor is configured for eight divider selections: /1, /2, /4, /8, /16, /32, /64, /256.

13.8 RFPLL Wrapper

The RFPLL wrapper generates two clocks:

- `spll_clk1` (PGM MHz)
 - Clock frequencies = 96 MHz/(1-255) and 64 MHz frequency choices and optional clock disable option
 - Clock ready indication
- `spll_clk2` (PGM MHz)
 - Clock frequencies = 96 MHz/(1-15) and optional clock disable option
 - Clock ready indication

Clocks are produced only when there is a request generated by the user; for `clk1`, it is CRU and for `clk2`, it is ADC charge-pump. Along with the clocks, individual clock ready is also generated, which indicates that clocks are ready for consumption.

13.9 Start-up Considerations

The presence of hardware NVR fuses on the PIC32CX-BZ2 device allows the system configuration fuses to be ready upon exiting Reset. The following start-up conditions exist:

- On any device Reset, no start-up time is required to transfer configuration values from the NVR memory into the configuration holding registers.
- Once the device is active, the user may change the primary system clock source from FRC to SPLL by using the `OSCCON` register.

13.10 Fail-Safe Clock Monitor

The Clock System includes a Fail-safe Clock Monitor (FSCM). The FSCM monitors the `SYS_CLK` for continuous operation. If it detects that the `SYS_CLK` failed, it switches the `SYS_CLK` over to the FRC oscillator and triggers an NMI. The FRC is an untuned 8 MHz oscillator that drives the `SYS_CLK` during an FSCM event. When the NMI is executed, software can restart the main oscillator or shut down the system.

In the Sleep mode, both the `SYS_CLK` and the FSCM halt, preventing FSCM detection.

13.11 Fast RC Oscillator

The on-chip 8 MHz Fast RC Oscillator (FRC) is intended to be a fast, with precise frequency, internal RC oscillator. The FRC supports calibration to $\pm 0.25\%$ accuracy pre-package. However, package-induced stress lowers the accuracy.

The FRC oscillator is accurate to provide the clock frequency necessary to maintain baud rate tolerance for serial data transmissions. FRC is enabled with conditions in [13.11.1. Enabling the FRC](#); otherwise, it is not enabled. Power-on Reset sets NOSC[3:0] = 0000; therefore, it is always ON when powered-up.

The oscillator module provides a 6-bit wide user tuning adjustment capability using the OSCTRM.TUN[5:0] bits.

13.11.1 Enabling the FRC

The FRC oscillator is powered when OSCCON.NOSC[3:0] = 4'b0000 or a Fail-safe clock monitor is enabled and a clock fail is detected, forcing a switch to FRC. It is also enabled whenever it gets requested by: ADC requesting for FRC, Configuring an SPLL2 source as FRC, PMU Controller requesting for FRC, Flash Controller requesting for FRC, Configuring LPCLK (32 KHz) source to FRC.

13.11.2 Frequency Tuning in User Mode

In addition to the factory calibration, the base frequency can be tuned in the user's application. This frequency tuning capability allows the user to deviate from the factory calibrated frequency. The user can tune the frequency by writing to the OSCTRM register. The tuning range of the FRC oscillator is $\pm 1.5\%$ of nominal in 3.75 kHz steps.

13.12 Secondary Oscillator

The Secondary Oscillator (SOSC) is a low-power 32.768 kHz crystal oscillator that provides accurate time keeping.

The Secondary Oscillator has the following features:

- 32.768 kHz operation
- Provides system clock output
- Provided to CRU or LPCLKGEN on request
- Can be disabled to reduce power
- Ultra-low power driver
- No calibration is required

13.13 Low Power RC Oscillator (LPRC)

The Low Power Internal RC Oscillator (LPRC) operates at a nominal frequency of 32.768 kHz.

Note: The LPRC is not a 50% duty cycle clock; however, it maintains an average frequency over a number of base clocks.

The LPRC can be used as both a source for the system clock and a reference for Backup core modules. These modules include the Deep Sleep Watchdog (DSWDT), clock monitor circuits and other modules that require a 32 kHz reference clock.

13.14 Reference Clock Generator

The Reference Clock Generator provides the Generic Clocks (GCLK_<Peripheral>) for system peripherals via Peripheral Channels. There are a total of 24 Peripheral Channels with the mapping as shown in following table.

Table 13-1. Peripheral Clock Generation

Peripheral Clock	Pchannel Index
GCLK_EIC, GCLK_CCL	0
GCLK_FREQM_MSR	1
GCLK_FREQM_REF	2
GCLK_SERCOM0_CORE, GCLK_SERCOM1_CORE	3

.....continued

Peripheral Clock	Pchannel Index
GCLK_SERCOM2_CORE, GCLK_SERCOM3_CORE	4
GCLK_TC0	5
GCLK_TC1	6
GCLK_TC2, GCLK_TC3	7
GCLK_EVSYS_CH_0	8
GCLK_EVSYS_CH_1	9
GCLK_EVSYS_CH_2	10
GCLK_EVSYS_CH_3	11
GCLK_EVSYS_CH_4	12
GCLK_EVSYS_CH_5	13
GCLK_EVSYS_CH_6	14
GCLK_EVSYS_CH_7	15
GCLK_EVSYS_CH_8	16
GCLK_EVSYS_CH_9	17
GCLK_EVSYS_CH_10	18
GCLK_EVSYS_CH_11	19
GCLK_TCC0	20
GCLK_TCC1, GCLK_TCC2	21
GCLK_AC	22
GCLK_CM4_TRACE	23

The mapping for the source of the clocks for both the CLKGEN generator and Reference clock generator are shown in following table.

Table 13-2. CRU Clock Mapping

clock_in[x]	MCS/COSC Mapping	REFO/ROSEL Mapping	FSCM Clock Source	Clock to Switch to on a FSCM Fail
0 - FRC	0000	0000	—	X
1 - SPL_L_CLK1	0001	0001	—	—
2 - POSC (16 MHz)	0010	0010	—	—
3- SOSC	0011	0011	—	—
4 - LPRC	0100	0100	X	—
5 - SPL_L_CLK3 (RFPLL, 96 MHz)	—	0101	—	—
6 - PB1_CLK	—	0110	—	—
7 - SYS_CLK	—	0111	—	—
8 - REFI Pin	—	1000	—	—

13.15 Resets

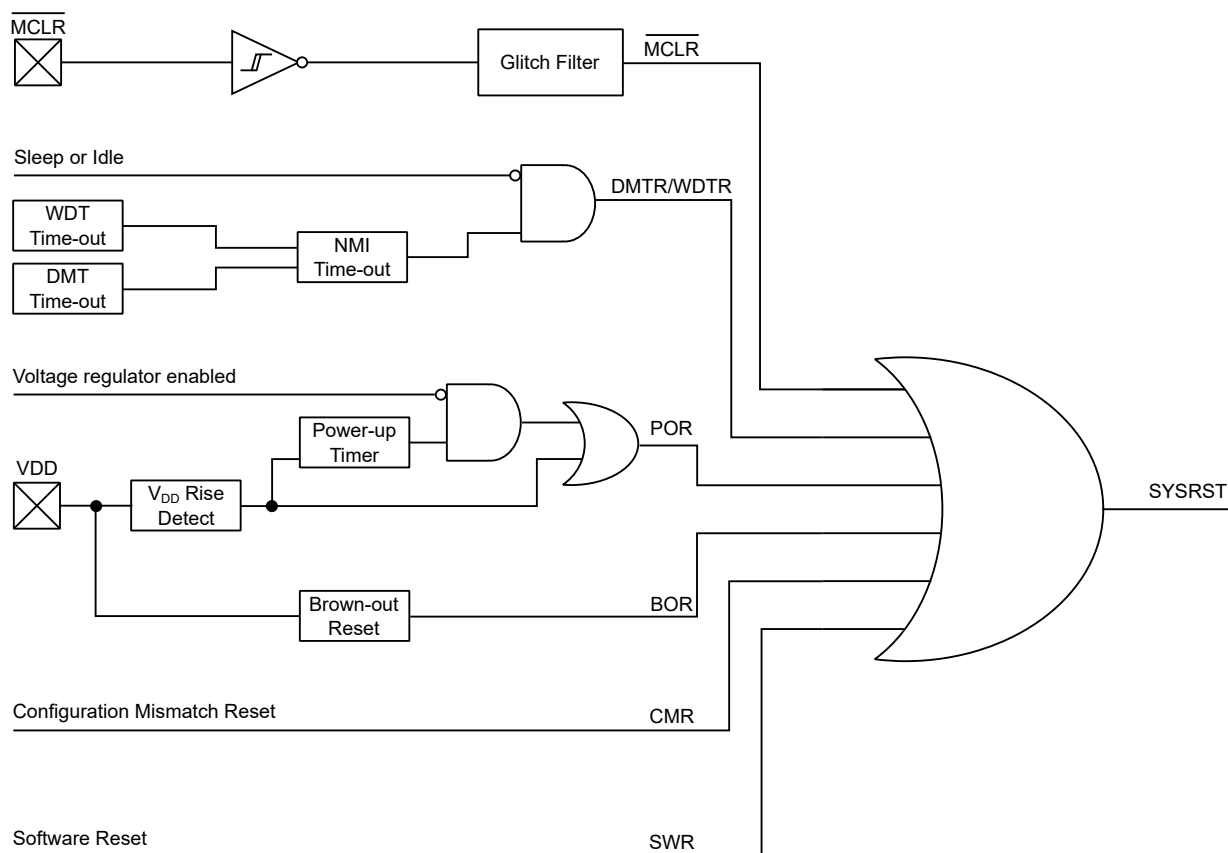
The Reset module combines all Reset sources and controls the device Master Reset signal, SYSRST. The device Reset sources are as follows:

- Power-on Reset (V_{dd} , IO, or POR)
- Brown-out Reset (BOR/ZPBOR)
- Master Clear Reset ($\overline{MCLR}$)
- Watchdog Timer Reset (NMI Counter)
- Dead Man Timer Reset (NMI Counter)
- Software Reset (SWR)

- Test mode Entry and Exit
- JTAG Reset
- Configuration Mismatch Reset (CMR)

A simplified block diagram of the Reset module is shown in the following figure. Any active source of reset will make the system Reset signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state.

Figure 13-4. System Reset Block Diagram



13.15.1 Control Registers

Most types of device resets will set corresponding Status bits in the RCON register to indicate the type of Reset (see *RCON* register from Related Links). The one exception is the Non-maskable Interrupt (NMI) time-out Reset. A Power-on Reset (POR) will clear all bits, except for the BOR and POR bits (*RCON*[1:0]), which are set. The user software may set or clear any of the bits at any time during code execution. The RCON bits serve only as Status bits. Setting a Reset status bit in software will not cause a system Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer (WDT) and device power-saving states. For more information on the function of these bits, see *Using the RCON Status Bits* from Related Links.

The RSWRST control register has only one bit, SWRST. This bit is used to force a software Reset condition.

A delay equal to the duration of the number of NMICNT system clocks begins as it is decremented to zero. During this interval, the program can clear the WDT or DMT flag bits, if desired, to avoid a

Reset. If the active flag is not cleared, the device will be reset at the end of the interval. The NMICNT value can be set to zero for no delay and up to 255 SYSCLK cycles.

The NMI interrupt can also be triggered by setting the SWNMI bit in software or if the CF bit is set by the FSCM, but these do not begin the countdown and do not automatically lead to a reset.

The Resets module consists of the following Special Function Registers (SFRs):

- RCON: Reset Control Register
- RSWRST: Software Reset Register
- RNMICON: Non-Maskable Interrupt (NMI) Control Register

The base address of these registers is 0x4400_0A00. The offset for each register is shown in Reset Register Map (see *Reset Register Map* in the *Reset Control Registers* from Related Links). Multiply the address offset in the table by 4. The other three addresses represent the CLR/SET/INV bitwise registers.

Related Links

[13.17.4. RCON](#)

[13.15.2. Reset Control Registers](#)

[13.15.4.3. Using the RCON Status Bits](#)

13.15.2 Reset Control Registers

Table 13-3. Reset Register Map

	Register	Bit Range	Bits																All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0XC	RCON	31:16	POR_IO	POR_CORE	—	—	BCFGERR	BCFGFAIL	NVMLTA	NVMEOL	—	—	—	—	—	—	—	VBAT	0000
		15:0	—	—	—	—	—	DPSLP	CMR	—	EXTR	SWR	DMTO	WDTO	SLEEP	IDLE	BOR	POR	0000
0X10	RSWRST	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST	0000
0X14	RNMICON	31:16	—	—	—	—	—	—	DMTO	WDTR	SWNMI	—	—	—	EXT	PLVD	CF	WDTS	0000
		15:0	NMICNT[15:0]																0000

13.15.3 Modes of Operation

13.15.3.1 System Reset (SYSRST)

The internal System Reset (SYSRST) can be generated from multiple Reset sources, such as:

- Power-on Reset (POR)
- Brown-out Reset (BOR/ZPBOR)
- Master Clear Reset ($\overline{\text{MCLR}}$)
- Watchdog Time-out Reset (WDTO)
- Deadman Timer Reset (DMTR)
- Software Reset (SWR)
- Configuration Mismatch Reset (CMR)
- Test Mode Entry and Exit Reset
- JTAG Reset

A system Reset is active at the first POR and asserted until device configuration settings are loaded and the oscillator clock sources become stable. The system Reset is then de-asserted, allowing the CPU to start fetching code after eight system clock cycles (SYSCLK). On any device Reset, no start-up time is required to transfer configuration values from the NVR memory into the configuration-holding registers. Once the device is active, the user may change the primary system clock source from FRC to SPLL by using the OSCCON register.

13.15.3.2 Power-on Reset (POR)

A power-on event generates an internal POR pulse when a VDD rise is detected above VPOR. The device supply voltage characteristics must meet the specified starting voltage and rise rate requirements to generate the POR pulse. In particular, VDD must fall below VPOR before a new POR is initiated. For more information on the VPOR and VDD rise-rate specifications, see *Electrical Characteristics* from Related Links.

This device has an on-chip internal voltage regulator and its power-on delay is designated as TPU. For more information on the TPU specification, see *Electrical Characteristics* from Related Links.

At this point, the POR event has expired but the device Reset is still asserted while the device configuration settings are loaded and the clock oscillator sources are configured. The clock monitoring circuitry waits for the oscillator source to become stable. The clock source of this device when exiting from Reset is always FRC (NOSC[3:0] bits (OSCCON[11:8])).

After these delays expire, the system Reset, SYSRST, is de-asserted. Before allowing the CPU to start code execution, eight system clock cycles are required before the synchronized Reset to the CPU core is de-asserted.

The power-on event sets the BOR and POR status bits (RCON[1:0]).

For more information on the values of the delay parameters, see *Electrical Characteristics* from Related Links.

Note: When the device exits the Reset condition (begins normal operation), the device operating parameters (voltage, frequency, temperature and so on) must be within their operating ranges; otherwise, the device will not function correctly. The user software must ensure that the delay between the time power is first applied and the time the system Reset is released is adequate to get all the operating parameters within the specification.

Related Links

[43. Electrical Characteristics](#)

13.15.3.3 Master Clear Reset

Whenever the master clear pin ($\overline{\text{MCLR}}$) is driven low, the Reset event is synchronized with the system clock, SYSCLK, before asserting the system Reset, SYSRST, provided the input pulse on $\overline{\text{MCLR}}$ is longer than a certain minimum width, as specified in the *Electrical Specifications*.

The $\overline{\text{MCLR}}$ pin provides a filter to minimize the effects of noise and to avoid unwanted Reset events. The status bit, EXTR (RCON[7]), is set to indicate the $\overline{\text{MCLR}}$ Reset.

The $\overline{\text{MCLR}}$ pin can be configured to generate a POR event, rather than a normal SYSRST event. This is configured through the SMCLR bit (CFGCON1[14]).

13.15.3.4 Software Reset (SWR)

This device does not provide a specific RESET instruction; however, a hardware Reset can be performed in software (software Reset) by executing a software Reset command sequence. The software Reset acts like a $\overline{\text{MCLR}}$ Reset. The software Reset sequence requires the system unlock sequence to be executed before the RSWRST bit (RSWRST[0]) can be written.

A software Reset is performed as follows:

1. Write the system unlock sequence.
2. Set the RSWRST bit (RSWRST[0]) = 1.
3. Read the RSWRST register.

Setting the RSWRST bit (RSWRST[0]) will arm the software Reset. The subsequent read of the RSWRST register triggers the software Reset, which must occur on the next clock cycle following the read operation. To ensure no other user code is executed before the Reset event occurs, it is recommended that four NOP instructions or a `while (1)` statement be placed after the READ instruction.

The SWR Status bit (RCON[6]) is set to indicate the software Reset.

13.15.3.5 Watchdog Timer Reset (WDTR)

A Watchdog Timer (WDT) Reset event is synchronized with the system clock (SYSCLK) before asserting the system Reset.

Note: A WDT time-out during the Sleep or Idle mode will wake-up the processor and branch to the reset vector, but it does not Reset the processor.

The only bits affected are WDTO and SLEEP or IDLE in the RCON register. See *Clock and Reset Unit (CRU)* from Related Links for more information on the WDT Reset.

Related Links

[13. Clock and Reset Unit \(CRU\)](#)

13.15.3.6 Brown-out Reset (BOR)

This device has a simple Brown-out Reset (BOR) capability. If the voltage supplied to the regulator is inadequate to maintain a regulated level, the regulator Reset circuitry will generate a BOR event, which is synchronized with the system clock, SYSCLK, before asserting the system Reset. This event is captured by the BOR flag bit (RCON[1]), see *Electrical Characteristics* from Related Links.

Related Links

[43. Electrical Characteristics](#)

13.15.3.7 Configuration Mismatch Reset (CMR)

To maintain the integrity of the stored configuration values, all device Configuration bits are loaded and implemented as a complementary set of bits. As the Configuration Words are being loaded, for each bit loaded as '1', a complementary value of '0' is stored into its corresponding background word location and vice versa. The bit pairs are compared every time the Configuration Words are loaded, including in Standby Sleep mode. During this comparison, if the Configuration bit values are not

found opposite to each other, a configuration mismatch event is generated, which causes a device Reset.

If a device Reset occurs as a result of a configuration mismatch, the CMR Status bit (RCON[9]) is set.

13.15.3.8 Deadman Timer Reset (DMTR)

A Deadman Timer (DMT) Reset is generated when the DMT count has expired.

The primary function of the DMT is to Reset the processor in the event of a software malfunction. The DMT is a free-running instruction fetch timer, which is clocked whenever an instruction fetch occurs until a count match occurs. Instructions are not fetched when the processor is in the Sleep mode.

The DMT consists of a 32-bit counter with a time-out count match value as specified by the DMTCNT[4:0] bits in the CFGCON2 Configuration register.

A DMT is typically used in mission critical and safety critical applications, where any single failure of the software functionality and sequencing must be detected. For more information on the DMT reset, see *Deadman Timer (DMT)* from Related Links.

Related Links

[17. Deadman Timer \(DMT\)](#)

13.15.3.9 Non-maskable Interrupt (NMI) Timer

The NMI timer provides a delay between DMT or WDT events and a device Reset. Set the delay in System Clock counts from 0 to 255 in the NMICNT[15:0] bits (RNMICON[15:0]). If these bits are set to zero, there will be no delay between the DMTO or WDTO flag and a device Reset. If set to a non-zero value, the NMI interrupt has that number of system clocks to clear flags or save data for debugging purposes.

If the corresponding NMI flag is not cleared in RNMICON before the counter reaches zero, then a device Reset will be issued. If the corresponding NMI flag in RNMICON is cleared before the counter reaches zero, then the counter is stopped, then reloaded with the NMICNT value again and waits for another NMI event to occur. A device Reset will not be asserted in this case and software can return from this interrupt.

The DMTO flag will be set if there is a DMT event. The device will be Reset after the NMI counter expires.

The WDTO flag will be set if there is a WDT event. The device will be Reset after the NMI counter expires.

The WDTS flag will be set if there is a WDT event during the Sleep mode. The WDTS flag will trigger the NMI interrupt but will not start the NMI counter nor cause a Reset.

The CF bit (RNMICON[17]) may be set by the Fail-Safe Clock Monitor (FSCM) if a clock failure is detected. The CF flag will trigger the NMI interrupt but will not start the timer nor cause a Reset.

The SWNMI bit (RNMICON[23]) can be set in software to cause an NMI interrupt but will not start the NMI counter nor cause a Reset.

13.15.3.10 Determining the Source of Device Reset

After a device Reset, the RCON register can be examined to confirm the source of the Reset. All reset status bits in the RCON register must be cleared after reading them to ensure the RCON value will provide meaningful results after the next device Reset.

13.15.4 Effects of Various Resets

The Reset value for the Reset Control register, RCON, will depend on the type of device Reset, as indicated in the following table.

Table 13-4. Status Bits, Their Significance and the Initialization Condition for RCON Register⁽¹⁾

Condition	Program Counter	EXTR	SWR	WDTO	DMTO	SLEEP ⁽²⁾	IDLE ⁽²⁾	CMR	BOR	POR
Power-on Reset or MCLR set as POR	—	0	0	0	0	0	0	0	1	1
Brown-out Reset		0	0	0	0	0	0	0	1	u
MCLR Reset during the Run mode		1	u	u	u	u	u	u	u	u
MCLR Reset during the Idle mode		1	u	u	u	u	1	u	u	u
MCLR Reset during the Sleep mode		1	u	u	u	1	u	u	u	u
Software Reset command		u	1	u	u	u	u	u	u	u
Configuration Word Mismatch Reset		u	u	u	u	u	u	1	u	u
WDT Time-out Reset during the Run mode		u	u	1	u	u	u	u	u	u
WDT Time-out Reset during the Idle mode		u	u	1	u	u	1	u	u	u
WDT Time-out Reset during the Sleep mode		u	u	1	u	1	u	u	u	u
DMT Time-out Reset		u	u	u	1	u	u	u	u	u
Interrupt Exit from the Idle mode	Vector	u	u	u	u	u	1	u	u	u
Interrupt Exit from the Sleep mode		u	u	u	u	1	u	u	u	u

1. Legends:

– u = unchanged

2. The SLEEP or IDLE states are entered when the correct sequence plus WAIT instruction is executed.

13.15.4.1 Special Function Register (SFR) Reset States

Most of the SFRs associated with the CPU and peripherals are reset to a particular value at a device Reset. This also applies to a WDT/DMT NMI condition, which is treated as a full device Reset by the CPU and peripherals.

The Reset value for the Reset Control register, RCON, is depending on the type of device Reset, see *Status Bits, Their Significance and Initialization Condition for RCON Register* table in *Effects of Various Resets* from Related Links.

Related Links

[13.15.4. Effects of Various Resets](#)

13.15.4.2 Configuration Word Register Reset States

All Reset conditions force the configuration settings to be reloaded. The POR and BOR reset all the Configuration Word registers before loading the configuration settings. For all other Reset conditions, the Configuration Word registers are not Reset prior to being reloaded.

13.15.4.3 Using the RCON Status Bits

The user software can read the RCON register after any system Reset to determine the cause of the reset. The following table provides a summary of the Reset flag bit operation.

Note: The status bits in the RCON register must be cleared after they are read, so that the next RCON register value after a device Reset will be meaningful.

Table 13-5. Reset Flag Bit Operation

Flag Bit	Set by	Cleared by
POR (RCON[0])	POR	User Software
BOR (RCON[1])	POR, BOR	User Software
IDLE (RCON[2])	WAIT Instruction	User Software, POR, BOR
STANDBY SLEEP (RCON[3])	WAIT Instruction	User Software, POR, BOR

.....continued

Flag Bit	Set by	Cleared by
WDTO (RCON[4])	WDT timeout and NMI counter expires	User Software, POR, BOR
DMTO (RCON[5])	DMT Timeout and NMI counter expires	User Software, POR, BOR
SWR (RCON[6])	Software Reset Command	User Software, POR, BOR
EXTR (RCON[7])	NMCLR Reset	User Software, POR, BOR
CMR (RCON[9])	Configuration Mismatch Reset	User Software, POR, BOR
BCFGFAIL (RCON[26])	Non-recoverable error in Primary and Alternate configuration words	User Software, POR, BOR
BCFGERR (RCON[27])	Recoverable error in primary configuration words	User Software, POR, BOR

13.15.5 CRU Configuration Registers

The BASE address of the CRU registers is 0x4400_0A00. The Register Summary table shows the mapping of the registers in memory as well as the details of the bit fields in each register. Each register has an associated SET/CLR/INV function register with the suffix appended to the register name, for example: <reg>SET, <reg>CLR, <reg>INV.

13.16 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	OSCCON	7:0	CLKLOCK			SLPEN	CF		SOSCEN	OSWEN
		15:8	COSC[3:0]				NOSC[3:0]			
		23:16	DRMEN		2SPDSL					
		31:24						FRCDIV[2:0]		
0x04 ... 0x0F	Reserved									
0x10	OSCTRM	7:0			TUN[5:0]					
		15:8								
		23:16								
		31:24								
0x14 ... 0x1F	Reserved									
0x20	SPLLCON	7:0			SPLL RST	SPLL FLOCK	SPLL PWDN			
		15:8	SPLL1POSTDIV1[7:0]							
		23:16					SPLL2POSTDIV2[3:0]			
		31:24	SPLL_BYP[1:0]							
0x24 ... 0x2F	Reserved									
0x30	RCON	7:0	EXTR	SWR	DMTO	WDTO	SLEEP	IDLE	BOR	POR
		15:8						DPSLP	CMR	
		23:16								VBAT
		31:24	POR_IO	POR_CORE			BCFGERR	BCFGFAIL	NVMLTA	NVMEOL
0x34 ... 0x3F	Reserved									
0x40	RSWRST	7:0								SWRST
		15:8								
		23:16								
		31:24								
0x44 ... 0x4F	Reserved									
0x50	RNMICON	7:0	NMICNT[7:0]							
		15:8	NMICNT[15:8]							
		23:16	SWNMI				EXT	PLVD	CF	WDTS
		31:24							DMTO	WDTR
0x54 ... 0x6F	Reserved									
0x70	REFO1CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
0x74 ... 0x7F	Reserved									
0x80	REFO1TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0x84 ... 0x8F	Reserved									
0x90	REFO2CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x94 ... 0x9F	Reserved									
0xA0	REFO2TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0xA4 ... 0xAF	Reserved									
0xB0	REFO3CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
0xB4 ... 0xBF	Reserved									
0xC0	REFO3TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0xC4 ... 0xCF	Reserved									
0xD0	REFO4CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
0xD4 ... 0xDF	Reserved									
0xE0	REFO4TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0xE4 ... 0xEF	Reserved									
0xF0	REFO5CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
0xF4 ... 0xFF	Reserved									
0x0100	REFO5TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0x0104 ... 0x010F	Reserved									
0x0110	REFO6CON	7:0					ROSEL3	ROSEL2	ROSEL1	ROSEL0
		15:8	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
		23:16	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
		31:24		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
0x0114 ... 0x011F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0120	REFO6TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0x0124	Reserved									
...										
0x012F										
0x0130	PB1DIV	7:0		PB1DIV[6:0]						
		15:8	PB1DIVON				PB1DIVRDY			
		23:16								
		31:24								
0x0134	Reserved									
...										
0x013F										
0x0140	PB2DIV	7:0		PB1DIV[6:0]						
		15:8	PB1DIVON				PB1DIVRDY			
		23:16								
		31:24								
0x0144	Reserved									
...										
0x014F										
0x0150	PB3DIV	7:0		PB1DIV[6:0]						
		15:8	PB1DIVON				PB1DIVRDY			
		23:16								
		31:24								
0x0154	Reserved									
...										
0x015F										
0x0160	SLEWCON	7:0						SLW_UP	SLW_DN	SLW_BUSY
		15:8						SLW_DIV[2:0]		
		23:16						SYS_DIV[3:0]		
		31:24						SLW_DELAY[3:0]		
0x0164	Reserved									
...										
0x016F										
0x0170	CLKSTAT	7:0		SPLL3RDY		LPRCRDY	SOSCRDY	POSCRDY	SPLL1RDY	FRCRDY
		15:8								
		23:16								
		31:24								
0x0174	Reserved									
...										
0x018F										
0x0190	CLK_DIAG	7:0		SPLL3_STOP	SPLL2_STOP	SPLL1_STOP	LPRC_STOP	FRC_STOP	SOSC_STOP	POSC_STOP
		15:8								
		23:16	NMICTR7	NMICTR6	NMICTR5	NMICTR4	NMICTR3	NMICTR2	NMICTR1	NMICTR0
		31:24	NMICTR15	NMICTR14	NMICTR13	NMICTR12	NMICTR11	NMICTR10	NMICTR9	NMICTR8

13.17 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the “Write-Synchronized” or the “Read-Synchronized” property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

Note: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET, and INV Registers* from Related Links.

Related Links

[6.1.9. CLR, SET and INV Registers](#)

13.17.1 CRU Oscillator Control

Name: OSCCON
Offset: 0x00
Reset: 0x00000000

Note: The system unlock sequence must be done before this register can be written.

Bit	31	30	29	28	27	26	25	24
						FRCDIV[2:0]		
Access						R/W/L	R/W/L	R/W/L
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DRMEN		2SPDSL					
Access	R/W/L		R/W/L					
Reset	0		1					

Bit	15	14	13	12	11	10	9	8
	COSC[3:0]				NOSC[3:0]			
Access	R	R	R	R	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKLOCK			SLPEN	CF		SOSCEN	OSWEN
Access	R/W/L			R/W/L	R/W/HS/L		R/W/L	R/W/HC/L
Reset	0			0	0		1	1

Bits 26:24 – FRCDIV[2:0] Fast RC Clock Divider bits

Value	Description
000	FRC Divide by 1 (default value)
001	FRC Divide by 2
010	FRC Divide by 4
011	FRC Divide by 8
100	FRC Divide by 16
101	FRC Divide by 32
110	FRC Divide by 64
111	FRC Divide by 256

Bit 23 – DRMEN Enable the Dream Mode bit

Value	Description
1	When the cpu has executed WFI instruction and SLPEN = 1, peripheral clock requests are NOT active causes to enter the Sleep mode
0	DMA transfer has no effect

Bit 21 – 2SPDSL 2-Speed Start-up enabled in the Sleep mode bit

Note: Default Reset Value is specified by cfg_two_speed_startup_en input.

Value	Description
1	When the device exits the Sleep Mode, the SYS_CLK will be from FRC until the selected clock is ready
0	When the device exits the Sleep Mode, the SYS_CLK will be from the selected clock

Bits 15:12 – COSC[3:0] Current Oscillator Selection bits (Read-only)

Notes:

- The default value on Reset is 4'b0000, which ensures that a virgin die has frc_clk running for ICDJTAG or EJTAG to program the NVR.
- Loaded with NOSC[3:0] at the completion of a successful clock switch.
- Set to FRC value (0000) when FSCM detects a failure and switches clock to FRC.

Value	Description
0000	Fast RC Oscillator (FRC) divided by OSCCON.FRCDIV
0001	System PLL Clock-1 (SPLL1 Module) (input clock and divider set by SPLLCON)
0010	Primary Oscillator (POSC)
0011	Secondary Oscillator (SOSC)
0100	Low Power RC Oscillator (LPRC)
0101–1111	Reserved for future use

Bits 11:8 – NOSC[3:0] New Oscillator Selection bits

Note: Default value on Reset is 4'b0000, which ensures that a virgin die has frc_clk running for ICDJTAG or EJTAG to program the NVR.

Value	Description
0000	Fast RC Oscillator (FRC) divided by OSCCON.FRCDIV
0001	System PLL Clock-1 (SPLL1 Module) (input clock and divider set by SPLLCON)
0010	Primary Oscillator (POSC)
0011	Secondary Oscillator (SOSC)
0100	Low Power RC Oscillator (LPRC)
0101–1111	Reserved for future use

Bit 7 – CLKLOCK Clock Lock Enabled bit

Notes:

- Once set, this bit can only be cleared via a Device Reset.
- When active, this bit prevents writes to the following registers: NOSC[3:0] and OSWEN.

Value	Description
1	All clock and PLL configuration registers are locked. These include OSCCON, OSCTRIM, SPLLCON, UPLLCON, PBxDIV
0	Clock and PLL selection registers are not locked, configurations may be modified.

Bit 4 – SLPEN Enable the Sleep Mode bit

Value	Description
1	When a WAIT Instruction is executed device will enter SLEEP Mode
0	When a WAIT instruction is executed device will enter IDLE Mode

Bit 3 – CF Clock Fail Detect bit (Read/writable/Clearable by application)

Notes:

- Writing a '1' to this bit will cause a clock switching sequence to be initiated by the clock switch state machine
- Resets when a valid clock switching sequence is initiated by the clock switch state machine
- This bit is set when clock fail event is detected

Value	Description
1	FSCM has detected clock failure
0	FSCM has not detected clock failure

Bit 1 – SOSCEM 32 kHz Secondary (LP) Oscillator Enable bit

Value	Description
1	Enable Secondary Oscillator
0	Disable Secondary Oscillator

Bit 0 – OSWEN Oscillator Switch Enable bit

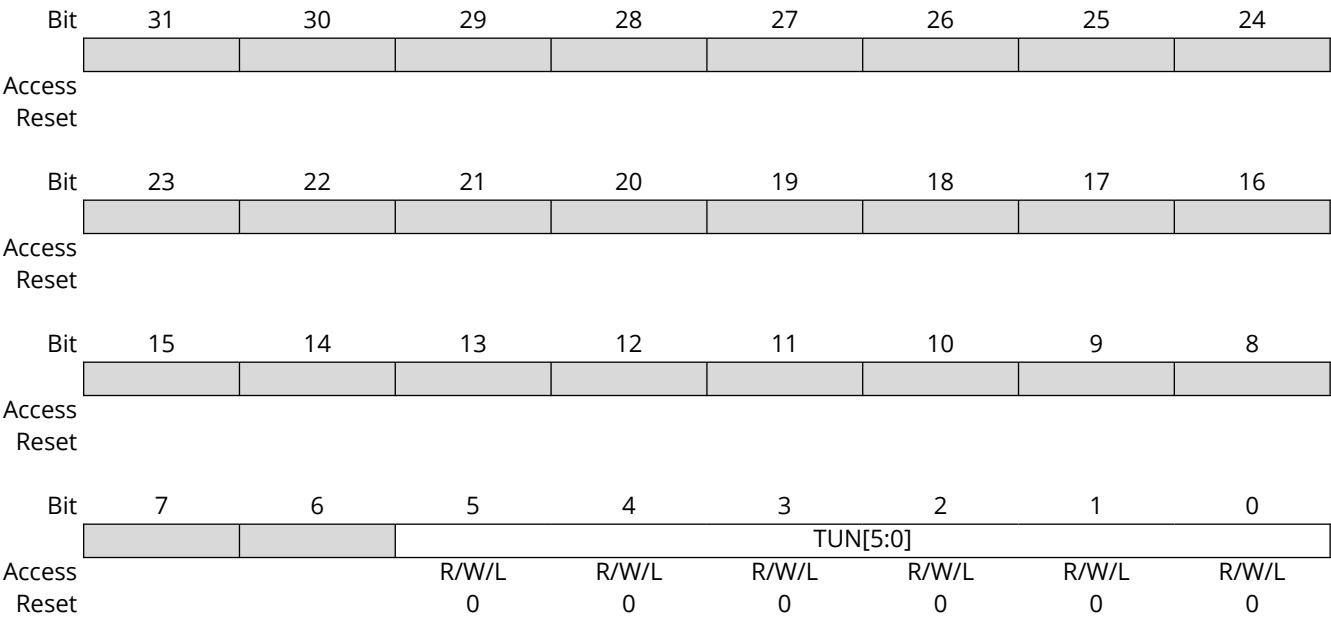
Notes:

- A Write of value '0' has no effect.
- Cleared by hardware after a successful clock switch
- Cleared by hardware after a redundant clock switch (NOSC = COSC)
- Cleared by hardware after FSCM switches the oscillator to Fail-Safe Clock Source (FRC)

Value	Description
1	Request oscillator switch to selection specified by NOSC[3:0] bits
0	Oscillator switch is complete

13.17.2 CRU Oscillator Trimming

Name: OSCTRM
Offset: 0x10
Reset: 0x00000000



Bits 5:0 – TUN[5:0] Internal Fast RC (FRC) Oscillator Tuning bits
This bit field specifies the user-tuning capability for the internal fast RC oscillator.
Note: The system unlock sequence must be done before this register can be written.

Value	Description
011111	Maximum Frequency
011110	
...	
000001	
000000	Center Frequency, oscillator is running at calibrated frequency
111111	
111110	
...	
100001	
100000	Minimum Frequency

13.17.3 SPLL (RFPLL/Wrapper) Control

Name: SPLLCON
Offset: 0x20
Reset: 0x00000000

Note: The system unlock sequence must be done before these registers can be written.

Bit	31	30	29	28	27	26	25	24
	SPLL_BYP[1:0]							
Access	R/W/L	R/W/L						
Reset	1	1						
Bit	23	22	21	20	19	18	17	16
					SPLL2POSTDIV2[3:0]			
Access					R/W/L	R/W/L	R/W/L	R/W/L
Reset					0	0	0	1
Bit	15	14	13	12	11	10	9	8
	SPLL1POSTDIV1[7:0]							
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
			SPLL_RST	SPLL_FLOCK	SPLL_PWDN			
Access			R/W/L	R/W/L	R/W/L			
Reset			1	0	1			

Bits 31:30 – SPLL_BYP[1:0] SPLL Bypass; when this bit is set, the input clock REF bypasses PLL to PLLOUTx.

Notes:

- Dictates clock source for ADC CP (Analog-to-Digital Converter Charge Pump) (SPLL2) Clock generation only
- Clock source must be preselected and kept ready before the need of ADC CP arrives. Failure to do so will result in the loss of clock for one or two cycles when ADC CP is enabled.

Value	Description
00	RFPLL Clock is the clock source for ADC CP clock generation.
x1	FRC is used as clock source for ADC CP clock generation.
10	POSC is used as clock source for ADC CP clock generation.

Bits 19:16 – SPLL2POSTDIV2[3:0] ADC-CP Post Divide Value

Value	Description
1 ≤ SPLL2POSTDIV2 ≤ 15	Divide-by SPLL2POSTDIV2
0	No Clock; Clock disabled

Bits 15:8 – SPLL1POSTDIV1[7:0] First Post Divide Value

Value	Description
2 ≤ SPLL1POSTDIV1 ≤ 255	Divide-by SPLL1POSTDIV1
0	Divide-by 1
1	Divide-by 1.5

Bit 5 – SPLLRST System PLL Reset

Value	Description
1	Assert the Reset to the SPLL
0	De-assert the Reset to the SPLL

Bit 4 – SPLLFLOCK System PLL Force Lock

Value	Description
1	Force the SPLL lock signal to be asserted
0	Do not force the SPLL lock signal to be asserted

Bit 3 – SPLLPWDN PLL Power Down Register bit

Value	Description
1	PLL is powered down
0	PLL is active

13.17.4 Reset Control Register

Name: RCON
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	POR_IO	POR_CORE			BCFGERR	BCFGFAIL	NVMLTA	NVMEOL
Access	R/W/HS	R/W/HS			R/W/HS	R/W/HS	R/W/HS	R/W/HS
Reset	0	0			0	0	0	0
Bit	23	22	21	20	19	18	17	16
								VBAT
Access								R/W/HS
Reset								0
Bit	15	14	13	12	11	10	9	8
						DPSLP	CMR	
Access						R/W/HS	R/W/HS	
Reset						0	0	
Bit	7	6	5	4	3	2	1	0
	EXTR	SWR	DMTO	WDTO	SLEEP	IDLE	BOR	POR
Access	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS
Reset	0	0	0	0	0	0	0	0

Bit 31 – POR_IO I/O Voltage POR Flag bit

Set by hardware at detection of an I/O POR event. User software must clear this bit to view next detection.

Note: User may write this bit to '1'. Does not cause a POR_IO.

Value	Description
1	A Power-on Reset has occurred due to I/O voltage
0	A Power-on Reset has not occurred due to I/O voltage

Bit 30 – POR_CORE Core Voltage POR Flag bit

Set by hardware at detection of a core POR event. User software must clear this bit to view the next detection.

Note: User may write this bit to '1'. Does not cause a POR_CORE.

Value	Description
1	A Power-on Reset has occurred due to I/O voltage
0	A Power-on Reset has not occurred due to I/O voltage

Bit 27 – BCFGERR BCFG Error Flag bit

A primary BCFG value had an error, but the secondary BCFG value was valid and used.

Value	Description
1	A BCFG error has occurred
0	A BCFG error has not occurred

Bit 26 – BCFGFAIL BCFG Failure Flag bit

Both the Primary and Secondary BCFG values had an unrecoverable error. Default values are in effect.

Value	Description
1	A BCFG error has occurred
0	A BCFG error has not occurred

Bit 25 – NVMLTA NVM Life Time Alert Flag bit

NVM Life Time Alert – Due to charge leakage, the NVM is nearing EOL.

Value	Description
1	A NVM LTA error has occurred
0	A NVM LTA error has not occurred

Bit 24 – NVMEOL NVM End of Life Flag bit

NVM End of Life – may not be visible to user, because the part will not come out of Reset if the bit is asserted.

Value	Description
1	A NVM EOL failure has occurred
0	A NVM EOL failure has not occurred

Bit 16 – VBAT VBAT Mode Flag bit

Value	Description
1	A POR exit from VBAT has occurred. A true POR must be established with the valid VBAT voltage level on the VBAT pin.
0	A POR exit from VBAT has not occurred.

Bit 10 – DPSLP Deep Sleep Mode Flag bit

Set by hardware at time of entry into Deep Sleep mode. User software must clear this bit to view next detection.

Value	Description
1	Deep Sleep mode has occurred
0	Deep Sleep mode has not occurred

Bit 9 – CMR Configuration Mismatch Reset Flag bit

Note: User may write this bit to '1'. Does not cause a Mismatch Reset.

Value	Description
1	A CMR event has occurred
0	A CMR event has not occurred

Bit 7 – EXTR External Reset ($\overline{\text{MCLR}}$) Status bit

Note: User may write this bit to '1'. Does not cause a ($\overline{\text{MCLR}}$).

Value	Description
1	A Master Clear (pin) Reset has occurred
0	A Master Clear (pin) Reset not occurred

Bit 6 – SWR Software Reset Flag bit

Note: User may write this bit to '1'. Does not cause SWR.

Value	Description
1	A SWR has occurred
0	A SWR not occurred

Bit 5 – DMT0 Deadman Timer Time-out Flag bit

Note: User may write this bit to '1'. Does not cause DMT Reset.

Value	Description
1	DMT Time-out has occurred and caused a Reset
0	DMT Time-out has not occurred

Bit 4 – WDTO Watchdog Timer Time-out Flag bit

Note: User may write this bit to '1'. Does not cause WDT Reset.

Value	Description
1	WDT Time-out has occurred and caused a Reset
0	WDT Time-out has not occurred

Bit 3 – SLEEP Wake from Sleep Flag bit

Note: User may write this bit to '1'. Does not invoke Sleep mode.

Value	Description
1	Device has been in Sleep mode
0	Device has not been in Sleep mode

Bit 2 – IDLE Wake from Idle Flag bit

Note: User may write this bit to '1'. Does not invoke Idle mode.

Value	Description
1	Device was in the Idle mode
0	Device was not in the Idle mode

Bit 1 – BOR BOR Flag bit

Set by hardware at detection of a BOR event. User software must clear this bit to view next detection.

Note: User may write this bit to '1'. Does not cause a BOR.

Value	Description
1	A BOR has occurred
0	A BOR has not occurred

Bit 0 – POR POR Flag bit

Set by hardware at detection of a POR event. User software must clear this bit to view next detection.

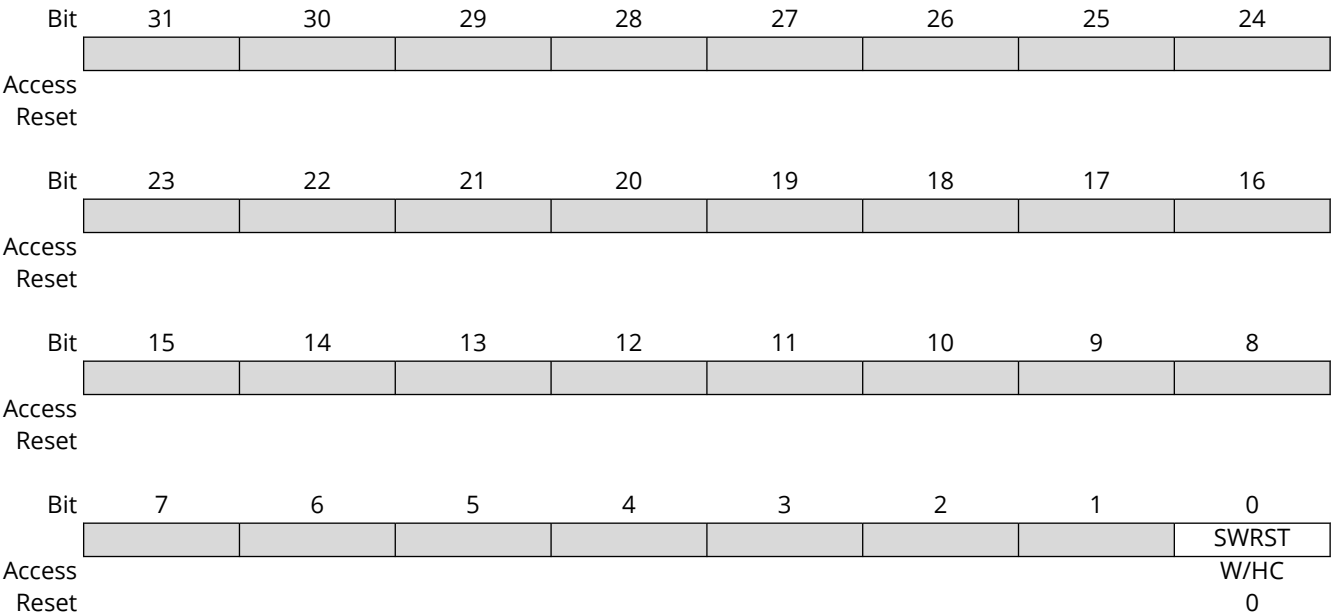
Note: User may write this bit to '1'. Does not cause a POR.

Value	Description
1	A Power-on Reset has occurred
0	A Power-on Reset has not occurred

13.17.5 Software Reset Register

Name: RSWRST
Offset: 0x40
Reset: 0x00000000
Property: -

Note: The system unlock sequence must be done before this register can be written.



Bit 0 – SWRST Software Reset Trigger bit
‘1’ = Enable SWR event. A subsequent read of this register triggers the system Reset sequence. The system unlock sequence must be done before the bit can be written. This bit always reads a value of logic ‘0’.

13.17.6 NMI Control Register

Name: RNMICON
Offset: 0x50
Reset: 0x00000000
Property: -

Note: The system unlock sequence must be done before this register can be written.

Bit	31	30	29	28	27	26	25	24
							DMTO	WDTR
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	SWNMI				EXT	PLVD	CF	WDTS
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NMICNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NMICNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – DMTO Deadman Timer Time-out Flag bit (This will cause a Reset when NMICNT expires.)

Note: User may write this bit to '1'. Causes a user-initiated DMT NMI event and NMICNT start.

Value	Description
1	DMT Time-out has occurred and caused a NMI
0	DMT Time-out has not occurred

Bit 24 – WDTR Watchdog Timer Time-out in Run Flag bit

Note: User may write this bit to '1'. Causes a user-initiated WDT NMI event and NMICNT start.

Value	Description
1	WDT Time-out has occurred and caused an NMI (This may cause a Reset if NMICNT expires.)
0	WDT Time-out has not occurred

Bit 23 – SWNMI Software NMI Trigger bit

Value	Description
1	Writing a '1' to this bit will cause an NMI to be generated
0	Writing a '0' to this bit will have no effect

Bit 19 – EXT External / Generic NMI Event bit

Note: User may write this bit to '1'. Causes a user-initiated EXT NMI event.

Value	Description
1	A general NMI event was detected and caused an NMI. Writing '0' to this bit will clear the NMI event
0	A general NMI event was not detected

Bit 18 – PLVD Programmable Low Voltage Detect Event bit

Note: User may write this bit to '1'. Causes a user-initiated PLVD NMI event.

Value	Description
1	PLVD detected a low voltage condition and caused an NMI
0	PLVD did not detect a low voltage condition

Bit 17 – CF Clock Fail Detect bit (Read/Clear-able by application)

Note: Writing a '1' to the CF bit will cause a user-initiated clock failure NMI event, but will not actually cause a clock switch.

Value	Description
1	FSCM detected clock failure and caused an NMI
0	FSCM did not detect a clock failure

Bit 16 – WDTS Watch-Dog Timer Time-out in Sleep Flag bit

Note: User may write this bit to '1'. Causes a user-initiated WDT NMI event.

Value	Description
1	WDT Time-out has occurred during Sleep mode and caused a wake-up from sleep
0	WDT Time-out has not occurred during Sleep mode

Bits 15:0 – NMICNT[15:0] NMI Reset counter value bit

This bit field specifies the reload value used by the NMI Reset counter.

```
0000_0000_0000_0000 = No delay between NMI assertion and device Reset event
0000_0000_0000_0001
0000_0000_0000_0010
.....
.....
.....
1111_1111_1111_1110
1111_1111_1111_1111 = Number of SYSCLK cycles that Software has to clear the NMI event before
a device Reset is performed. If the NMI event is cleared before the counter reached zero,
then NO device Reset is asserted.
```

Note: When a WDT NMI event occurs (when not in the Sleep mode), the NMICNT starts incrementing from the zero NMICNT value. When a DMT NMI event is triggered, the NMICNT starts decrementing. When NMICNT reaches zero, the device is Reset. This NMI reset counter is only applicable to these two specific NMI events.

13.17.7 Reference Oscillator x Control

Name: REFOxCON
Offset: 0x70 + (x-1)*0x20 [x=1..6]
Reset: 0x00000000

Notes:

- REFOCON.ROSEL must not be written while the REFOCON.ACTIVE bit is '1' – This will result in undefined behavior.
- REFOCON must not be written when REFOCON[ON] != REFOCON[ACTIVE] – This will result in undefined behavior.
- This register can always be accessed regardless of the cfg_sys_unlock value.

Bit	31	30	29	28	27	26	25	24
		RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ON	FRZ	SIDL	OE	RSLP		DIVSW_EN	ACTIVE
Access	R/W	R/W	R/W	R/W	R/W		HC/ R/W	HS/HC/ R
Reset	0	0	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
					ROSEL3	ROSEL2	ROSEL1	ROSEL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 – RODIV Reference Clock Divider bits

Specifies 1/2 period of reference clock in the source clocks.

Example: Period of refo_clk = [Reference source * 2] * RODIV

Value	Description
1111111111111111	REFOx clock is Base clock frequency divided by 65,534 (32,767 * 2)
1111111111111110	REFOx clock is Base clock frequency divided by 65,532 (32,766 * 2)
...	
...	
...	
0000000000000011	REFOx clock is Base clock frequency divided by 6 (3*2)
0000000000000010	REFOx clock is Base clock frequency divided by 4 (2*2)
0000000000000001	REFOx clock is Base clock frequency divided by 2 (1*2)
0000000000000000	REFOx clock is same frequency as Base clock (no divider)

Bit 15 – ON Output Enable bit

Value	Description
1	Reference Oscillator Module is enabled
0	Reference Oscillator Module is disabled

Bit 14 – FRZ Freeze in Debug mode bit

Value	Description
1	When the emulator is in the Debug mode, module freezes operation
0	When the emulator is in the Debug mode, module continues operation

Bit 13 – SIDL Peripheral Stop in Idle Mode bit

Value	Description
1	Discontinues module operation when device enters the Idle mode
0	Continues module operation in the Idle mode

Bit 12 – OE Reference Clock Output Enable bit

Value	Description
1	Reference clock is driven out on REFOx pin
0	Reference clock is not driven out on REFOx pin

Bit 11 – RSLP Reference Oscillator Run in Sleep bit

Note: This bit is ignored when ROSEL[3:0] = (0000 or 0001).

Value	Description
1	Reference Oscillator output continues to run in the Sleep mode
0	Reference Oscillator output is disabled in the Sleep mode

Bit 9 – DIVSW_EN Clock RODIV/ROTRIM switch enabled

Value	Description
1	Clock Divider Switching is currently in progress
0	Clock Divider Switch has completed

Bit 8 – ACTIVE Reference Clock Request Status bit

Value	Description
1	Reference clock request is active (User must not update this REFOCON register)
0	Reference clock request is not active (User can update this REFOCON register)

Bits 0, 1, 2, 3 – ROSEL Reference Clock Source Select bits

Select one of various clock sources to be used as the reference clock.

Value	Description
1001-1111	Reserved
1000	REFI pin
0111	System clock (reference clock reflects any device clock switching)
0110	Peripheral clock (reference clock reflects any peripheral clock switching)
0101	System PLL (Clock-3)
0100	LPRC
0011	SOSC
0010	POSC
0001	System PLL (Clock-1)
0000	FRC

13.17.8 Reference Oscillator x Trim

Name: REFOxTRIM
Offset: $0x80 + (x-1)*0x20$ [x=1..6]
Reset: 0x00000000

Notes:

- REFOxTRIM must not be written when REFOxCON[ON] != REFOxCON[ACTIVE] – This will result in undefined behavior.
- This register can always be accessed regardless of the cfg_sys_unlock value.

Bit	31	30	29	28	27	26	25	24
	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ROTRIM0							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23, 24, 25, 26, 27, 28, 29, 30, 31 – ROTRIM Trim bits – Provides fractional additive to RODIV value for 1/2 period of REFOx clock

Note: ROTRIM values greater than zero are only valid when RODIV values are greater than 0.

Value	Description
0000_0000_0	0/512 (0.0) divisor added to RODIV value
0000_0000_1	1/512 (0.001953125) divisor added to RODIV value
0000_0001_0	2/512 (0.00390625) divisor added to RODIV value
...	...
...	...
100000000	256/512 (0.5000) divisor added to RODIV value
...	...
...	...
1111_1111_0	510/512 (0.99609375) divisor added to RODIV value
1111_1111_1	511/512 (0.998046875) divisor added to RODIV value

13.17.9 PBx Clock Divisor Control

Name: PBxDIV
Offset: 0x0130 + (x-1)*0x10 [x=1..3]
Reset: 0x00000000

Note: The system unlock sequence must be done before this register can be written. PBx registers include PB1DIV, PB2DIV and PB3DIV. Ensure the PB3DIV[6:0] value is equal to 0x09 or greater if the user is not using Microchip-provided boot code.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access	PB1DIVON					PB1DIVRDY		
Reset	R		R					
	1		1					
Bit	7	6	5	4	3	2	1	0
Access	PB1DIV[6:0]							
Reset		R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0	0	0	0	0	0	1

Bit 15 – PB1DIVON Output Enable bit

Value	Description
1	PB1 Output clock is enabled
0	PB1 Output clock is disabled Note: PB1DIV[PB1DIVON] bit cannot be written to a '0', as this clock is used by the system CLK_RST macro.

Bit 11 – PB1DIVRDY PB1 Peripheral Clock Divisor Ready

Value	Description
1	Indicates the PB clock divisor logic is not switching divisors and the PB1DIV may be written.
0	Indicates the PB clock divisor logic is currently switching values and the PB1DIV cannot be written.

Bits 6:0 – PB1DIV[6:0] PB1 Peripheral Clock Divisor Control value

Value	Description
000_0000	Divide by 1 PB1 Clock same frequency as SYS_CLK
000_0001	Divide by 2 PB1 Clock is 1/2 of SYS_CLK
000_0010	Divide by 3 PB1 Clock is 1/3 of SYS_CLK
000_0011	Divide by 4 PB1 Clock is 1/4 of SYS_CLK
...	...
...	...
000_1111	Divide by 16 PB1 Clock is 1/16 of SYS_CLK
001_0000	Divide by 17 PB1 Clock is 1/17 of SYS_CLK
...	...
...	...
111_1110	Divide by 127 PB1 Clock is 1/127 of SYS_CLK
111_1111	Divide by 128 PB1 Clock is 1/128 of SYS_CLK

13.17.10 Slew Rate Control for Clock Switching

Name: SLEWCON
Offset: 0x160
Reset: 0x00000000

Notes:

- The system unlock sequence must be done before this register can be written.
- Updates to this register do not take effect until OSCCON[OSWEN] is set.

Bit	31	30	29	28	27	26	25	24
						SLW_DELAY[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					c	c	c	c
Bit	23	22	21	20	19	18	17	16
						SYS_DIV[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					c	c	c	c
Bit	15	14	13	12	11	10	9	8
							SLW_DIV[2:0]	
Access						R/W	R/W	R/W
Reset						c	c	c
Bit	7	6	5	4	3	2	1	0
						SLW_UP	SLW_DN	SLW_BUSY
Access						R/W	R/W	R/W
Reset						c	c	c

Bits 27:24 – SLW_DELAY[3:0] Number of clocks generated at each slew step for a clock switch

Note: The reset value of this register field is defined by the input `cfg_slewcon_sel[]`.

Value	Description
0000	1 clock will be generated at each slew step
0001	2 clocks will be generated at each slew step
...	...
1111	16 clocks will be generated at each slew step

Bits 19:16 – SYS_DIV[3:0] PBx Peripheral Clock Divisor Control value

Value	Description
0000	Divide by 1 – SYS_CLK_OUT same frequency as SYS_CLK source - Default
0001	Divide by 2 – SYS_CLK_OUT is 1/2 of SYS_CLK source
0010	Divide by 3 – SYS_CLK_OUT is 1/3 of SYS_CLK source
...	...
1111	Divide by 16 – SYS_CLK_OUT is 1/16 of SYS_CLK source

Bits 10:8 – SLW_DIV[2:0] Divisor steps used when doing slewed clock switches

Note: Each Divisor step lasts four clocks

Value	Description
000	No divisor is used
001	Divide by 2 (2^1), then no divisor
010	Divide by 4 (2^2), then by 2, then no divisor
011	Divide by 8 (2^3), then by 4, then by 2, then no divisor

Value	Description
100	Divide by 16 (2^4), then by 8, then by 4, then by 2, then no divisor
...	...
111	Divide by 128 (2^7), then by 64, then by 32, then by 16, then by 8, then by 4, then by 2, then no divisor

Bit 2 – SLW_UP Clock slew enable for switching up to faster clocks

Value	Description
0	Clock Slewing is disabled
1	Clock Slewing is enabled on a clock switch OR exit from Sleep

Bit 1 – SLW_DN Clock slew enable for switching down to slower clocks

Value	Description
0	Clock Slewing is disabled
1	Clock Slewing is enabled on a clock switch

Bit 0 – SLW_BUSY Clock Switch Slewing Active Status Bit – Read-Only

Value	Description
0	Clock Switch has reached its final value
1	Clock frequency is being actively Slewed

13.17.11 Clock Status

Name: CLKSTAT
Offset: 0x170
Reset: 0x00000000

Note: The corresponding RDY bits are updated only after the clock switch request is initiated via OSCCON.NOSC[3:0].

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SPLL3RDY		LPRCRDY	SOSCRDY	POSCRDY	SPLL1RDY	FRCRDY
Access		R/HS/HC		R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset		0		0	0	0	0	0

Bit 6 – SPLL3RDY System PLL (Clock-3) Ready Status value

Value	Description
1	SPLL3 is stable and ready
0	SPLL3 is not stable and not ready

Bit 4 – LPRCRDY LPRC Ready Status value

Value	Description
1	LPRC is stable and ready
0	LPRC is not stable and not ready

Bit 3 – SOSCRDY SOSC Ready Status value

Value	Description
1	SOSC is stable and ready
0	SOSC is not stable and not ready

Bit 2 – POSCRDY Primary Oscillator Ready Status value

Value	Description
1	POSC is stable and ready
0	POSC is not stable and not ready

Bit 1 – SPLL1RDY System PLL (Clock-1) Ready Status value

Value	Description
1	SPLL1 is stable and ready
0	SPLL1 is not stable and not ready

Bit 0 – FRCRDY FRC Ready Status value

Value	Description
1	FRC is stable and ready
0	FRC is not stable and not ready

13.17.12 User Clock Diagnostics Control

Name: CLK_DIAG
Offset: 0x190
Reset: 0x00000000

Note: The system unlock sequence must be done before this register can be written.

Bit	31	30	29	28	27	26	25	24
	NMICTR15	NMICTR14	NMICTR13	NMICTR12	NMICTR11	NMICTR10	NMICTR9	NMICTR8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NMICTR7	NMICTR6	NMICTR5	NMICTR4	NMICTR3	NMICTR2	NMICTR1	NMICTR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SPLL3_STOP	SPLL2_STOP	SPLL1_STOP	LPRC_STOP	FRC_STOP	SOSC_STOP	POSC_STOP
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NMICTR Internal value of internal NMI Counter

This field reflects the actual value of the internal NMI counter

Bit 6 – SPLL3_STOP SPLL Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	SPLL3 clock source runs as normal
1	SPLL3 clock source is stopped

Bit 5 – SPLL2_STOP SPLL Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	SPLL2 clock source runs as normal
1	SPLL2 clock source is stopped

Bit 4 – SPLL1_STOP SPLL Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	SPLL1 clock source runs as normal
1	SPLL1 clock source is stopped

Bit 3 – LPRC_STOP LPRC Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	LPRC clock source runs as normal
1	LPRC clock source is stopped

Bit 2 – FRC_STOP FRC Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	FRC clock source runs as normal
1	FRC clock source is stopped

Bit 1 – SOSC_STOP SOSC Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	SOSC clock source runs as normal
1	SOSC clock source is stopped

Bit 0 – POSC_STOP POSC Clock Stop Control value

Note: Gating logic is outside of this macro

Value	Description
0	POSC clock source runs as normal
1	POSC clock source is stopped

14. RAM Error Correction Code (RAMECC)

14.1 Overview

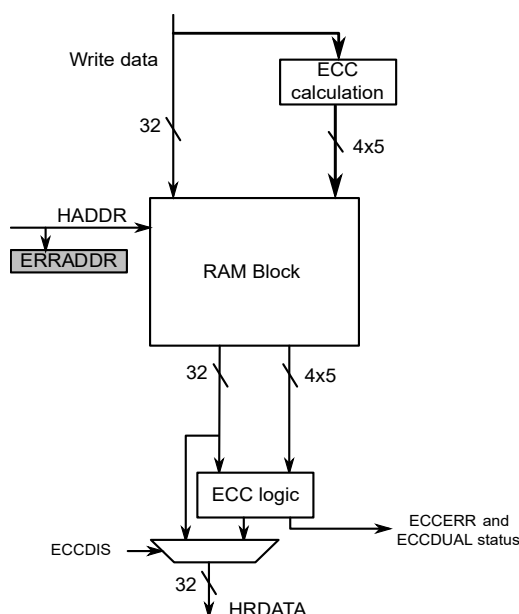
Single bit error correction and dual bit error detection is available for RAM.

14.2 Features

- Single bit correction and dual bit detection.
- Error Interrupt.
- Operates in any Sleep mode.
- Interrupts generated by RAMECC can be used to wake up the device from sleep modes.

14.3 Block Diagram

Figure 14-1. RAMECC Block Diagram



14.4 Signal Description

Not applicable.

14.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

14.5.1 I/O Lines

Not applicable.

14.5.2 Power Management

The RAMECC will continue to operate in any sleep mode where the selected source clock is running. The RAMECC's interrupts can be used to wake up the device from sleep modes. See *Power Management Unit (PMU)* from Related Links for details on the different sleep modes.

Related Links[15. Power Management Unit \(PMU\)](#)**14.5.3 DMA**

Not applicable.

14.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the RAMECC interrupt(s) requires the interrupt controller to be configured first.

14.5.5 Events

Not applicable.

14.5.6 Debug Operation

When the CPU is halted in debug mode the RAMECC will correct and log ECC errors based on the table below.

Table 14-1. ECC Debug Operation

DBGCTRL.ECCELOG	DBGCTRL.ECCDIS	Description
0	0	ECC errors from debugger reads are corrected but not logged in INTFLAG.
1	0	ECC errors from debugger reads are corrected and logged in INTFLAG.
X	1	ECC errors from debugger reads are not corrected or logged in INTFLAG.

If the RAMECC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

14.5.7 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register
- Status (STATUS) register.

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger (see *Peripheral Access Controller (PAC)* from Related Links).

Related Links[26. Peripheral Access Controller \(PAC\)](#)**14.5.8 Analog Connections**

Not applicable.

14.6 Functional Description**14.6.1 Principle of Operation**

Error Correcting Code (ECC) is implemented to detect and correct errors that may arise in the RAM arrays. The ECC logic is capable of double error detection and single error correction per 8-bit byte.

Upon single bit error detection, the Single Bit Error interrupt flag is raised (INTFLAG.SINGLEEE). If a dual error is detected, the Dual Error interrupt flag (INTFLAG.DUALE) is raised. When the first

error is detected, the ERRADDR register is frozen with the failing address and remains frozen until INTFLAG.DUALE and INTFLAG.SINGLEEE are cleared. If a dual bit error occurs while INTFLAG.SINGLEEE is set, the ERRADDR register is updated with the dual bit error information and INTFLAG.DUALE is also set.

The INTFLAG.SINGLEEE and INTFLAG.DUALE bits are both cleared on ERRADDR read.

The block diagram shows the ECC interface. When ECC is disabled (CTRLA.ECCDIS=1), the ECC field in RAM is left unchanged on writes. On reads, ECC errors are not corrected or flagged.

Related Links

[14.3. Block Diagram](#)

14.6.2 Interrupts

The RAMECC has the following interrupt sources:

- Dual Bit Error (DUALE): Indicates that a dual bit error has been detected.
- Single Bit Error (SINGLEEE): Indicates that a single bit error has been detected.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the ERRADDR register is read, the interrupt is disabled, or the RAMECC is reset.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

[14.8.3. INTFLAG](#)

14.7 Register Summary - RAMECC

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	7:0							DUALE	SINGLEE
0x01	INTENSET	7:0							DUALE	SINGLEE
0x02	INTFLAG	7:0							DUALE	SINGLEE
0x03	STATUS	7:0								ECCDIS
0x04	ERRADDR	7:0	ERRADDR[7:0]							
		15:8	ERRADDR[15:8]							
		23:16								ERRADDR[16]
		31:24								
0x08	Reserved									
...										
0x0E										
0x0F	DBGCTRL	7:0							ECCELOG	ECCDIS

14.8 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description (see *Register Access Protection* from Related Links).

Related Links

[14.5.7. Register Access Protection](#)

14.8.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
							DUALE	SINGLEEE
Access							R/W	R/W
Reset							0	0

Bit 1 – DUALE Dual Bit Error Interrupt Enable Clear

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Dual Bit Error Interrupt Enable bit, which disables the Dual Bit Error interrupt.

Value	Description
0	The Dual Bit Error interrupt is disabled.
1	The Dual Bit Error interrupt is enabled.

Bit 0 – SINGLEEE Single Bit Error Interrupt Enable Clear

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Single Bit Error Interrupt Enable bit, which disables the Single Bit Error interrupt.

Value	Description
0	The Single Bit Error interrupt is disabled.
1	The Single Bit Error interrupt is enabled.

14.8.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x01
Reset: 0x00
Property: Write-Protected

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
							DUALE	SINGLEE
Access							R/W	R/W
Reset							0	0

Bit 1 – DUALE Dual Bit Error Interrupt Enable Set

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Dual Bit Error Interrupt Enable bit, which enables the Dual Bit Error interrupt.

Value	Description
0	The Dual Bit Error interrupt is disabled.
1	The Dual Bit Error interrupt is enabled.

Bit 0 – SINGLEE Single Bit Error Interrupt Enable Set

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Single Bit Error Interrupt Enable bit, which disables the Single Bit Error interrupt.

Value	Description
0	The Single Bit Error interrupt is disabled.
1	The Single Bit Error interrupt is enabled.

14.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x02
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							DUALE	SINGLEEE
Access							R/W	R/W
Reset							0	0

Bit 1 – DUALE Dual Bit ECC Error Interrupt

This flag is set on the occurrence of a dual bit ECC error.

Writing a '0' to this bit has no effect.

Reading the ECCADDR register will clear the Dual Bit Error interrupt flag.

Value	Description
0	No dual bit errors have been received since the last clear.
1	At least one dual bit error has occurred since the last clear.

Bit 0 – SINGLEEE Single Bit ECC Error Interrupt

This flag is set on the occurrence of a single bit ECC error.

Writing a '0' to this bit has no effect.

Reading the ECCADDR register will clear the Single Bit Error interrupt flag.

Value	Description
0	No errors have been received since the last clear.
1	At least one single bit error has occurred since the last clear.

14.8.4 Status

Name: STATUS
Offset: 0x03
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
								ECCDIS
Access								R
Reset								0

Bit 0 – ECCDIS ECC Disable

This bit is fuse updated at startup. When enabled, the calculated ECC is written to RAM along with data. ECC correction and detection is enabled for reads.

Value	Description
0	ECC detection and correction is enabled.
1	ECC detection and correction is disabled.

14.8.5 Error Address

Name: ERRADDR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								ERRADDR[16]
Access								R
Reset								0
Bit	15	14	13	12	11	10	9	8
	ERRADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ERRADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 16:0 – ERRADDR[16:0] ECC Error Address

The RAM address offset from RAM start that caused an ECC error. If a single bit error is followed by a dual bit error, this register will be updated with the address of the dual bit error, otherwise it stalls on the first error occurrence. This register will read as zero unless INTFLAG.SINGLEEE and/or INTFLAG.DUALE are 1.

14.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ECCELOG	ECCDIS
Access							R/W	R/W
Reset							0	0

Bit 1 – ECCELOG ECC Error Log

When DBGCTRL.ECCDIS=0, This bit controls whether ECC errors are logged in the INTFLAG register.
When DBGCTRL.ECCDIS=1, this bit has no meaning.

Value	Description
0	ECC errors for debugger reads are not logged.
1	ECC errors for debugger reads are logged if DBGCTRL.ECCDIS=0.

Bit 0 – ECCDIS ECC Disable

By default, ECC errors during debugger reads are corrected and logged based on DBGCTRL.ECCELOG. Setting this bit will disable ECC correction and logging.

Value	Description
0	ECC errors are corrected for debugger reads and logged based on DBGCTRL.ECCELOG.
1	ECC errors are masked for debugger reads.

15. Power Management Unit (PMU)

15.1 Overview

This chapter describes the Power Management Unit (PMU) in the PIC32CX-BZ2 wireless microcontroller with the various modes it provides. It is for information purposes only.

Note: The PMU is a complex power controller that requires specific configuration and handling by the software for correct, safe operation of the SOC. The software SDK and operational stacks provided by Microchip handles the start-up and operation of the PMU. Therefore, Microchip highly recommends using this software framework for all application development on the device.

15.2 Power Modes

To minimize power consumption, the PIC32CX-BZ2 incorporates a PMU that provides both DC-DC (Buck) and LDO power conversion. The input voltage range to the PMU is 1.9V to 3.6V – VDD (main). On power-up, the PIC32CX-BZ2 operates in LDO mode; the software must enable the DC-DC Buck converter.

See *Electrical Characteristics* from Related Links for detailed electrical information.

The power modes and power management features provided by the PMU are shown in the following table.

Table 15-1. Power Modes and Power Management Features

Functions	Device Power Modes					
	Run	Idle	Dream ¹	Sleep(Standby)	Deep Sleep(Backup)	eXtreme Deep Sleep(Off)
CPU	On	Clock Gated	Clock Gated	Clock Gated	Off	Off
Peripherals	On	On	On Demand	Clock Gated	Off	Off
Flash	On	On	On Demand	Clock Gated	Off	Off
Core System Memory	On	On	On Demand	Retention Mode	Off	Off
Radio	On	On	On Demand	Clock Gated	Off	Off
DSWDT	On	On	On	On	On	Off
RTCC	On	On	On	On	On	Off
Backup RAM	On	On	On Demand	Retention Mode	Retention Mode	Off
XTAL(16 MHz)	On	On	On	On	Off	Off
SPLL	On	On	On Demand	Off	Off	Off
ADC	On	On	On Demand	On Demand	Off	Off
Analog Comp	On	On	On Demand	On Demand	Off	Off
FRC	On	On	On Demand	Off	Off	Off
LPRC	On	On	On	On	On	On

Note:

1. Dream (Sleep Walking) is not a mode by itself but is a companion mode to the Sleep mode. This mode cannot be entered directly through a software command.

Current consumption can be reduced on peripherals/modules in "On" state by clock gating and/or disabling the module, see *Module Enable/Disable Controls* in the *Register Description* from Related Links. All transitions from the Run state to any of the low power states is simply initiated by WFI command from the CPU. However, prior to initiating the WFI command.

The software performs the following actions:

- Disabling all interrupts
- Setting up the DSCON[DSEN], OSCCON[SLPEN], OSCCON[DRMEN] and Wireless Subsystem Sleep mode Controls

- Set the Wireless Subsystem into the Low Power mode
- Enable the appropriate wake-up events
- Checking for any pending interrupts and, if present, abort deep sleep and service the interrupt
- If no pending interrupts, then issue a SLEEP/WAIT command from the CPU to get into the appropriate Low Power mode

In the Run mode, the CPU is actively executing code. Run mode provides normal operation of the processor and all peripherals that are currently enabled.

In the Idle mode, all active peripherals can be clocked, but the CPU core is clock gated off and no code is executed.

In the Dream mode (or Sleep Walking mode), the CPU is clock gated but peripherals can be turned on on-demand by events related to those peripherals. No code is executed.

In the Sleep and Deep Sleep modes, the backup RAM is in retention mode, while the CPU and most peripherals are clock gated off and no code is executed.

In the eXtreme Deep Sleep mode, only the Low Power RC oscillator is enabled. Exiting Deep sleep/XDS is equivalent to Power-on Reset. The RCON register provides the status whether it is a normal power up or exiting from deep sleep/XDS.

Note: All desired register/configuration inputs (for example, DSZPBOREN, DSWDTPS) to be preserved must be set to expected values before setting DSCON.DSEN = 1; failure to do so would result in faulty configuration fault indications.

The low power mode entry and exit commands and wake up resources are shown in the following table.

Table 15-2. Low Power Mode Entry and Exit commands and Wake-up Resources

Device Power Modes	Entry Commands	Wake-up Resources
Run	—	—
Idle	WFI Instruction + ~OSCCON[SLPEN]	IRQ, Reset, Others ⁽¹⁾
Dream	OSCCON[DREAM] + Event + Sleep mode	IRQ, Reset, Others
Sleep	~DSCON[DSEN]+OSCCON[SLPEN]+Wireless Sleep followed by WFI	IRQ, Reset, Others
Deep Sleep	DSCON[DSEN]+ {RTCC (Enabled) or DSWDT (Enabled)} + Wireless Sleep followed by WFI	INT0, RTCC, DSWDT ⁽²⁾ , Reset
eXtreme Deep Sleep	DSCON[DSEN]+ {RTCC (Disabled) and DSWDT (Disabled)} + Wireless Sleep followed by WFI	INT0, Reset

Notes:

1. Others = System Wake-up (Dream), WDT (Timeout Event), DMT (Timeout Event), PLVD Event, PMU Event, External NMI/INT, DSU/ICD Debug Event, CPU Debug Event.
2. To enable the Deep Sleep Mode Watchdog Timer(DSWDT) in deep sleep mode, Configuration Control Register 4(CFGCON4). DSWDT is a separate timer from the device's watchdog timer that is used in run mode. The device Watchdog Timer (WDT) does not have to be enabled for the DSWDT to function.
3. For more information, see [DSCON](#) register from Related Links.

Related Links

[15.4. DSCON](#)

[20.4.2. Register Description](#)

[43. Electrical Characteristics](#)

15.3 PMU Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DSCON	7:0							ZPBOR	DSSR
		15:8	DSEN		XSEMAEN	RTCMD				RTCCWDIS
		23:16								
		31:24								

15.4 Deep Sleep Control Register

Name: DSCON
Offset: 0x00
Reset: 0x00
Property: -

Note:

1. All register bits are reset only in the case of a V_{DDBAT} POR event.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access	DSEN		XSEMAEN	RTCMD				RTCCWDIS
Reset	R/W/HC		R/W	R/W				R/W
Reset	0		0	0				0
Bit	7	6	5	4	3	2	1	0
Access							ZPBOR	DSSR
Reset							R/W/C/HS	R/C/HS/HC
Reset							0	0

Bit 15 – DSEN Deep Sleep Enable Bit

This bit is only writable when `cfg_deep_sleep_en = 1`.

Value	Description
1	Deep Sleep mode is entered on a SLEEP/WAIT command
0	Sleep mode is entered on a SLEEP/WAIT command

Bit 13 – XSEMAEN Extended Semaphore Enable Bit

Value	Description
1	Extended semaphores retention is enabled in Deep Sleep mode
0	Indeterminate extended semaphore retention in Deep Sleep

Bit 12 – RTCMD RTCC Module Disable Bit

Value	Description
1	RTCC is not enabled
0	RTCC is enabled

Bit 8 – RTCCWDIS RTCC Wake-up Disable Bit

Value	Description
1	Wake-up from RTCC is disabled
0	Wake-up from RTCC is enabled

Bit 1 – ZPBOR Zero-Power BOR Event Bit

Note:

Unlike all other events, a Zero-Power BOR event will not cause a wake-up from Deep Sleep. This bit is present only as a status bit.

Value	Description
1	The ZPBOR was active and a BOR event was detected during Deep Sleep
0	The ZPBOR was not active or was active, but did not detect a BOR event during Deep Sleep

Bit 0 – DSSR Deep Sleep State Restored Bit

Clearing this bit will cause the `xds_keepctrl_en_lv` output to be negated, indicating it is safe to release all Deep Sleep configuration keeper cells. If the wake-up source was something other than an MCLR or ICD Reset, the `xds_keepio_en_lv` output will also be negated at the same time, indicating it is safe to release all I/O keeper cells. This bit must be cleared by software.

16. Watchdog Timer (WDT)

16.1 Overview

The Watchdog Timer (WDT) can be used to detect system software malfunctions by resetting the device if the WDT is not cleared periodically in software. The user can configure the WDT in Windowed mode or non-Windowed mode. Various WDT time-out periods can be selected using the WDT postscaler. The WDT can also be used to wake the device from Sleep or Idle mode.

16.2 Features

- Configuration using Fuses (DEVCFG) or Software controlled
- Up to 32 Configurable Time-Out Periods
- Can wake the Device from Standby Sleep or Idle mode
- Independent Run and Standby Sleep mode Counters
- WDT may use alternate Clock source and Postscaler for Run mode Counter
- Independent 5-bit Postscalers for Run and Standby Sleep mode Counters
- Hardware and Software enabled
- Two Clock sources
- Windowed WDT

Note: When the CPU is running on the same clock or clock frequency as the WDT, the lowest pre-scale values may not allow the CPU to have enough time to reset the WDT before it expires.

16.3 Applications

The WDT is a free-running timer with a configurable postscaler. The counter is clocked with an external reference clock until the counter value exceeds the selected WDT period. If enabled, the WDT will continue to operate even if the main processor clock (for example, the crystal oscillator) fails.

The WDT uses separate internal counters for use in Run mode and Sleep/Idle modes. One counter operates only in Run mode; the count value of this counter is frozen when the device is in Sleep or Idle modes.

The second counter operates only in Sleep and Idle modes; it is reset when entering Sleep or Idle. This provides the following benefits:

- A different WDT clock source can be used in Run mode(PB1_CLK or LPRC) vs. Sleep/Idle modes (LPRC only).
- A different post-scale value may be used in Run mode vs. Sleep/Idle modes.
- The Run mode WDT count is preserved while in Sleep or Idle modes, which makes it easier to manage the WDT while in windowed mode.

The WDT can have two modes of operation, Windowed and non-Windowed. In Windowed mode, software can clear the WDT only when the counter is in its final window before a period match occurs. There are four window size options (25%, 37.5%, 50% and 75% of the total WDT period). This window is active when the timer counter is greater than a predetermined value for each option. The window size is determined by the WDTWINSZ configuration fuses (see *System Configuration Registers (CFG)* from Related Links). Any attempts to clear the WDT when the window is not active would cause a device Reset. In non-Windowed configuration, software can clear the WDT any time before the period match occurs.

Note: Windowed WDT operation is not applicable in Sleep or Idle modes.

Related Links

[18. System Configuration and Register Locking \(CFG\)](#)

16.3.1 Enabling the WDT

The WDT can be enabled through hardware or software control. The WDT is enabled if WDTEN=1 in the DEVCFG2 fuse register. If WDTEN=0, the WDT is disabled, and can be controlled using the ON bit (WDTCON[15]).

16.3.2 Selecting WDT Clock Source

The Sleep Mode Counter always uses the 32 kHz LPRC clock source. The Run Mode Counter will use the clock source selected by the WDTRMCS bit in fuses Configuration Bits register DEVCFG2 to be either the 32 kHz LPRC clock or the PB1_CLK bus clock (sysclk). See *WDTRMCS[1:0]* bits in the *CFGCON2(L)* register from Related Links.

Related Links

[18.4.3. CFGCON2\(L\)](#)

16.3.3 Clearing the Watch Dog Timer

To clear the WDT, software must write to the WDTKEY register with the value 16'h5743 before the timer expires. The upper two bytes of WDTCON must be written simultaneously, as a 16-bit write. Any other write with a different value or byte size will NOT clear the timer.

16.3.4 Selecting WDT Period

The WDT can be configured for various time-out periods by controlling the WDT postscaler value. The settings are chosen using the WDTPSR[4:0] inputs for the Run Mode Counter and the WDTPSS[4:0] inputs for the Sleep Mode Counter. These values range from 1 to 2²⁰ and allow for time-out periods of from 1 ms to over 17 minutes when using the 21 kHz LPRC clock.

16.3.5 Events that Reset both WDT Counters

The following events will reset both internal WDT counters:

- A generic device reset
- Disabling the WDT via the ON bit

16.3.6 Events that Reset only the Run Mode Counter

The following events reset the Run Mode Counter:

- Any counter value greater than the selected WDT period
- Detecting a correct write value to WDTCON.WDTKEY within the correct time window

16.3.7 Events that Reset only the Sleep Mode Counter

The following events reset the Sleep Mode Counter:

- Entry into Sleep or Idle modes
- Any counter value greater than the selected WDT period

16.3.8 Run Mode WDT Event

Once the WDT period is exceeded, or if a correct write value to the WDTCON.WDTKEY is executed in a Windowed mode when the window is not yet active, the WDT event is activated. The event remains active for one clock period.

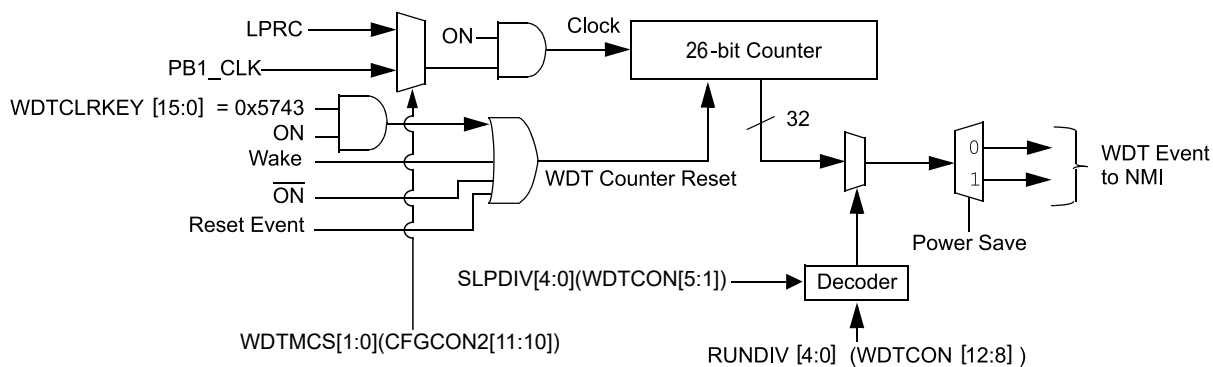
The WDT is reset after the count is exceeded and continues to run.

16.3.9 Sleep Mode WDT Event

Once the WDT period is exceeded, the WDT event is activated. The event remains active for one clock period. The WDT is reset after the count is exceeded.

16.4 Block Diagram

Figure 16-1. Watchdog Timer Block Diagram



16.5 Configuration

The WDT is configured using the following config register bits/fields:

- Window size (CFGCON2.FWINSZ[1:0])
- Windowing disable (CFGCON2.WINDIS)
- Post-scaler selection (CFGCON2.WDTPS[4:0])
- Run Mode Counter clock source (CFGCON2.WDTRMCS[1:0])

16.6 Register Summary - WDT

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0		
0x00	WDTCON	7:0					SLPDIV[4:0]				WDTWINEN	
		15:8	ON					RUNDIV[4:0]				
		23:16					WDTCLRKEY[7:0]					
		31:24					WDTCLRKEY[15:8]					

16.7 Register Description

The following are the list of conventions available in the register description:

- – R = Readable bit
- – W = Writable bit
- – U = Unimplemented bit, read as '0'
- – -n = Value at POR
- – 1 = Bit is set
- – 0 = Bit is cleared
- – x = Bit is unknown
- y = Values set from Configuration bits on POR
- Reset values are shown in hexadecimal.

Related Links

[6.1.9. CLR, SET and INV Registers](#)

16.7.1 WDTCON - Watchdog Timer Control Register

Name: WDTCON
Offset: 0x00
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
	WDTCLRKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WDTCLRKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ON			RUNDIV[4:0]				
Access	R/W			R	R	R	R	R
Reset	y			y	y	y	y	y
Bit	7	6	5	4	3	2	1	0
			SLPDIV[4:0]					WDTWINEN
Access			R	R	R	R	R	R/W
Reset			y	y	y	y	y	0

Bits 31:16 – WDTCLRKEY[15:0] Watchdog Timer Clear Key bits

To clear the WDT to prevent a time-out, software must write the value 0x5743 to this location using a single 16-bit write. Anything other than a 16-bit write will not reset the WDT. You must use a 16-bit write for the WDTCLRKEY[15:0] bits.

Bit 15 – ON Watchdog Timer Enable bit

Note:

1. This bit only has control when the WDTEN bit (DEVCFG2/CFGCON2[23]) = 0.

Value	Description
1	WDT is enabled
0	WDT is disabled

Bits 12:8 – RUNDIV[4:0] Watchdog Timer Postscaler Run Counter Value bits

On Reset, these bits are set to the values of the WDTCSR[4:0] Configuration bits in CFGCON2.

Bits 5:1 – SLPDIV[4:0] Watchdog Timer Postscaler Sleep Counter Value bits

On Reset, these bits are set to the values of the WDTCSR[4:0] Configuration bits in CFGCON1.

Bit 0 – WDTWINEN Watchdog Timer Window Enable bit

Value	Description
1	Enable windowed WDT
0	Disable windowed WDT

17. Deadman Timer (DMT)

17.1 Overview

The Deadman Timer (DMT) module is designed to enable users to be able to monitor the health of their application software by requiring periodic timer interrupts within a user-specified timing window. The DMT module is a synchronous counter and when enabled, counts instruction fetches and causes a system reset if the DMT counter is not cleared within a set number of instructions. The DMT is typically connected to the system clock that drives the processor. The user specifies the timer time-out value and a mask value that specifies the range of the window, which is the range of counts that is not considered for the comparison event.

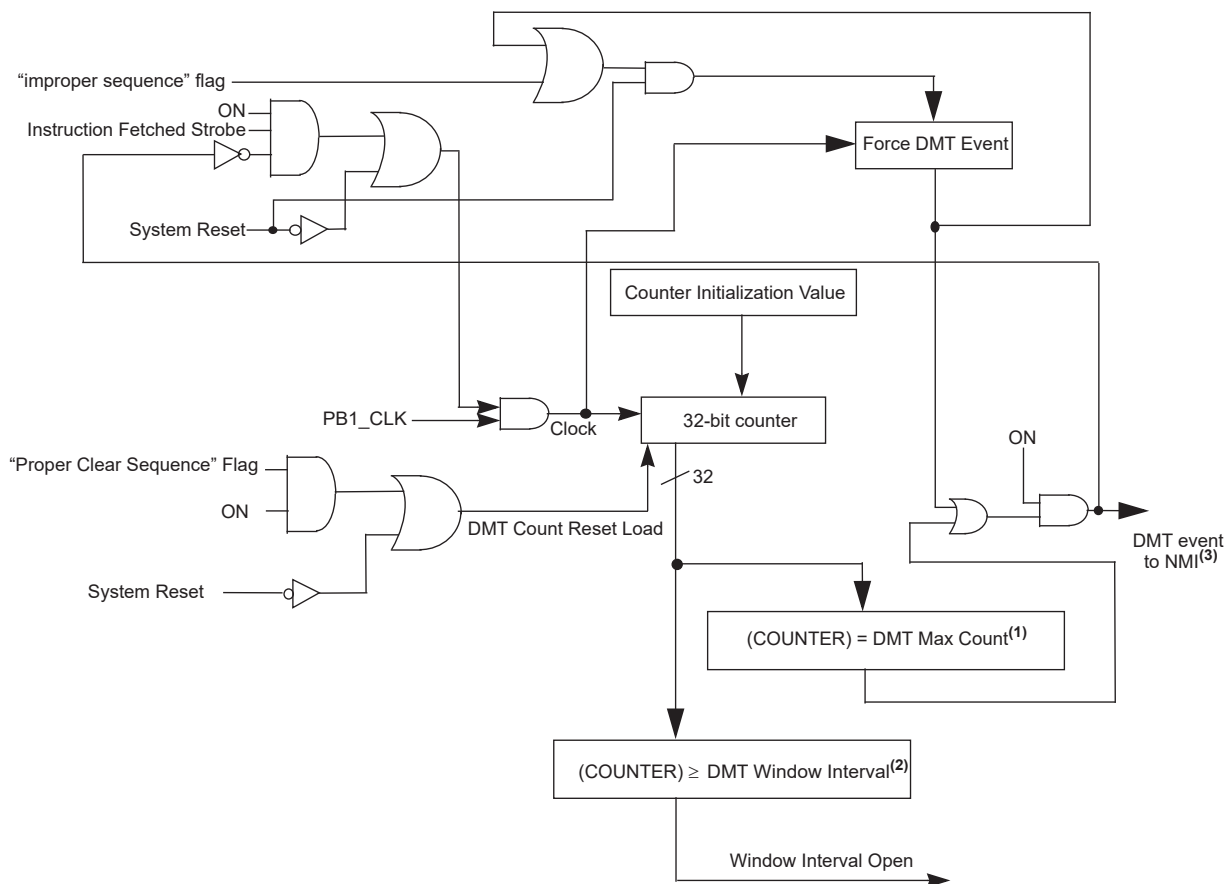
17.2 Features

- 32-bit Configurable Count-limit Based on Counting Instructions Fetched
- Hardware- or Software-enabled Control
- User-configurable Time-out Period or Instruction Count
- Two Instruction Sequence to Clear Timer
- 32-bit Configurable Window to Clear Timer

17.3 Block Diagram

The following figure shows the block diagram of the Deadman Timer.

Figure 17-1. Deadman Timer Block Diagram



Notes:

1. The DMT Max Count is controlled by the DMTCNT[4:0] bits in the CFGCON2 register “Maximum = 2^{31} ”.
2. The DMT Window Interval is controlled by the DMTINTV[2:0] bits in the CFGCON2 register.
3. For more details, see *Resets* from Related Links.

Related Links

[13.15. Resets](#)

17.4 DMT Operation

17.4.1 Mode of Operation

The primary function of the Deadman Timer (DMT) module is to reset the processor in the event of a software malfunction. The DMT module, which works on the system clock, is a free running instruction fetch timer, which is clocked whenever an instruction fetch occurs until a count match occurs. The instructions are not fetched when the processor is in the Standby Sleep mode.

The DMT module consists of a 32-bit counter, the read-only DMTCNT register with a time-out count match value as specified by the 32-bit DMT count configuration fuse bits CFGCON2.DMTCNT[4:0]. Whenever the count match occurs, a DMT reset event will occur and the DMTEVENT bit in DMTSTAT register will be set.

A DMT module is typically used in mission critical and safety critical applications, where any failure of the software functionality and sequencing must be detected.

17.4.2 Enabling and Disabling the DMT Module

Because of the nature of the Deadman Timer, the PMD register bit is not provided to enable/disable the module. The DMT module can be enabled or disabled by the DMT enable (DMTEN) bit in the Configuration Control Register 2 (CFGCON2) fuse register or it can be enabled through software by writing to the Deadman Timer Control (DMTCON) register. Once the DMT is enabled, it may not be disabled without a device reset.

If the DMTEN configuration bit in the CFGCON2 fuse register is set, the DMT is always enabled. The ON control bit (DMTCON[15]) in DMTCON register will reflect this by reading a ‘1’. In this mode, the ON bit (DMTCON[15]) cannot be cleared in software. To disable the DMT, the DMTEN configuration bit must be cleared in the CFGCON2 fuse register. When DMTEN is cleared to ‘0’ in the CFGCON2, the DMT is disabled in hardware.

Software can enable the DMT by setting the ON bit in the DMTCON register. However, for software control, the DMTEN configuration bit in the CFGCON2 fuse register must be set to ‘0’.

17.4.3 DMT Count Windowed Interval

The DMT module has the Windowed Operation mode. The DMT interval (DMTINV[2:0]) bits in the Configuration Control Register 2 (CFGCON2) fuse register sets the window interval value. The PSINTV[31:0] bits in DMT interval post status register (DMTPSINTV) allows the software to read the DMT window interval value. That means this register reads the value that is written to the DMT interval (DMTINV[2:0]) bits in the Configuration Control Register 2 (CFGCON2) fuse register.

In the Windowed mode, software can clear the DMT only when the counter is in its final window before a count match occurs. That is, if the DMT counter value is greater than or equal to the value written to the window interval value, only then the DMT clear sequence can be executed in the DMT module. If the DMT is cleared before the allowed window, a DMT reset event is immediately generated.

17.4.4 DMT Count Selection

The Deadman Timer count is set by the DMT count configuration (DMTCNT[4:0]) bits in the Configuration Control Register 2 (CFGCON2) fuse register. The current DMT count value can be obtained by reading the DMT count register DMTCNT.

The PSCNT[31:0] bits in the DMT count post status register (DMTPSCNT) allows the software to read the maximum count selected for the DMT. The PSCNTx bit values are the values that are initially written to the DMTCNTx bits in the CFGCON2 fuse register. Whenever the DMT event occurs, the user can always compare to see whether the current counter value in the DMTCNT register is equal to the value of the DMTPSCNT register, which holds the maximum count value.

Whenever the DMT current counter value in DMTCNT reaches the value of the DMTPSINTV register, the window interval opens permitting the user to execute the DMT clear sequence. The open window interval is indicated by the WINOPN bit in DMTSTAT register.

The UPRCNT[15:0] bits in the DMT hold register (DMTHOLDREG) holds the value of the last read DMT upper count values whenever DMTCNT is last read.

17.4.5 DMT Operation in Power-Saving Modes

As the DMT module is only incremented by instruction fetches, the count value will not change when the core is inactive. The DMT module remains inactive in the Standby Sleep and Idle modes. As soon as the device wakes-up from Standby Sleep or Idle, the DMT counter starts incrementing again for every instruction fetch.

17.4.6 Resetting the DMT

The DMT can be reset in two ways: one way is after a system reset and another way is by writing an ordered sequence to the DMT pre-clear register (DMTPRECLR) and DMT clear register (DMTCLR) in a specific two-step sequence.

Clearing the DMT counter value requires the following sequence of operations:

1. The STEP1[7:0] bits in the DMTPRECLR register must be written as '01000000' (0x40). This action sets the "enable for clearing" state, which enables the DMT to be cleared by step 2.
2. The STEP2[7:0] bits in the DMTCLR register must be written as '00001000' (0x08). This can only be done if preceded by step 1 and if the DMT is in the open window interval.

Once these values are written, following are cleared to zero:

- DMTCNT counter
- DMTPRECLR register
- DMTCLR register
- DMTSTAT register

If any value other than 0x40 is written to the STEP1x bits, the BAD1 bit in the DMTSTAT register will be set and it causes a DMT event to occur. Any value other than 0x08, written to the STEP2x bits, will cause the BAD2 bit to be set in the DMTSTAT register. Also, if step 2 is not preceded by step 1, or step 2 is not carried out in the open window interval, it causes the BAD2 flag to be set. Immediately, a DMT event will occur. In both these cases, the DMT EVENT bit in the DMTSTAT register will be set. Refer to the flowchart shown in the following figure.

Figure 17-2. Flowchart for Clearing the DMT



17.5 Register Summary

See the *DMT* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DMTCON	7:0								
		15:8	ON							
		23:16								
		31:24								
0x04 ... 0x0F	Reserved									
0x10	DMTPRECLR	7:0								
		15:8	STEP1[7:0]							
		23:16								
		31:24								
0x14 ... 0x1F	Reserved									
0x20	DMTCLR	7:0	STEP2[7:0]							
		15:8								
		23:16								
		31:24								
0x24 ... 0x2F	Reserved									
0x30	DMTSTAT	7:0	BAD1	BAD2	DMT_EVENT					WINOPN
		15:8								
		23:16								
		31:24								
0x34 ... 0x3F	Reserved									
0x40	DMTCNT	7:0	COUNTER[7:0]							
		15:8	COUNTER[15:8]							
		23:16	COUNTER[23:16]							
		31:24	COUNTER[31:24]							
0x44 ... 0x4F	Reserved									
0x50	DMTHOLDREG	7:0	UPRCNT[7:0]							
		15:8	UPRCNT[15:8]							
		23:16								
		31:24								
0x54 ... 0x5F	Reserved									
0x60	DMTPSCNT	7:0	PSCNT[7:0]							
		15:8	PSCNT[15:8]							
		23:16	PSCNT[23:16]							
		31:24	PSCNT[31:24]							
0x64 ... 0x6F	Reserved									
0x70	DMTPSINTV	7:0	PSINTV[7:0]							
		15:8	PSINTV[15:8]							
		23:16	PSINTV[23:16]							
		31:24	PSINTV[31:24]							

17.6 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the “Write-Synchronized” or the “Read-Synchronized” property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

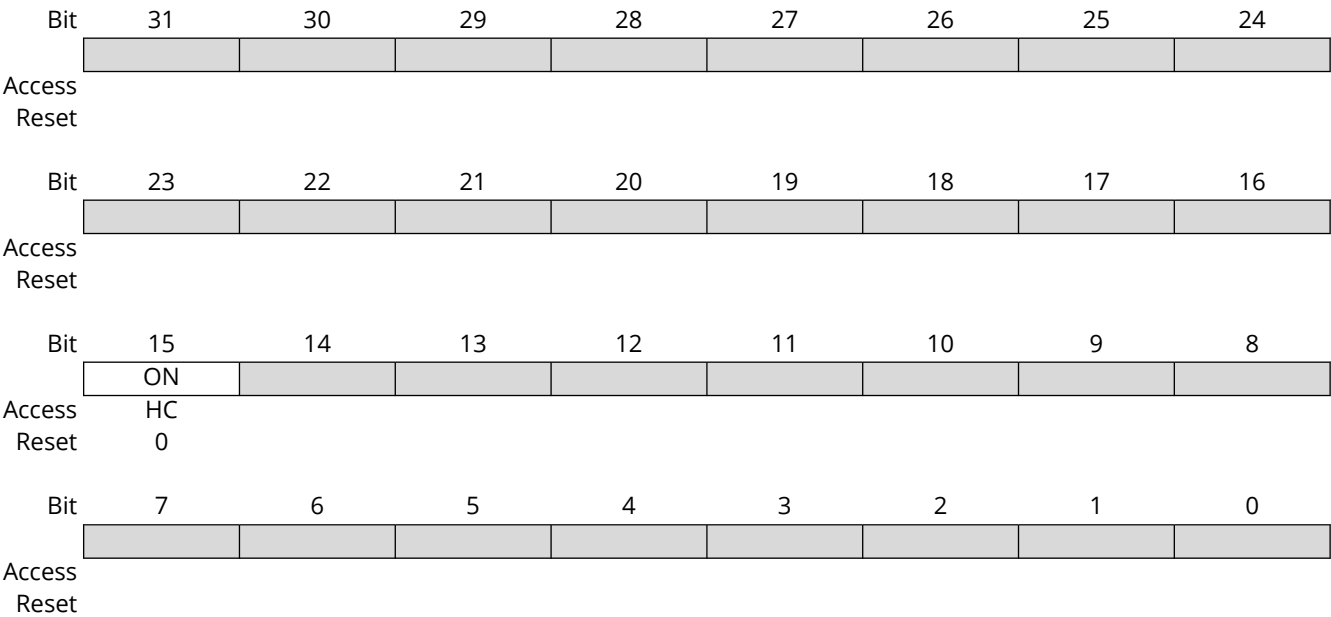
Note: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links

Related Links

[6.1.9. CLR, SET and INV Registers](#)

17.6.1 Deadman Timer Control

Name: DMTCON
Offset: 0x00
Reset: 0x00
Property: -



Bit 15 – ON On bit
The ON bit reflects the status of the configuration fuse CFGCON2.DMTEN, if the fuse is set.

Value	Description
1	Enables the Deadman Timer if the event configuration fuse is not enabled.
0	The DMT disabled.

17.6.2 Deadman Timer Preclear

Name: DMTPRECLR
Offset: 0x10
Reset: 0x00
Property: -

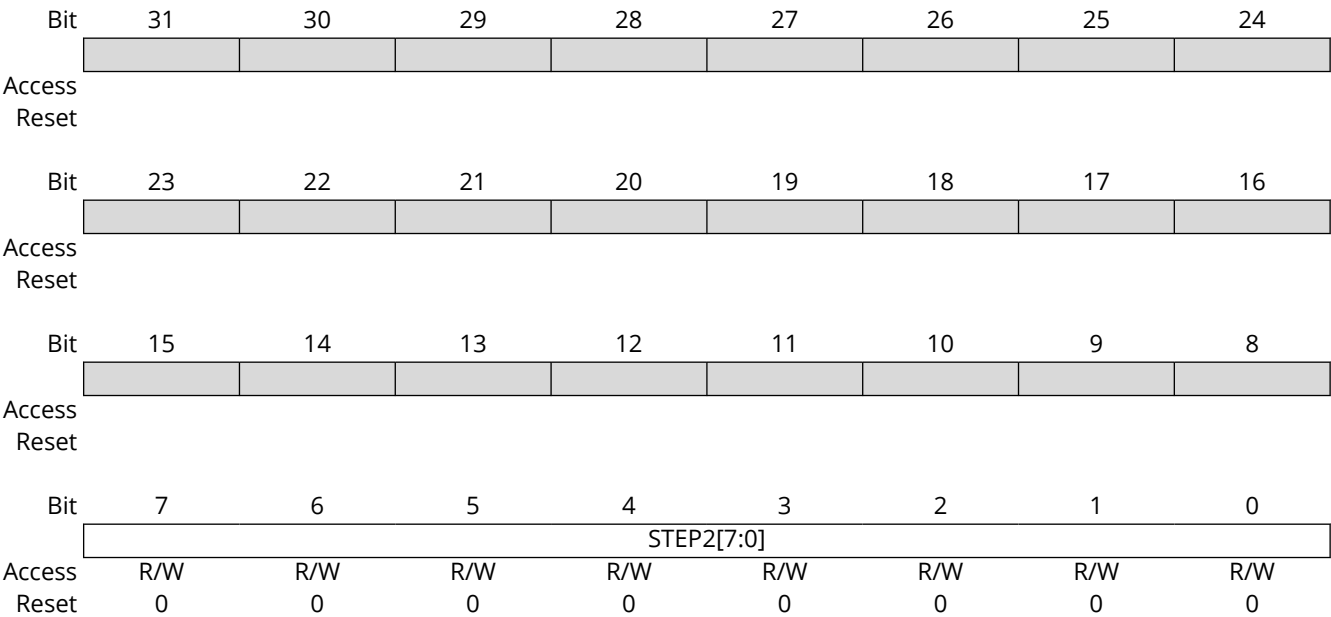
Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	STEP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 15:8 – STEP1[7:0] Pre-Clear Enable bit
This bit pre-clears the DMT when Write Pattern is set as following:

Value	Description
01000000	Enables the Deadman Timer Pre-Clear (STEP 1).
All other write patterns	Sets DMTSTAT.BAD1 flag to '1'. Note: Bits 15:8 are cleared when a DMT Reset event occurs. STEP1 is also cleared if DMTCLR.STEP2 is loaded with the correct value in the correct sequence.

17.6.3 Deadman Timer Clear

Name: DMTCLR
Offset: 0x20
Reset: 0x00
Property: -



Bits 7:0 – STEP2[7:0] Clear Timer bit

This bit clears the DMT when Write Pattern is set as following:

Value	Description
00001000	Clears DMTPRECLR.STEP1, DMTCLR.STEP2 and the Dead Man Timer if and only if preceded by the correct loading of Pre-Clear Enable (STEP1) in the correct sequence. The write to the DMTCLR.STEP2 field may be verified by reading DMTCNT and observing the counter being Reset.
all other write patterns	The DMTSTAT.BAD2 flag is set to '1', the value in the DMTPRECLR.STEP1 remains unchanged and the new value being written to DMTCLR.STEP2 is captured. Note: These bits 7:0 are also cleared when a DMT Reset event occurs.

17.6.4 Deadman Timer Status

Name: DMTSTAT
Offset: 0x30
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BAD1	BAD2	DMT_EVENT					WINOPN
Access	R	R	R					R
Reset	0	0	0					0

Bit 7 – BAD1

When an incorrect DMTPRECLR.STEP1 value is detected, this bit is set. It is cleared by a Reset.

Bit 6 – BAD2

When an incorrect value of DMTCLR.STEP2 is detected, this bit is set. It is cleared by a Reset.

Bit 5 – DMT_EVENT

This bit is set when the Deadman timer event is detected (counter expired or bad STEP1[7:0] or STEP2[7:0] value is entered prior to the counter increment). This bit remains set and is cleared only by a Reset.

Bit 0 – WINOPN Deadman Timer Clear Window bit.

A value of '1' indicates that a STEP2 “clear” action can take place, and if this “clearing” action occurs as part of a correct sequence of actions, the DMT counter will be cleared.

17.6.5 Deadman Timer Count

Name: DMTCNT
Offset: 0x40
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
	COUNTER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COUNTER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COUNTER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNTER[31:0] Read Current Contents of DMT Counter
This bit reads the available contents of DMT counter.

17.6.6 Deadman Timer Count Holding Register

Name: DMTHOLDREG
Offset: 0x50
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	UPRCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	UPRCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – UPRCNT[15:0]

It is the content of DMTCNT.COUNTER[31:16] when the counter was last read to ensure a synchronous snapshot of the counter. This register is initialized to '0' on reset and is only loaded when the DMTCNT register is read.

17.6.7 Post Status Configure DMT Count Status Register

Name: DMTPSCNT
Offset: 0x60
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
	PSCNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PSCNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PSCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PSCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PSCNT[31:0] DMT Instruction Count Value Configuration Fuse Status bits
This bit always reflects the value of CFGCON2.DMTCNT.

17.6.8 Post Status Configure DMT Interval Status Register

Name: DMTPSINTV
Offset: 0x70
Reset: 0x00
Property: -

Bit	31	30	29	28	27	26	25	24
	PSINTV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PSINTV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PSINTV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PSINTV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PSINTV[31:0] DMT Window Interval Configuration Status bits.
This bit reflects the value of CFGCON2.DMTINTV.

18. System Configuration and Register Locking (CFG)

18.1 Overview

This device provides several user writable configuration registers related to the configuration and operation of the system. The registers marked with (L) are loadable from Flash via their corresponding FBCFG* registers in the following table. The user must program these FBCFG* registers, which, then, loads the appropriate register after Reset.

This device provides a single user writable configuration register related to boot configuration of the device. The BCFG0 register provides control, selection and locking for various features of the device, including Flash BCFG7-0 valid, Flash Signature Bit (read only), Code Protect Status (read only). BCFG0 is a read-only register loaded from Flash register FBCFG0 in the following table.

Table 18-1. Writable Configuration Registers

Register	Address	Destination
FBCFG0	0x0004_5F9C	BCFG0 (0x4400_0200)
FBCFG1/DEVCFG0	0x0004_5F98	CFGCON0(L) (0x4400_0000)
FBCFG2/DEVCFG1	0x0004_5F94	CFGCON1(L) (0x4400_0010)
FBCFG3/DEVCFG2	0x0004_5F90	CFGCON2(L) (0x4400_0020)
FBCFG4/DEVCFG4	0x0004_5F8C	CFGCON4(L) (0x4400_0040)
FBCFG5/FUSERID	0x0004_5F88	USERID(L) (0x4400_00A0)

CFGCON0(L) – Provides control, selection and locking for various features of the device.

- PPS register locking
- PMD register locking
- CFGPG register locking
- Config register locking
- JTAG port enable and configuration
- Trace port enable
- Flash ECC control

CFGCON1(L) – Provides control, selection and locking for various features of the device.

- Debug port and feature configuration CFGCON0 locking control
- Class B functionality enable

CFGCON2(L) – Provides control, selection and locking for various features of the device.

- Deadman timer enable and configuration
- Watchdog timer enable and configuration
- Clock monitoring and control
- Oscillator enable and configuration
- 2-Speed start-up enabled in the Sleep mode bit

CFGCON4(L) – Provides control, selection and locking for various features of the device.

- Deep sleep modules control
- SOSC configuration control

CFGPGQOS – The CFGPGQOS register defines the permission group settings for various bus hosts on the device bus matrix.

USER_ID(L) – The USER_ID register is used to provide the end user with a 16-bit ID field that may be read out directly through the JTAG interface via the USER_ID JTAG instruction.

BCFG0 – Used to set Code Protect, Sign Bit and control PCHE cache mode.

18.2 Applications

18.2.1 Loading User/Device Configuration Registers

The non-BCFG system configuration registers (CFGCON* and CFGCON(L)*) are available in a memory mapped area under software-based locking control.

The following registers that do not have permanent storage:

- Wi-Fi (RF, BBP, PHY) System Configurations
- Calibration values, System PLL
- CRU clock switching
- Analog calibration values

For these registers, the firmware (boot code and/or application code) must allocate nonvolatile storage of system configuration values and load them into the memory mapped system configuration registers during a device boot. The recommended non-volatile storage space in NVR boot memory is shown in the Flash Memory Organization table. See *Flash Memory Organization* from Related Links.

In general, if the Reset value or Flash-loaded value of a system configuration register is the value that is required, then it is not necessary to load the system configuration register during a device boot.

Related Links

[24.5.2. Flash Memory Organization](#)

18.2.2 Locking and Unlocking the System Configuration Registers

Write access to the system configuration registers is controlled via the CFGCON0.CFGLOCK[1:0] register bits.

18.2.3 NMI Events

The only system configuration that gets reset on an NMI event are CPUPG bits in the CFGPGQOS register. This allows application firmware to pass control back to the bootloader and re-enable reads of all configuration words from Boot Flash NVR pages if reads of the Boot Flash pages were disabled using group permissions.

18.2.4 Alternate System Configuration

To provide better data retention for the configuration data (compared to the data retention of the rest of the device Flash), each set of DEVCFGx, FBCFG0 registers are duplicated in an alternate set. This improves the chances of correct system configuration out of Reset. In the event of an uncorrectable error in a word (ECC DED), the Flash controller uses the alternate set to obtain the correct data.

Table 18-2. Alternate Register

Register	Alternate Register
FBCFG0	ALTFCFG0(0004_5E9C)
DEVCFG0	ALTDEVCFG0(0004_5E98)
DEVCFG1	ALTDEVCFG1(0004_5E94)
DEVCFG2	ALTDEVCFG2(0004_5E90)
DEVCFG4	ALTDEVCFG4(0004_5E8C)

.....continued

Register	Alternate Register
FUSERID	ALTFUSERID(0004_5E88)

18.3 CFG Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CFGCON0(L)	7:0	CPENFILT	ACCOMP1_ALT EN	GPSOSCE*	ADCOPVR	JTAGEN	TROEN	SWOEN	TDOEN
		15:8	CFGLOCK[1:0]		IOLOCK	PMDLOCK	PGLOCK	PMULOCK	RTCOUT_ALTE N	RTCINO_ALTE N
		23:16	SLRTEN2	SLRTEN1	SLRTEN0	HPLUGDIS	SMBUSEN2	SMBUSEN1	SMBUSEN0	VBCMODE
		31:24		FRECCDIS	FECCCON/ECCCTL[1:0]		ADCFEN	INT0P	INT0E	PCM
0x04 ... 0x0F	Reserved									
0x10	CFGCON1(L)	7:0	ZBTWKSYS		TRCEN	ICESEL[1:0]			DEBUG[1:0]	
		15:8	QSCHE_EN	SMCLR	SLRCTRL2	SLRCTRL1	SLRCTRL0	CLASSBDIS	CMP1_OE	CMP0_OE
		23:16	I2CDSEL2	I2CDSEL1	I2CDSEL0	CCL_OE	SCOM2_HSEN	SCOM1_HSEN	SCOM0_HSEN	QSPI_HSEN
		31:24		CLKZBREF	QSPIDDRM	WDTPSS[4:0]				
0x14 ... 0x1F	Reserved									
0x20	CFGCON2(L)	7:0	PMUTEST_VD D_EN		DMTINTV[2:0]			ACMP_CYCLE[2:0]		
		15:8	FSCMEN	CKSWEN	WAKE2SPD	SOSCSEL	WDTRMCS[1:0]		POSCMD[1:0]	
		23:16	WDTEN	WINDIS	WDTSPGM	WDTPSR[4:0]				
		31:24	DMTEN	DMTCNT[4:0]				WINSZ[1:0]		
0x24 ... 0x3F	Reserved									
0x40	CFGCON4(L)	7:0	SOSC_CFG[7:0]							
		15:8	MLPCLK_MO D	VBKP_DIVSEL	VBKP_32KCSEL[1:0]		VBKP_1KCSEL	RTCEVENT_E N	RTCEVENTSEL[1:0]	
		23:16	DSWDTPS[2:0]			DSZPBOREN	CPEN_DLY[2:0]			RTCEVTYPE
		31:24	RTCNTM_CSE L	LPOSCEN	UVREGROVR	DSBITEN	DSWDTEN	DSWDTLPRC	DSWDTPS[4:3]	
0x44 ... 0x4F	Reserved									
0x50	CFGPGQOS	7:0	ICDJQOS[1:0]		ICDJP[1:0]		CPUQOS[1:0]		CPUPG[1:0]	
		15:8							DMAPG[1:0]	
		23:16	ICMQOS[1:0]		ICMPG[1:0]		ADCQOS[1:0]		ADCPG[1:0]	
		31:24	WISIBQOS[1:0]		FCQOS[1:0]				DSUPG[1:0]	
0x54 ... 0x5F	Reserved									
0x60	CFGPCLKGEN1	7:0	RCD	FREQMRCSEL[2:0]			EICCD	EICCSEL[2:0]		
		15:8	S01CD	SERCOM01CSEL[2:0]			MCD	FREQMMMCSEL[2:0]		
		23:16	TCC12CD	TCC12CSEL[2:0]			S23D	SERCOM23CSEL[2:0]		
		31:24	CM4TD	CM4TCSEL[2:0]			TC23CD	TC23CSEL[2:0]		
0x64 ... 0x6F	Reserved									
0x70	CFGPCLKGEN2	7:0	C2D	EVSYS2SEL[2:0]			C1D	EVSYS1SEL[2:0]		
		15:8	C4D	EVSYS4SEL[2:0]			C3D	EVSYS3SEL[2:0]		
		23:16	C6D	EVSYS6SEL[2:0]			C5D	EVSYS5SEL[2:0]		
		31:24	C8D	EVSYS8SEL[2:0]			C7D	EVSYS7SEL[2:0]		
0x74 ... 0x7F	Reserved									
0x80	CFGPCLKGEN3	7:0	C10D	EVSYS10SEL[2:0]			C9D	EVSYS9SEL[2:0]		
		15:8	C12D	EVSYS12SEL[2:0]			C11D	EVSYS11SEL[2:0]		
		23:16	TCC0CD	TCC0CSEL[2:0]			ACCD	ACCLKSEL[2:0]		
		31:24	TC1CD	TC1CSEL[2:0]			TC0CD	TC0CSEL[2:0]		

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x84 ... 0x9F	Reserved									
0xA0	USER_ID	7:0	USER_ID[7:0]							
		15:8	USER_ID[15:8]							
		23:16								
		31:24								
0xA4 ... 0x01FF	Reserved									
0x0200	BCFG0	7:0							PCSCMODE	
		15:8								
		23:16								
		31:24	BINFOVALID0			SIGN	CP			

18.4 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the “Write-Synchronized” or the “Read-Synchronized” property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

Note: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET, and INV Registers* from Related Links.

Related Links

[6.1.9. CLR, SET and INV Registers](#)

18.4.1 Configuration Control Register 0

Name: CFGCON0(L)
Offset: 0x00
Reset: 0x7100000b
Property: -

The CFGLOCK[1:0] register bits are writable only when CFGLOCK[0] = 1'b0.

The IOLOCK, PMDLOCK and PGLOCK register bits can only be cleared on a system reset. Thereafter, it is controlled as described above.

This register is loaded with trusted data from FBCFG1 during the pre-boot period. Trusted data from Flash means when there is no BCFG* fail status and BINFOVALID = 0 during Flash configuration word reads. If accompanied by a fail status or blank/erase indication, then reset values (described in the register description below) are retained and new values from FBCFG1 are not loaded.

Under all conditions, Flash loading is omitted for the following bits in the CFGCON0 and HPLUGDIS register:

- IOLOCK
- CFGLOCK[1:0]
- PMDLOC
- PGLOCK
- PMULOCK
- JTAGEN

Bit	31	30	29	28	27	26	25	24
		FRECCDIS	FECCCON/ECCCTL[1:0]	ADCFEN	INT0P	INT0E		PCM
Access		R/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset		1	1	1	0	0	0	1
Bit	23	22	21	20	19	18	17	16
	SLRTEN2	SLRTEN1	SLRTEN0	HPLUGDIS	SMBUSEN2	SMBUSEN1	SMBUSEN0	VBCMODE
Access	R/W/L	R/W/L	R/W/L	R/W	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CFGLOCK[1:0]		IOLOCK	PMDLOCK	PGLOCK	PMULOCK	RTCOUT_ALT EN	RTCIN0_ALTE N
Access	R/W/L	R/W/L	R/S/L	R/S/L	R/S/L	R/S/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CPENFILT	ACCOMP1_ALT EN	GPSOSCE*	ADCOPVR	JTAGEN	TROEN	SWOEN	TDOEN
Access	R/W/L	R/W/L	R/W/L	R/W	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	1	0	1	1

Bit 30 – FRECCDIS Flex RAM ECC Control

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- Only a read-only fuse bit, sets the initialization value of RAMECC Control. "True" RAMECC override is available in RAMECC module.

Value	Description
1	ECC is disabled
0	ECC is enabled

Bits 29:28 – FECCCON/ECCCTL[1:0] Flash ECC Control

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	ECC and dynamically ECC are disabled
10	ECC and dynamically ECC are disabled
01	Dynamically ECC is enabled
00	ECC is enabled (NVMOP = Word Programming disabled)

Bit 27 – ADCFCEN ADC FC Channel Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Exclusive ADC FC Channel Enable (Disables all second/third class channels)
0	ADC FC Channel Disable (Only second/third class channels are enabled)

Bit 26 – INT0P INT0P Polarity

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	INT0 Polarity (Positive)
0	INT0 Polarity (Negative)

Bit 25 – INT0E INT0 Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	INT0 is enabled
0	INT0 is disabled

Bit 24 – PCM PCHE I/D Cacheable Mode

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Always enabled from outside. Can be further enabled/disabled by PCHE SFR registers.
0	The cache-ability is controlled by the CPU via HPROT[3]. This feature is not available on all the ARM cores.

Bit 23 – SLRTEN2 SLRT Enable for SERCOM2

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	Slew rate is enabled
0	Slew rate is disabled

Bit 22 – SLRTEN1 SLRT Enable for SERCOM1

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Slew rate is enabled
0	Slew rate is disabled

Bit 21 – SLRTEN0 SLRT Enable for SERCOM0

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Slew rate is enabled
0	Slew rate is disabled

Bit 20 – HPLUGDIS Hot Plugging Disable (outside fuse loading)
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Hot Plugging is disabled
0	Hot Plugging is enabled

Bit 19 – SMBUSEN2 SMBus Enable for SERCOM2
Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	SMBus is enabled
0	SMBus is disabled

Bit 18 – SMBUSEN1 SMBus Enable for SERCOM1
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	SMBus is enabled
0	SMBus is disabled

Bit 17 – SMBUSEN0 SMBus Enable for SERCOM0
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	SMBus is enabled
0	SMBus is disabled

Bit 16 – VBCMODE VBC Operating Mode
Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- Do not change this field if there are pending accesses to VDDBKUPCORE memory map. Failing to do so may result in unexpected data.

Value	Description
1	Indirect addressing. The VDDBKUPCORE IO mapped using PMU Controller.
0	Direct addressing. The VDDBKUPCORE memory mapped on PB-Bridge-B.

Bits 15:14 – CFGLOCK[1:0] Configuration Register Lock
Note: This bit is only writable when CFGLOCK[1:0] is '00' or '10'.

Value	Description
11	All NVR memory self-writes, Boot Configuration (BCFG0) and System Configuration registers (CFG* and USER_ID) are locked and cannot be written – CFGLOCK value cannot be changed.
10	All NVR memory self-writes, Boot Configuration (BCFG0) and System Configuration registers (CFG* and USER_ID) are locked and cannot be written – CFGLOCK value can be changed.
01	Reserved for future use
00	All NVR memory self-writes, Boot Configuration (BCFG0) and System Configuration registers (CFG* and USER_ID) are not locked and can be written – CFGLOCK value can be changed.

Bit 13 – IOLOCK I/O Lock
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I/O Remap SFR bits are locked and cannot be modified
0	I/O Remap SFR are not locked and can be modified

Bit 12 – PMDLOCK Peripheral Module Disable (PMD) Lock

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	PMDx SFR bits are locked and cannot be modified
0	PMDx SFR bits are not locked and can be modified

Bit 11 – PGLOCK Permission Group Lock

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	CFGPG SFR bits are locked and cannot be modified
0	CFGPG SFR bits are not locked and can be modified

Bit 10 – PMULOCK PMU Controller Register Lock

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	PMU* SFR bits are locked and cannot be modified
0	PMU* SFR bits are not locked and can be modified

Bit 9 – RTCOUT_ALTEN RTCOUT Alternate Enable

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- RTC alternate output is unavailable on PA10 in sleep modes (Deep Sleep and Extreme Deep Sleep).

Value	Description
1	RTC/OUT is available on PA10
0	RTC/OUT is available on PA4

Bit 8 – RTCIN0_ALTEN RTCIN0 Alternate Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	RTC/IN0 is available on PA9
0	RTC/IN0 is available on PA3

Bit 7 – CPENFILT ADC CP Filter Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	ADC CP filter is enabled
0	ADC CP filter is disabled

Bit 6 – ACCMP1_ALTEN AC CMP1 Alternate Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	AC/CMP1 Out is available on PA6
0	AC/CMP1 Out is available on PA1

Bit 5 – GPSOSCE* GPIO/SOSC Enable* This bit is not applicable to 48-pin variants.

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	SOSC is selected
0	GPIO is selected

Bit 4 – ADCOPVR ADC Charge Pump Override

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is not fuse loadable.

Value	Description
1	Overridden (software controlled)
0	Hardware controlled

Bit 3 – JTAGEN JTAG Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	JTAG port is enabled
0	JTAG port is disabled

Bit 2 – TROEN Trace Output Enable

Notes:

- When CFGCON1.TRCEN = 0, the value of this bit is ignored but has the effect of being '0'.
- This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Starts Trace Clock and enables Trace Outputs (Trace probe must be present)
0	Stops Trace Clock and disables Trace Outputs

Bit 1 – SWOEN SWO enable on 2-wire debug interface

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	SWO is enabled
0	SWO is disabled

Bit 0 – TDOEN TDO enable for 2-wire JTAG

Implementing the JTAG protocol over the 2-wire interface requires four 2-wire clocks for each TCK if TDO is required. However, if the values shifted out TDO are predetermined, then TDO can be disabled, saving two 2-wire clocks.

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	2-wire JTAG protocol uses TDO (Four phase (Full Duplex) protocol)
0	2-wire JTAG protocol does not use TDO (Two phase (Half Duplex) protocol)

18.4.2 Configuration Control Register 1

Name: CFGCON1(L)
Offset: 0x10
Reset: 0x1f00443b
Property: -

This register is loaded with trusted data from FBCFG2 during the pre-boot period. Thereafter, it is controlled as described above.

Trusted data from Flash means when there is no BCFG* fail status during Flash configuration word reads. If accompanied by fail status or blank/erase indication, then reset values (described in the register description below) are retained, and new values from FBCFG2 are not loaded.

Under all conditions, Flash loading is omitted for the following bits in the CFGCON1 register:

- DEBUG[1:0]

Bit	31	30	29	28	27	26	25	24
		CLKZBREF	QSPIDDRM	WDTPSS[4:0]				
Access		R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset		0	0	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	I2CDSEL2	I2CDSEL1	I2CDSEL0	CCL_OE	SCOM2_HSE N	SCOM1_HSE N	SCOM0_HSE N	QSPI_HSEN
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	QSCHE_EN	SMCLR	SLRCTRL2	SLRCTRL1	SLRCTRL0	CLASSBDIS	CMP1_OE	CMP0_OE
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
	ZBTWKSYS		TRCEN	ICESEL[1:0]			DEBUG[1:0]	
Access	R/W/L		R/W/L	R/W/L	R/W/L		R/W/L	R/W/L
Reset	0		1	1	1		1	1

Bit 30 – CLKZBREF External Reference Clock Zigbee Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Enable clk_zb_to_ext on PPS.REFO1
0	No clk_zb_to_ext on PPS.REFO1, PPS.REFO1 is unchanged

Bit 29 – QSPIDDRM QSPI DDR Mode Clock Enable

Notes:

- When using the QSPI DDR Mode, System Clock (SYS_CLK) must be <= 48 MHz.
- This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	QSPI DDR mode clock is enabled
0	Disabled

Bits 28:24 – WDTPSS[4:0] Watchdog Timer Post-scale Select Sleep bits

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
10100	1:1048576
10011	1:524288
10010	1:262144
10001	1:131072
10000	1:65536
01111	1:32768
01110	1:16384
01101	1:8192
01100	1:4096
01011	1:2048
01010	1:1024
01001	1:512
01000	1:256
00111	1:128
00110	1:64
00101	1:32
00100	1:16
00011	1:8
00010	1:4
00001	1:2
00000	1:1

Bit 23 – I2CDSEL2 I2C Delay Select for SERCOM2

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	I ² C delay is enabled
0	I ² C delay is disabled

Bit 22 – I2CDSEL1 I2C Delay Select for SERCOM1

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I ² C delay is enabled
0	I ² C delay is disabled

Bit 21 – I2CDSELO I2C Delay Select for SERCOM0

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I ² C delay is enabled
0	I ² C delay is disabled

Bit 20 – CCL_OE CCL Pads (via PPS) Output Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	CCL pads (via PPS) output is enabled
0	CCL pads (via PPS) output is disabled

Bit 19 – SCOM2_HSEN SERCOM2 (Direct) Enable

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	Direct mode (High-Speed) is enabled
0	Via PPS is enabled

Bit 18 – SCOM1_HSEN SERCOM1 (Direct) Enable

Note:

- This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Direct mode (High-Speed) is enabled
0	Via PPS is enabled

Bit 17 – SCOM0_HSEN SERCOM0 (Direct) Enable

Note:

- This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Direct mode (High-Speed) is enabled
0	Via PPS is enabled

Bit 16 – QSPI_HSEN QSPI (Direct) Enable

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	Direct Mode (High-Speed) is enabled
0	Via PPS is enabled

Bit 15 – QSCHE_EN QSPI Address Space Cache Attribute

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Cache attribute is enabled
0	Caching is disabled

Bit 14 – SMCLR Selects CRU handling of MCLR Control

Note: This bit is only writable when CFGLOCK[1:0] is '00' or '10'.

Value	Description
1	Legacy mode (system clear does not reset all state of device)
0	MCLR causes a faux POR

Bit 13 – SLRCTRL2 I2C Delay Select for SERCOM2

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
1	Slew rate control is configured via SERCOM configuration
0	Slew rate control is configured via GPIO configuration

Bit 12 – SLRCTRL1 I2C Delay Select for SERCOM1

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Slew rate control is configured via SERCOM configuration
0	Slew rate control is configured via GPIO configuration

Bit 11 – SLRCTRL0 I2C Delay Select for SERCOM0

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Slew rate control is configured via SERCOM configuration
0	Slew rate control is configured via GPIO configuration

Bit 10 – CLASSBDIS Disable CLASSB Device Functionality

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	CLASSB functions are disabled
0	CLASSB functions are enabled

Bit 9 – CMP1_OE Analog Comparator-1 Output Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	AC1 Output is enabled
0	AC1 Output is disabled

Bit 8 – CMP0_OE Analog Comparator-0 Output Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	AC0 Output is enabled
0	AC0 Output is disabled

Bit 7 – ZBTWKSYS Zigbee Bluetooth Subsystem External Wake-up source

Notes:

- Write-only bit, with read-as-zero; when written to '1', creates one clk_lp_cycle wide pulse on Zigbee Bluetooth Subsystem.external_NMI0 pin. This enables external system wake-up to Bluetooth subsystem. This allows CPU and Bluetooth subsystem wake-up/sleep to be independent of each other.
- Flash fuse loading is excluded for this bit.

Bit 5 – TRCEN Trace Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Trace features in the CPU are enabled
0	Trace features in the CPU are disabled

Bits 4:3 – ICESEL[1:0] EMUC/EMUD Communication Channel Select

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	ICE EMUC1/EMUD1 pins are shared with PGC1/PGD1
10	ICE EMUC2/EMUD2 pins are shared with PGC2/PGD2
01	ICE EMUC3/EMUD3 pins are shared with PGC3/PGD3 (Not used on this device)
00	ICE EMUC4/EMUD4 pins are shared with PGC4/PGD4

Bits 1:0 – DEBUG[1:0] Background Debugger Access Selection

Notes:

1. JTAGEN = 0 prevents 4-wire JTAG Debugging but not EMUC/EMUD debugging.
2. If CPN = 0, then the JTAG TAP controller denies access to the EJTAG TAP Controller (i.e., the SWTAP command is ignored) and, therefore, external access to the debugging features is denied.
3. This bit is only writable when CFGLOCK[1:0] = '00'.

Value	Description
11	4-wire JTAG I/F is enabled; EMUC/EMUD is disabled; ICD module is disabled
10	4-wire JTAG I/F is enabled; EMUC/EMUD is disabled; ICD module is enabled
01	EMUC/EMUD is enabled; 4-wire JTAG I/F is disabled; ICD module is disabled
00	EMUC/EMUD is enabled; 4-wire JTAG I/F is disabled; ICD module is enabled

18.4.3 Configuration Control Register 2

Name: CFGCON2(L)
Offset: 0x20
Reset: 0x00
Property: -

This register is loaded with trusted data from FBCFG3 during pre-boot period. Thereafter, it is controlled as described above.

Trusted data from Flash means when there is no BCFG* fail status during Flash configuration word reads. If accompanied by fail status or blank/erase indication then reset values (described in the register description below) are retained and new values from FBCFG3 are not loaded.

Bit	31	30	29	28	27	26	25	24
	DMTEN	DMTCNT[4:0]				WINSZ[1:0]		
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	WDTEN	WINDIS	WDTSPGM	WDTPSR[4:0]				
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	FSCMEN	CKSWEN	WAKE2SPD	SOSCSEL	WDTRMCS[1:0]		POSCMD[1:0]	
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	PMUTEST_VD D_EN		DMTINTV[2:0]			ACMP_CYCLE[2:0]		
Access	R/W/L		R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0		1	1	1	0	0	0

Bit 31 – DMTEN Dead Man Timer Enable bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	DMT is enabled
0	DMT is disabled (control is placed on the DMTCON.ON bit)

Bits 30:26 – DMTCNT[4:0] Dead Man Timer Count Select bits

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- On devices where the DMTCNT[4:0] configuration field is loaded.

Value	Description
00000	Counter value is 2^8 for cfg_dmt_cnt[31:0]
00001	Counter value is 2^9 for cfg_dmt_cnt[31:0]
...	...
10100	Counter value is 2^{28} for cfg_dmt_cnt[31:0]
10101	Counter value is 2^{29} for cfg_dmt_cnt[31:0]
10110	Counter value is 2^{30} for cfg_dmt_cnt[31:0]
10111	Counter value is 2^{31} for cfg_dmt_cnt[31:0]

Value	Description
11000 – 11111	Reserved

Bits 25:24 – WINSZ[1:0] Watchdog Timer Window Size bits

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
00	Window size is 75%
01	Window size is 50%
10	Window size is 37.5%
11	Window size is 25%

Bit 23 – WD TEN Watchdog Timer Enable bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	WDT is enabled
0	WDT is disabled (control is placed on the SWDTEN bit)

Bit 22 – WINDIS Windowed Watchdog Timer Disable bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Standard WDT is selected; windowed WDT disabled
0	Windowed WDT is enabled

Bit 21 – WDTSPGM Watchdog Timer Stop during Flash Programming bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	The WDT stops during NVR programming (legacy)
0	The WDT runs during NVR programming (for read/execute while programming Flash systems)

Bits 20:16 – WDTPSR[4:0] Watchdog Timer Post-scale Select Run bits

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
10100	1:1048576
10011	1:524288
10010	1:262144
10001	1:131072
10000	1:65536
01111	1:32768
01110	1:16384
01101	1:8192
01100	1:4096
01011	1:2048
01010	1:1024
01001	1:512
01000	1:256
00111	1:128
00110	1:64
00101	1:32
00100	1:16
00011	1:8
00010	1:4
00001	1:2

Value	Description
00000	1:1

Bit 15 – FSCMEN Fail-Safe Clock Monitor Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	FSCM enabled
0	FSCM disabled

Bit 14 – CKSWEN Software Clock Switching Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Software clock switching is enabled
0	Software clock switching is disabled

Bit 13 – WAKE2SPD 2-Speed startup enabled in Sleep mode bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	When the device exits the Sleep mode, the SYS_CLK will be from FRC until the selected clock is ready.
0	Not applicable.

Bit 12 – SOSSEL SOSC Selection Configuration bit

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Crystal (SOSCI/SOSCO) mode is selected
0	Digital (SCLKI) mode is selected

Bits 11:10 – WDTRMCS[1:0] WDT RUN Mode Clock Select

Note: This bit is only writable when CFGLOCK[1:0] is '00' or '10'.

Value	Description
11	LPRC
10	Do not use
01	Do not use
00	Module PB Clock

Bits 9:8 – POSCMD[1:0] Primary Oscillator Configuration bits

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	Primary oscillator is disabled
10	HS oscillator mode is selected
01	HS oscillator mode is selected
00	HS oscillator mode is selected

Bit 7 – PMUTEST_VDD_EN PMU Test Output or VDD/2 Enable via ADC IE12

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	PMU test output monitor is enabled
0	VDD/2 monitor is enabled

Bits 5:3 – DMTINTV[2:0] Dead Man Timer Count Window Interval bits

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
000	Window/Interval value is zero for cfg_dmt_intv[31:0] - windowed mode is disabled
001	Window/Interval value is 1/2 Counter value for cfg_dmt_intv[31:0]
010	Window/Interval value is 3/4 Counter value for cfg_dmt_intv[31:0]
011	Window/Interval value is 7/8 Counter value for cfg_dmt_intv[31:0]
100	Window/Interval value is 15/16 Counter value for cfg_dmt_intv[31:0]
101	Window/Interval value is 31/32 Counter value for cfg_dmt_intv[31:0]
110	Window/Interval value is 63/64 Counter value for cfg_dmt_intv[31:0]
111	Window/Interval value is 127/128 Counter value for cfg_dmt_intv[31:0]

Bits 2:0 – ACMP_CYCLE[2:0] AC SIB Comparator Result Wait Cycles

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
n	Wait for $32\mu s * ACMP_CYCLE + 1$ cycles to generate comparator done indication

18.4.4 Configuration Control Register 4

Name: CFGCON4(L)
Offset: 0x40
Reset: 0x840e4000
Property: -

This register is loaded with trusted data from FBCFG4/DEVCFG4 during the pre-boot period.

Trusted data from Flash means when there is no BCFG* fail status during Flash configuration word reads. If accompanied by fail status BCFGFAIL (RCON[26]) or blank/erase indication, then reset values (described in the following register description) are retained and new values from FBCFG4 are not loaded.

Bit	31	30	29	28	27	26	25	24
	RTCNTM_CSEL	LPOSCEN	UVREGROVR	DSBITEN	DSWDTEN	DSWDTLPRC	DSWDTPS[4:3]	
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	1	0	0	0	0	1	0	0
Bit	23	22	21	20	19	18	17	16
	DSWDTPS[2:0]			DSZPBOREN	CPEN_DLY[2:0]			RTCEVTYPE
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	1	1	1	0
Bit	15	14	13	12	11	10	9	8
	MLPCLK_MO	VBKP_DIVSEL	VBKP_32KCSEL[1:0]		VBKP_1KCSEL	RTCEVENT_EN	RTCEVENTSEL[1:0]	
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SOSC_CFG[7:0]							
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0

Bit 31 – RTCNTM_CSEL RTCC Counter Mode Clock Select

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Raw 32 KHz clock
0	Processed 32 KHz clock

Bit 30 – LPOSCEN Low Power (Secondary) Oscillator Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Low Power (Secondary) Oscillator, also at Reset is enabled
0	Low Power (Secondary) Oscillator is disabled

Bit 29 – UVREGROVR ULPVREG Retention Mode Override

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	ULPVREG forced in the Retention mode
0	ULPVREG controlled by XDS/DS FSM

Bit 28 – DSBITEN Deep Sleep Bit Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	DS bit in DSCON is enabled
0	DS bit in DSCON is disabled

Bit 27 – DSWDTEN Deep Sleep Watchdog Timer Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	DSWDT during deep sleep is enabled
0	DSWDT during deep sleep is disabled

Bit 26 – DSWDTLPRC Deep Sleep Watchdog Timer Reference Clock Select

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Select LPRC as DSWDT reference clock
0	Select SOSC as DSWDT reference clock

Bits 25:21 – DSWDTPS[4:0] Deep Sleep Watchdog Timer Postscale Select

The DS WDT prescaler is 32; this creates an approximate base time unit of 1 ms.

Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
11111	1:2 ³⁶ (25.7 days)
11110	1:2 ³⁵ (12.8 days)
11101	1:2 ³⁴ (6.4 days)
11100	1:2 ³³ (77.0 hours)
11011	1:2 ³² (38.5 hours)
11010	1:2 ³¹ (19.2 hours)
11001	1:2 ³⁰ (9.6 hours)
11000	1:2 ²⁹ (4.8 hours)
10111	1:2 ²⁸ (2.4 hours)
10110	1:2 ²⁷ (72.2 minutes)
10101	1:2 ²⁶ (36.1 minutes)
10100	1:2 ²⁵ (18.0 minutes)
10011	1:2 ²⁴ (9.0 minutes)
10010	1:2 ²³ (4.5 minutes)
10001	1:2 ²² (135.3 s)
10000	1:2 ²¹ (67.7 s)
01111	1:2 ²⁰ (33.825 s)
01110	1:2 ¹⁹ (16.912 s)
01101	1:2 ¹⁸ (8.456 s)
01100	1:2 ¹⁷ (4.228 s)
01011	1:65536 (2.114 s)
01010	1:32768 (1.057 s)
01001	1:16384 (528.5 ms)
01000	1:8192 (264.3 ms)
00111	1:4096 (132.1 ms)
00110	1:2048 (66.1 ms)
00101	1:1024 (33 ms)
00100	1:512 (16.5 ms)
00011	1:256 (8.3 ms)
00010	1:128 (4.1 ms)
00001	1:64 (2.1 ms)

Value	Description
00000	1:32 (1 ms)

Bit 20 – DSZPBOREN Deep Sleep Zero-Power BOR Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	ZPBOR during deep sleep is enabled
0	ZPBOR during deep sleep is disabled

Bits 19:17 – CPEN_DLY[2:0] Charge-pump Ready Digital Delay (Safety delay to Analog CP Ready)

$n = (n+1)$ LPRC Clock Cycle Delay

Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Bit 16 – RTCEVTYPE RTCC Event Type

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	RTC_EVENT
0	RTC_OUT

Bit 15 – MLPCLK_MOD LPCLK Modifier in Counter/Delay Mode

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Divide-by 1.024 (Recommended, when LPCLK = 32.768 KHz)
0	Divide-by 1 (Recommended, when LPCLK = 32 KHz)

Bit 14 – VBKP_DIVSEL VDDBUKPCORE LPCLK Clock Divider Selection

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Divide by 31.25 (Recommended, when LPCLK = 32 KHz)
0	Divide-by 32 (Recommended, when LPCLK = 32.768 KHz)

Bits 13:12 – VBKP_32KSEL[1:0] VDDBUKPCORE 32 KHz Clock Source Selection

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	LPRC
10	SOSC
01	POSC
00	FRC

Bit 11 – VBKP_1KSEL VDDBUKPCORE LPCLK Clock Selection

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Divide by 32 or 31.25 clock depending on VBKP_DIVSEL
0	32 KHz low power clock

Bit 10 – RTCEVENT_EN Output Enable for RTCC Event Output

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	RTCC-Event output is enabled
0	RTCC-Event output is disabled

Bits 9:8 – RTCEVENTSEL[1:0] RTCC Event Selection

Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
00	1-Second clock
01	Alarm pulse
1x	32 KHz clock

Bits 7:0 – SOSC_CFG[7:0] SOSC Configuration Bits

Note: These bits are only writable when CFGLOCK[1:0] is '00'.

18.4.5 Permission Group Configuration

Name: CFGPGQOS
Offset: 0x50
Reset: 0xe040004c
Property: -

All bits in this register are writable only when CFGCON0.PGLOCK = 0.

There is no Flash location for this register because the purpose of this register is to provide a software-based protection mechanism to a device memory-mapped region, which is typically handled by a trusted boot/OScode.

Note: Ensure this register is programmed to the values shown: 0xE040_004C if you are not using Microchip-provided boot code.

Bit	31	30	29	28	27	26	25	24
	WISIBQOS[1:0]		FCQOS[1:0]				DSUPG[1:0]	
Access	R/W/L	R/W/L	R/W/L	R/W/L			R/W/L	R/W/L
Reset	1	1	1	0			0	0
Bit	23	22	21	20	19	18	17	16
	ICMQOS[1:0]		ICMPG[1:0]		ADCQOS[1:0]		ADCPG[1:0]	
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
							DMAPG[1:0]	
Access							R/W/L	R/W/L
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	ICDJQOS[1:0]		ICDJPG[1:0]		CPUQOS[1:0]		CPUPG[1:0]	
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	1	0	0	1	1	0	0

Bits 31:30 – WISIBQOS[1:0] Wireless SIB QOS Control bits

Note: This field is only writable when CFGCON0.PGLOCK = 0.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 29:28 – FCQOS[1:0] FC Controller QOS Control bits

Note: This field is only writable when CFGCON0.PGLOCK = 0.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 25:24 – DSUPG[1:0] DSU Permission Group, drive the inputs `cfg_dsu_pg[1:0]` directly to the SSX
The DSU bus host has access to Access Controlled memory regions as defined by the bit-fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `DSUPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `DSUPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `DSUPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `DSUPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Bits 23:22 – ICMQOS[1:0] ICM QOS Control bits

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 21:20 – ICMPG[1:0] ICM Permission Group, drive the inputs `cfg_icm_pg[1:0]` directly to the SSX
The ICM bus host has access to Access Controlled memory regions as defined by the bit-fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `ICMPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `ICMPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `ICMPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `ICMPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Bits 19:18 – ADCQOS[1:0] ADC Controller QOS Control bits

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 17:16 – ADCPG[1:0] ADC Controller Permission Group, drive the inputs `cfg_adc_pg[1:0]` directly to the SSX

The ADC bus host has access to Access Controlled memory regions as defined by the bit-fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `ADCPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `ADCPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `ADCPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `ADCPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Bits 9:8 – DMAPG[1:0] DMA (Rd/Wr) Permission Group, drive the inputs `cfg_dma_pg[1:0]` directly to the SSX. The DMA bus host has access to Access Controlled memory regions as defined by the bit-fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `DMAPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `DMAPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `DMAPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `DMAPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Bits 7:6 – ICDJQOS[1:0] ICD-JTAG Bus QOS Control bits

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 5:4 – ICDJPG[1:0] ICD-JTAG Permission Group, drive the inputs `cfg_icdj_pg[1:0]` directly to the SSX. The ICD-JTAG bus host has access to Access Controlled memory regions as defined by the bit-fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `ICDJPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `ICDJPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `ICDJPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `ICDJPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Bits 3:2 – CPUQOS[1:0] CPU I/D and System Bus QOS Control bits

Note: This field is only writable when `CFGCON0.PGLOCK = 0`.

Value	Description
00	Disable; Background
01	Low; Sensitive bandwidth
10	Medium; Sensitive latency
11	High; Critical latency

Bits 1:0 – CPUPG[1:0] CPU (Code) Permission Group, drive the inputs `cfg_cpu_pg[1:0]` directly to the SSX. The CPU Bus host has access to Access Controlled memory regions as defined by the bit fields `RDPER[3:0]` and `WRPER[3:0]` in the Bus Structure's Permission Groups SFRs for these memory regions. The encoding works as follows:

- `CPUPG[1:0] == 2'b11` : Read Access if `RDPER[3]==1`, Write Access if `WRPER[3]==1` (Perm. Grp. 3)
- `CPUPG[1:0] == 2'b10` : Read Access if `RDPER[2]==1`, Write Access if `WRPER[2]==1` (Perm. Grp. 2)
- `CPUPG[1:0] == 2'b01` : Read Access if `RDPER[1]==1`, Write Access if `WRPER[1]==1` (Perm. Grp. 1)
- `CPUPG[1:0] == 2'b00` : Read Access if `RDPER[0]==1`, Write Access if `WRPER[0]==1` (Perm. Grp. 0)

Notes:

- CPUPG[1:0] automatically reverts to 2'b00 when the CPU acknowledges entering into an NMI exception as indicated by its STAUS[NMI] bit, which is carried by the cpu1_si_nmitaken system signal.
- This field is only writable when CFGCON0.PGLOCK = 0.

18.4.6 Peripheral Clock Generator 1

Name: CFGPCLKGEN1
Offset: 0x60
Reset: 0x00
Property: -

The CFGPCLKGEN1 dictates the peripheral clock selection described in the *Clock System* chapter.

There is no Flash location for this register because the purpose of this register is to provide an application-based peripheral clocking selection. This is best handled in the application software drivers.

Bit	31	30	29	28	27	26	25	24
	CM4TD	CM4TCSEL[2:0]			TC23CD	TC23CSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TCC12CD	TCC12CSEL[2:0]			S23D	SERCOM23CSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	S01CD	SERCOM01CSEL[2:0]			MCD	FREQMMSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RCD	FREQMRCSEL[2:0]			EICCD	EICSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – CM4TD CM4 Trace Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 30:28 – CM4TCSEL[2:0] CM4 Trace Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 27 – TC23CD TC2 and TC3 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 26:24 – TC23CSEL[2:0] TC2 and TC3 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 23 – TCC12CD TCC1 and TCC2 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 22:20 – TCC12CSEL[2:0] TCC1 and TCC2 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = '00'.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – S23D SERCOM2 and SERCOM3 Peripheral Clock Disable

Notes:

- This field is only writable when CFGLOCK[1:0] = '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – SERCOM23CSEL[2:0] SERCOM2 and SERCOM3 Peripheral Clock Selection

Notes:

- This field is only writable when CFGLOCK[1:0] = '00'.
- This bit is only applicable in 48-pin variants.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – S01CD SERCOM0 and SERCOM1 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – SERCOM01CSEL[2:0] SERCOM0 and SERCOM1 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – MCD FREQM Measurement Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – FREQMMSEL[2:0] FREQM Measurement Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – RCD FREQM Reference Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – FREQMRCSEL[2:0] FREQM Reference Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – EICCD EIC Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – EICCSEL[2:0] EIC Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

18.4.7 Peripheral Clock Generator 2

Name: CFGPCLKGEN2
Offset: 0x70
Reset: 0x00
Property: -

The CFGPCLKGEN2 dictates the peripheral clock selection described in the *Clock System* chapter.

Note that the following bits EVSYSCX range from 1-8 which corresponds to channel 0 to 7. There is no Flash location for this register because the purpose of this register is to provide an application-based peripheral clocking selection. This is best handled in the application software drivers.

Bit	31	30	29	28	27	26	25	24
	C8D	EVSYS8SEL[2:0]			C7D	EVSYS7SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	C6D	EVSYS6SEL[2:0]			C5D	EVSYS5SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	C4D	EVSYS4SEL[2:0]			C3D	EVSYS3SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	C2D	EVSYS2SEL[2:0]			C1D	EVSYS1SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – C8D EVSYSC8 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 30:28 – EVSYSC8SEL[2:0] EVSYSC8 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 27 – C7D EVSYSC7 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 26:24 – EVSYSC7SEL[2:0] EVSYSC7 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 23 – C6D EVSYSC6 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 22:20 – EVSYSC6SEL[2:0] EVSYSC6 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – C5D EVSYSC5 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – EVSYSC5SEL[2:0] EVSYSC5 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – C4D EVSYSC4 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – EVSYSC4SEL[2:0] EVSYSC4 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – C3D EVSYSC3 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – EVSYSC3SEL[2:0] EVSYSC3 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – C2D EVSYSC2 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – EVSYSC2SEL[2:0] EVSYSC2 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – C1D EVSYSC1 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – EVSYSC1SEL[2:0] EVSYSC1 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

18.4.8 Peripheral Clock Generator 3

Name: CFGPCLKGEN3
Offset: 0x80
Reset: 0x00
Property: -

The CFGPCLKGEN3 dictates the peripheral clock selection described in the *Clock System* chapter.

Note that the following bits EVSYSCX range from 9-12 which corresponds to channel 8 to 11. There is no Flash location for this register because the purpose of this register is to provide an application-based peripheral clocking selection. This is best handled in the application software drivers.

Bit	31	30	29	28	27	26	25	24
	TC1CD	TC1CSEL[2:0]			TC0CD	TC0CSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TCC0CD	TCC0CSEL[2:0]			ACCD	ACCLKSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	C12D	EVSYS12SEL[2:0]			C11D	EVSYS11SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	C10D	EVSYS10SEL[2:0]			C9D	EVSYS9SEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – TC1CD TC1 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 30:28 – TC1CSEL[2:0] TC1 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 27 – TC0CD TC0 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 26:24 – TC0CSEL[2:0] TC0 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 23 – TCC0CD TCC0 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 22:20 – TCC0CSEL[2:0] TCC0 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – ACCD AC Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – ACCLKSEL[2:0] AC Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – C12D EVSYSC12 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – EVSYSC12SEL[2:0] EVSYSC12 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – C11D EVSYSC11 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – EVSYSC11SEL[2:0] EVSYSC11 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – C10D EVSYSC10 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – EVSYSC10SEL[2:0] EVSYSC10 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – C9D EVSYSC9 Peripheral Clock Disable

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – EVSYSC9SEL[2:0] EVSYSC9 Peripheral Clock Selection

Note: This field is only writable when CFGLOCK[1:0] = 00.

Value	Description
0	No clock is selected
1–6	REFO1-6 clock is selected
7	Low power clock is selected

18.4.9 User Unique ID

Name: USER_ID
Offset: 0xA0
Reset: 0x00
Property: -

The User ID is a 16-bit ID that may be programmed to differentiate products that use the same device. The User ID value may be read directly out of the USER_ID register or through the JTAG interface via the MCHP_CMD.USER ID command.

There is no dedicated JTAG status bit to indicate when the User ID value is loaded into the USER_ID register and is ready to be read from JTAG. It is assumed that a non-zero value for the User ID will be used to indicate that the User ID is loaded.

The USER_ID register is reset on power-up, then is loaded with trusted data from FBCFG5 during the pre-boot period and it is controlled.

Trusted data from Flash means when there is no BCFG* fail status during Flash configuration word reads. If accompanied by fail status or blank/erase indication, then Reset values (described in the register description below) are retained and new values from FBCFG5 are not loaded.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	USER_ID[15:8]							
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	USER_ID[7:0]							
Access	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – USER_ID[15:0] User unique ID, readable using the JTAG USER_ID instruction

Note: This field is only writable when CFGLOCK[1:0] = 00.

18.4.10 Boot Configuration 0

Name: BCFG0
Offset: 0x200
Reset: 0x00
Property: -

Note: Safe value of BCFG0 is 0xFFFF_FFFF as applicable only to the implemented bits.

Bit	31	30	29	28	27	26	25	24
	BINFOVALID0		SIGN	CP				
Access	R		R	R				
Reset	c		c	c				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							PCSCMODE	
Access							R	
Reset							c	

Bit 31 – BINFOVALID0 First 256-bit BCFG information is valid

Notes:

1. This bit is added to provide a mechanism to determine if information coming from Flash is valid or invalid. The BCFG area is critical to the device boot-up.
2. Trusted FBCFG* data = (BINFOVALID = 0) and (BCFGFAIL = 0).
3. It is recommended that the application program this bit to zero for proper operation.

Value	Description
1	FBCFG0 to FBCFG5 is not valid (Untrusted, Flash values are ignored and safe values are used)
0	FBCFG0 to FBCFG5 is valid (Trusted and loaded from Flash)

Bit 29 – SIGN Flash SIGN bit

Value	Description
1	Unsigned
0	Signed

Bit 28 – CP Boot Code Protect (~FCPN0.CPN && ~FSIGN0.SIGN).

Value	Description
1	Protection is enabled
0	Protection is disabled

Bit 1 – PCSCMODE PCHE Single Cache mode

Value	Description
1	PCHE ICache Only. CPU Instructions (code, data) go to PCHE ICache only.
0	PCHE ICache and DCache. CPU opcodes go to PCHE ICache port and data goes to PCHE DCache port.

19. Register Locking

This device provides several different types of register-level locking:

Locking using the System Lock Register – This mechanism, described in System Lock Register (see *System Lock Register* from Related Links), provides for a 2-way (locking and unlocking) write lock of system critical registers. It includes protection for the following registers:

- CRU.OSCCON
- CRU.OSCTRM
- CRU.SPILLCON
- CRU.RSWRST
- CRU.RNMICON
- CRU.PB1DIV
- CRU.PB2DIV
- CRU.PB3DIV
- CRU.SLEWCON
- CRU.CLK_DIAG

Locking using the CFGCON0.IOLOCK, CFGCON0.PMDLOCK, CFGCON0.PMUCLOCK and CFGCON0.PGLOCK register bits – This mechanism provides for a 1-way lock (once locked, only a reset can unlock) of the following registers:

- All PPS registers (IOLOCK bit)
- All PMD registers (PMDLOCK bit)
- CFGPG register (PGLOCK bit)
- All PMU registers (PMULOCK bit)

Locking using the CFGCON0.CFGLOCK[1:0] register bits – This mechanism provides for a 1-way or 2-way lock (software selectable). It applies to the following registers and memories:

- BCFG0
- CFGCON0
- CFGCON1
- CFGCON2
- CFGCON3
- USER_ID
- CFGCON4
- CFGPCLKGENx

Related Links

[19.1. System Lock Register](#)

19.1 System Lock Register

Several modules contain registers that are protected from errant code causing unwanted changes by the system lock feature. When the system lock is in effect, which is the default state, registers protected by it are not writable. The system lock feature protects registers that are system critical such as the boot time option.

Each module that uses the system lock feature describes which register bits and functions it affects. A specific sequence of writes to the SYSKEY register unlock the access to those register bits and features.

19.1.1 Unlock Requirements

The unlock sequence must be atomic. If any other peripheral bus access occurs on the same peripheral bus on which SYSKEY resides during the unlock attempt sequence, the unlock fails. Therefore, turn off all bus initiators like DMA, USB and so on, and disable interrupts.

19.1.2 Unlock Sequence

It is recommended that application code performs step 2 and 3 before step 5 and 6. The unlock sequencer, however, only looks for step 5 and 6 to be atomic. For this sequence, atomic means that there is no other activity on the peripheral bus between step 5 and 6. Step 4 is only needed to ensure that the sequence starts from a known locked state.

1. Suspend all other Peripheral Bus accesses.
2. Load 0xAA996655 to CPU register X.
3. Load 0x556699AA to CPU register Y.
4. Store CPU register r0 to SYSKEY.
5. Store CPU register X to SYSKEY.
6. Store CPU register Y to SYSKEY.

19.1.3 Lock Sequence

When the system is unlocked, any write to the SYSKEY register causes the system lock to become active.

19.1.4 Lock/Unlock Indication

The SYSKEY register read value indicates the status of the unlock sequence. A value of 0x00000000 indicates the system is still locked. A value of 0x00000001 indicates the sequence succeeded and the system is unlocked.

19.2 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0xAF	Reserved									
0xB0	SYSKEY	7:0	SYSKEY7	SYSKEY6	SYSKEY5	SYSKEY4	SYSKEY3	SYSKEY2	SYSKEY1	SYSKEY0
		15:8	SYSKEY15	SYSKEY14	SYSKEY13	SYSKEY12	SYSKEY11	SYSKEY10	SYSKEY9	SYSKEY8
		23:16	SYSKEY23	SYSKEY22	SYSKEY21	SYSKEY20	SYSKEY19	SYSKEY18	SYSKEY17	SYSKEY16
		31:24	SYSKEY31	SYSKEY30	SYSKEY29	SYSKEY28	SYSKEY27	SYSKEY26	SYSKEY25	SYSKEY24

19.3 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the “Write-Synchronized” or the “Read-Synchronized” property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

Note: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET, and INV Registers* from Related Links.

Related Links

[6.1.9. CLR, SET and INV Registers](#)

19.3.1 System Key Register

Name: SYSKEY

Offset: 0xB0

Reset: 0x0

Bit	31	30	29	28	27	26	25	24
	SYSKEY31	SYSKEY30	SYSKEY29	SYSKEY28	SYSKEY27	SYSKEY26	SYSKEY25	SYSKEY24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SYSKEY23	SYSKEY22	SYSKEY21	SYSKEY20	SYSKEY19	SYSKEY18	SYSKEY17	SYSKEY16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SYSKEY15	SYSKEY14	SYSKEY13	SYSKEY12	SYSKEY11	SYSKEY10	SYSKEY9	SYSKEY8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SYSKEY7	SYSKEY6	SYSKEY5	SYSKEY4	SYSKEY3	SYSKEY2	SYSKEY1	SYSKEY0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SYSKEY System Key

Keys are written to this register as part of a sequence to unlock system critical registers. A successful key write to this register will set the system signal.

20. Peripheral Module Disable Register (PMD)

20.1 Overview

This section describes the following peripheral module disable functions:

- Device peripheral configuration
- Configuration defined by product variants
- Peripheral disable for power conservation

20.2 Enabling Peripherals

Peripheral Module Disable (PMD) register bits control the operation of individual peripherals on the device. When a peripheral's associated PMD bit is zero (0), the peripheral is enabled and operates as programmed. However, when the associated PMD bit is one (1), the peripheral logic, memory map and SFR bits are completely removed from visibility and the peripheral is held in Reset. This disabled state provides for the lowest power state of the peripheral.

Before a peripheral may be configured or used, it must be enabled by clearing the corresponding PMD register bit.

There are some caveats to using PMD bits. The following must be observed:

1. Disabling a peripheral while its ON bit is zero (0) results in undefined behavior of the external interface.
2. For bus initiators, software must verify the module is not busy after setting the ON bit to zero (0) before disabling it.
3. Setting the PMD bit when there is a pending interrupt results in undefined behavior. Therefore, all Interrupt Flags must be cleared prior to setting the associated PMD.

20.3 Registers and Bits

Note: PMD registers can be write-locked using the CFGCON0.PMDLOCK register bit. If this bit is set, then writing of PMD registers has no effect.

See *PMD1*, *PMD2* and *PMD3* in the *PMD Register Summary* from Related Links for a description of the PMD registers, and to identify the location of the register, see *Device Configuration Map* in the *Product Memory Mapping Overview* from Related Links.

Related Links

- [8. Product Memory Mapping Overview](#)
- [20.4.1. PMD Register Summary](#)

20.4 PMD Register

20.4.1 PMD Register Summary

Note: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0xBF	Reserved									
0xC0	PMD1	7:0	ADCMD	ACMD		PLVDMD	LPAMD	MPAMD	BTMD	ZBMD
		15:8								ADCSARMD
		23:16								RTCCMD
		31:24			SQIMD					

Related Links

[6.1.9. CLR, SET and INV Registers](#)

20.4.2 Register Description

Some peripherals include module enable bits internally. The PMD bit is used for clock gating of the PBx_CLK and GCLK for all peripherals. If the peripheral also includes the internal enable bit, the PMD bit and internal enable configuration bit must be configured by software for that peripheral.

The following table summarizes each peripheral's enable and disable controls. For more details on the internal enable/disable control, see *Peripheral Access Controller (PAC)* from Related Links.

Table 20-1. Module Enable/Disable Controls

Module	PMD control	Module control	Enable/Disable Strategy
AC	Present	Present	Disable at PMD or Module
AES	Present	Present	Disable at PMD or Module
CCL	NA	Present	Disable at Module
CMCC	NA	Present	Disable at Module
DMAC	NA	Present	Disable at Module
DSU	NA	NA	Always Enabled (Dynamic On/Off)
EIC	NA	Present	Disable at Module
EVSYS	NA	NA	Always Enabled (Dynamic On/Off)
FREQM	NA	Present	Disable at Module
ICM	Present	Present	Enable both/Disable at PMD or Module
PAC	NA	NA	Always Enabled (Dynamic On/Off)
PUKCC	Present	NA	Disable at PMD
QSPI	Present	Present	Enable both/Disable at PMD or Module
RAMECC	NA	NA	Disable using fuse bit
RTCC	Present	Present	Enable both/Disable at PMD or Module
SERCOM	Present	Present	Enable both/Disable at PMD or Module
TC	Present	Present	Enable both/Disable at PMD or Module
TCC	Present	Present	Enable both/Disable at PMD or Module
TRNG	Present	Present	Enable both/Disable at PMD or Module

Note: For Modules with both PMD control and Module control, Enable = PMD_x=0 AND Module Enable=1, Disable =PMD_x=1 OR Module Enable=0.

Related Links

[26. Peripheral Access Controller \(PAC\)](#)

20.4.3 PMD1 – Peripheral Module Disable 1 Register

Name: PMD1
Offset: 0x00C0
Reset: 0x00000000
Property: -

Notes:

- This bit is only writable when CFGCON0.PMDLOCK = 0.
- All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.

Bit	31	30	29	28	27	26	25	24
			SQIMD					
Access			R/W/L					
Reset			0					

Bit	23	22	21	20	19	18	17	16
								RTCCMD
Access								R/W/L
Reset								0

Bit	15	14	13	12	11	10	9	8
								ADCSARMD
Access								R/W/L
Reset								0

Bit	7	6	5	4	3	2	1	0
	ADCMD	ACMD		PLVDMD	LPAMD	MPAMD	BTMD	ZBMD
Access	R/W/L	R/W/L		R/W/L	R/W/L	R/W/L	R/W/L	R/W/L
Reset	0	0		0	0	0	0	0

Bit 29 – SQIMD SQI Module Disable

Value	Description
1	SQI module is disabled
0	SQI module is enabled

Bit 16 – RTCCMD RTCC Module Disable (Unused at top level, part of XDS controller SFR)

Value	Description
1	RTCC module is disabled
0	RTCC module is enabled

Bit 8 – ADCSARMD Shared ADC SAR Core Module Disable bit

Value	Description
1	ADC SAR Core module is disabled. When disabled, the corresponding ADC SAR SHARED will be disabled.
0	ADC SAR Core module is enabled

Bit 7 – ADCMD ADC Controller Module Disable

Value	Description
1	ADC Controller module is disabled
0	ADC Controller module is enabled

Bit 6 – ACMD AC Module Disable

Value	Description
1	AC module is disabled

Value	Description
0	AC module is enabled

Bit 4 – PLVDMD PLVD Module Disable bit

Value	Description
1	PLVD module is disabled. When disabled, the corresponding PLVD will be disabled.

Bit 3 – LPAMD RF LPA Module Disable bit

Value	Description
1	RF LPA module is disabled
0	RF LPA module is enabled

Bit 2 – MPAMD RF MPA Module Disable bit

Value	Description
1	RF MPA module is disabled
0	RF MPA module is enabled

Bit 1 – BTMD Bluetooth Module Disable bit

Value	Description
1	Bluetooth module is disabled
0	Bluetooth module is enabled

Bit 0 – ZBMD Zigbee Module Disable bit

Value	Description
1	Zigbee module is disabled
0	Zigbee module is enabled

Related Links

6.1.9. CLR, SET and INV Registers

20.5 PMDx Initialization Values by Variant Name

Table 20-2. PMDx Initialization Values

[illegible]

PartName	31							24		23					16	15			8	7							0	Hex			
	PMD3																														
																	TCC2MD	TCC1MD	TCC0MD	TC3MD	TC2MD	TC1MD	TC0MD	AESMD	RNGMD	PUKCCMD	ICMMD	SER3MD	SER2MD	SER1MD	SER0MD
PIC32CX1012BZ25048	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ¹	0 ¹	0 ¹	0 ¹	0 ¹	0 ¹	0 ¹	0 ¹	0000_00001

Note:

1. The value depends on the specific Device Variant, as specified in Configuration Summary, see *Configuration Summary* from Related Links.

21. Real-Time Counter and Calendar (RTCC)

21.1 Overview

The Real-Time Counter and Calendar (RTCC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTCC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms.

The RTCC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768 kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

21.2 Features

- 32-Bit Counter with 10-Bit Prescaler
- Multiple Clock Sources
- 32-Bit or 16-Bit Counter Mode
- Clock/Calendar Mode
 - Time in seconds, minutes and hours (12/24)
 - Date in day of month, month and year
 - Leap year correction
- Digital Prescaler Correction/Tuning for Increased Accuracy
- Overflow, Alarm/Compare Match and Prescaler Interrupts and Events
 - Optional clear on alarm/compare match
- Four General Purpose Registers
- One Backup Register with Retention Capability
- Tamper Detection
 - Timestamp on event or up to four inputs with debouncing
 - Active layer protection

21.3 Block Diagram

Figure 21-1. RTCC Block Diagram (Mode 0 — 32-Bit Counter)

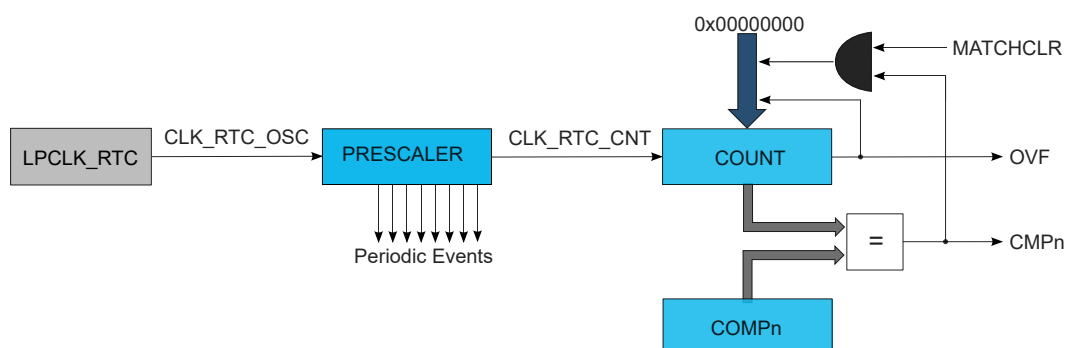


Figure 21-2. RTCC Block Diagram (Mode 1 — 16-Bit Counter)

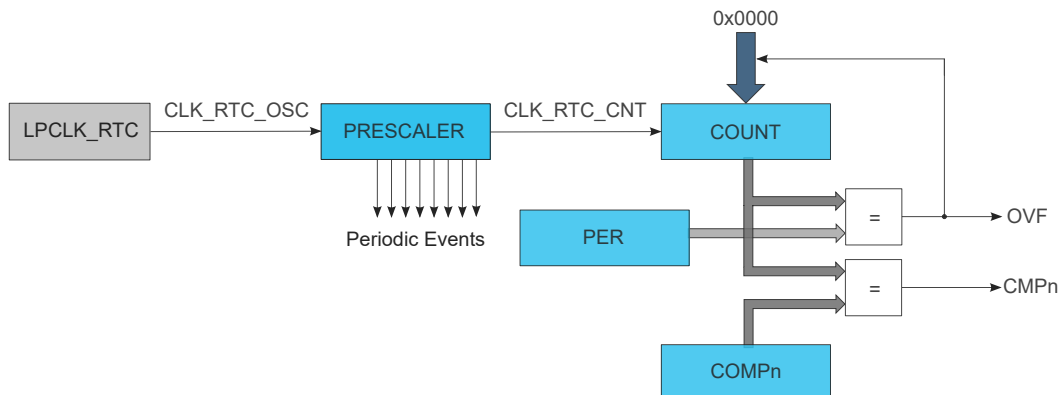


Figure 21-3. RTCC Block Diagram (Mode 2 — Clock/Calendar)

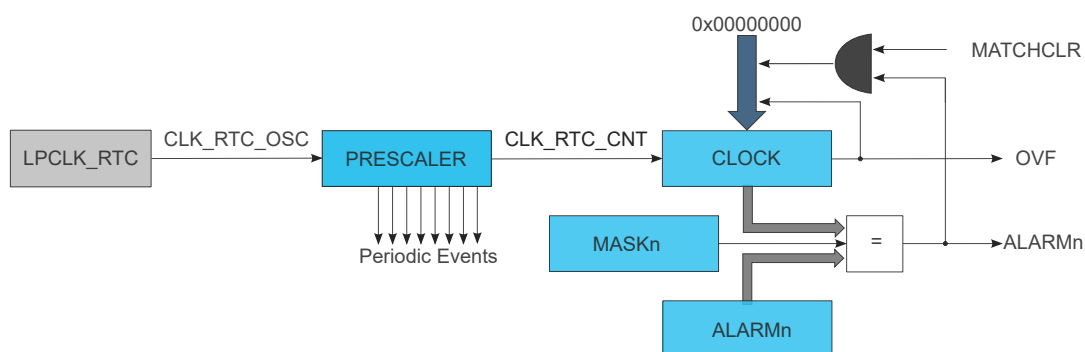
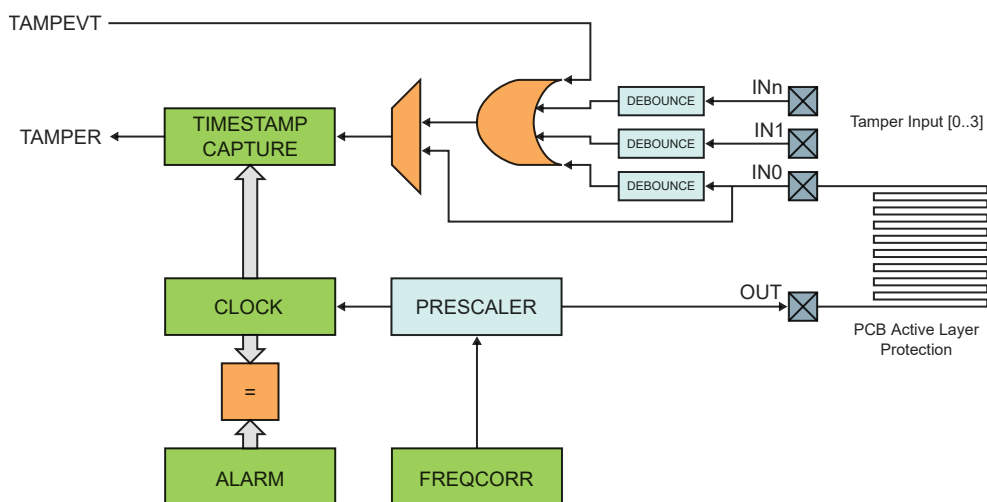


Figure 21-4. RTCC Block Diagram (Tamper Detection)



21.4 Signal Description

Table 21-1. Signal Description

Signal	Description	Type
INn [n=0..3]	Tamper Detection Input	Digital input
OUT	Tamper Detection Output	Digital output

.....continued

Signal	Description	Type
RTC_EVENT	RTC Event Output	Digital output

21.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

21.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the RTC must be enabled and no higher priority peripherals for the RTC pins can be enabled. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

[6. I/O Ports and Peripheral Pin Select \(PPS\)](#)

21.5.2 Power Management

The RTC will continue to operate in any sleep modes where the selected source clock is running. The RTC interrupts can be used to wake-up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. See *Power Management Unit (PMU)* from Related Links for details on the different sleep modes.

The RTCC can only be reset by a power on reset (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST = 1).

Related Links

[15. Power Management Unit \(PMU\)](#)

21.5.3 Clocks

A 32 KHz or 1 KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. The 32 KHz clock source can be FRC, POSC, SOSC or LPRC based on the mux selection controlled by the CFG.CFGCON4.VBKP_32KSEL bit. The 1 KHz clock source is based on the mux selection controlled by the CFG.CFGCON4.VBKP_1KSEL bit.

This oscillator clock is asynchronous to the bus clock (PB3_CLK). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains.

21.5.4 DMA

21.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

21.5.6 Events

The events are connected to the *Event System*. See *Event System (EVSYS)* from Related Links.

Related Links

[28. Event System \(EVSYS\)](#)

21.5.7 Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. See *DBGCTRL* from Related Links.

Related Links

[21.8.7. DBGCTRL](#)

21.6 Functional Description

21.6.1 Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

21.6.2 Basic Operation

21.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)
- BKUP Registers Reset On Tamper bit in Control A register (CTRLA.BKTRST)
- GP Registers Reset On Tamper Enable in Control A register (CTRLA.GPTRST)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Control B register (CTRLB)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1.024 kHz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{\text{CLK_RTC_CNT}} = \frac{f_{\text{CLK_RTC_OSC}}}{2^{\text{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{\text{CLK_RTC_OSC}}$, and $f_{\text{CLK_RTC_CNT}}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

21.6.2.2 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

21.6.2.3 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. See *RTC Block Diagram (Mode 0 — 32-Bit Counter)* figure in the *Block Diagram* from Related Links. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP). When a compare match occurs, the Compare Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMP occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMP and INTFLAG.OVF will both be set simultaneously on a compare match with COMP.

Related Links

[21.3. Block Diagram](#)

21.6.2.4 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode. See *RTC Block Diagram (Mode 1 — 16-Bit Counter)* figure in the *Block Diagram* from Related Links. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..) is set on the next 0-to-1 transition of CLK_RTC_CNT.

Related Links

[21.3. Block Diagram](#)

21.6.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode. See *RTC Block Diagram (Mode 2 — Clock/Calendar)* figure in the *Block Diagram* from Related Links. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM). When an alarm match occurs, the Alarm Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm Mask register (MASK.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see *Periodic Intervals* from Related Links).

Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARMn and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM.

Related Links

[21.3. Block Diagram](#)

[21.6.8.1. Periodic Intervals](#)

21.6.3 DMA Operation

21.6.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled, see *Periodic Intervals* from Related Links.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC, see *Nested Vector Interrupt Controller (NVIC)* from Related Links. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated, see *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

[10.2. Nested Vector Interrupt Controller \(NVIC\)](#)

[21.6.8.1. Periodic Intervals](#)

21.6.5 Events

The RTC can generate the following output events and these events can be used by EVSYS module:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled, see *Periodic Intervals* from Related Links.
- Periodic Daily (PERD): Generated when the COUNT/CLOCK has incremented at a fixed period of time.
- RTC Event (RTC_EVENT): Generates specific external signal on the RTC EVENT I/O pin.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. See *Event System (EVSYS)* from Related Links for more details on configuring the event system.

The RTC can take the following actions on an input event:

- Tamper (TAMPEVT): Capture the RTC counter to the timestamp register. See *Tamper Detection* from Related Links.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.xxxEI) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event.

RTC Event (RTC_EVENT): Other than the above events, which are mapped to the EVSYS module, the following events can generate specific external signal on the RTC EVENT I/O pin.

- 32 KHz clock
- Alarm pulse
- 1-second clock

These event signals are configured using CFGCON4.RTCEVENTSEL[1:0] bits.

Note: The RTC_OUT and RTC_EVENT signals are multiplexed and any one of the signal can be out at a time in pin limited variants. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links. The selection between RTC_OUT and RTC_EVENT is configurable through CFGCON4.RTCEVTYPE bit.

Related Links

[6. I/O Ports and Peripheral Pin Select \(PPS\)](#)

[21.6.8.1. Periodic Intervals](#)

[21.6.8.5. Tamper Detection](#)

[28. Event System \(EVSYS\)](#)

21.6.6 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU continues executing right from the first instruction that followed the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *Event System (EVSYS)* from Related Links.

Related Links

[28. Event System \(EVSYS\)](#)

21.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

21.6.8 Additional Features

21.6.8.1 Periodic Intervals

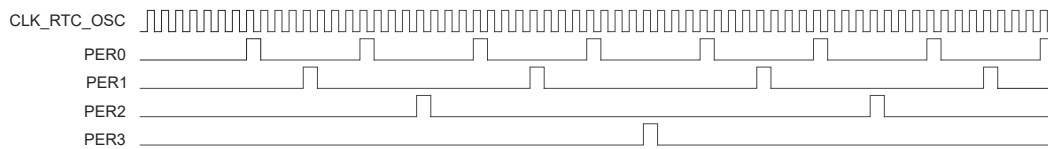
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{\text{PERIODIC}(n)} = \frac{f_{\text{CLK_RTC_OSC}}}{2^{n+3}}$$

$f_{\text{CLK_RTC_OSC}}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREO[n] bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 21-5. Example Periodic Events



Note: This example also applies to interrupts. Just replace EVCTRL.PEREN with the PERn fields of INTENCLR, INTENSET, and INTFLAG. For Modes 0 and 2, $n = 0, \dots, 7$. For Mode 1 $n = 2 \dots 7$.

21.6.8.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over of these periods. The resulting correction is as follows:

This results in a resolution of ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

21.6.8.3 Backup Registers

The RTC includes one Backup register (BKUP0). This register maintain its content in Backup/Deep Sleep mode. It can be used to store user-defined values.

If more user-defined data must be stored than the Backup register can hold, the General Purpose registers (GPn) can be used.

21.6.8.4 General Purpose Registers

The RTC includes four General Purpose registers (GPn). These registers are reset only when the RTC is reset, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

The general purpose registers $2 \cdot n$ and $2 \cdot n + 1$ are enabled by writing a '1' to the General Purpose Enable bit n in the Control B register (CTRLB.GPnEN).

The GP registers share internal resources with the COMPARE/ALARM features. Each COMPARE/ALARM register have a separate read buffer and write buffer. When the general purpose feature is enabled the even GP uses the read buffer while the odd GP uses the write buffer.

When the COMPARE/ALARM register is written, the write buffer hold temporarily the COMPARE/ALARM value until the synchronisation is complete (bit SYNCBUSY.COMPn going to 0). After the write is completed the write buffer can be used as a odd general purpose register without affecting the COMPARE/ALARM function.

If the COMPARE/ALARM function is not used, the read buffer can be used as an even general purpose register. In this case, writing the even GP will temporarily use the write buffer until the synchronisation is complete (bit SYNCBUSY.GPn going to 0). Thus an even GP must be written before writing the odd GP. Changing or writing an even GP needs to temporarily save the value of the odd GP.

Before using an even GP, the associated COMPARE/ALARM feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value.

It is recommended to use the Backup register (BKUPn) first to store user-defined values, and use the GPn only when the user-defined values exceed the capacity of the provided BKUPn.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the COMPARE/ALARM read or write buffer in all RTC modes.

Table 21-2. General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

Register	Mode 0	Mode 1	Mode 2	Write Before
GPn	COMPn/2 write buffer	(COMPn , COMPn+1) write buffer	ALARMn/2 write buffer	GPn+1
GPn+1	COMPn/2 read buffer	(COMPn , COMPn+1) read buffer	ALARMn/2 read buffer	—

21.6.8.5 Tamper Detection

The RTC provides four tamper channels that can be used for tamper detection.

The action of each tamper channel is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT):

- Off: Detection for tamper channel n is disabled.
- Wake: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register.
- Capture: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.
- Active Layer Protection: A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.

A single interrupt request (TAMPER) is available for all tamper channels.

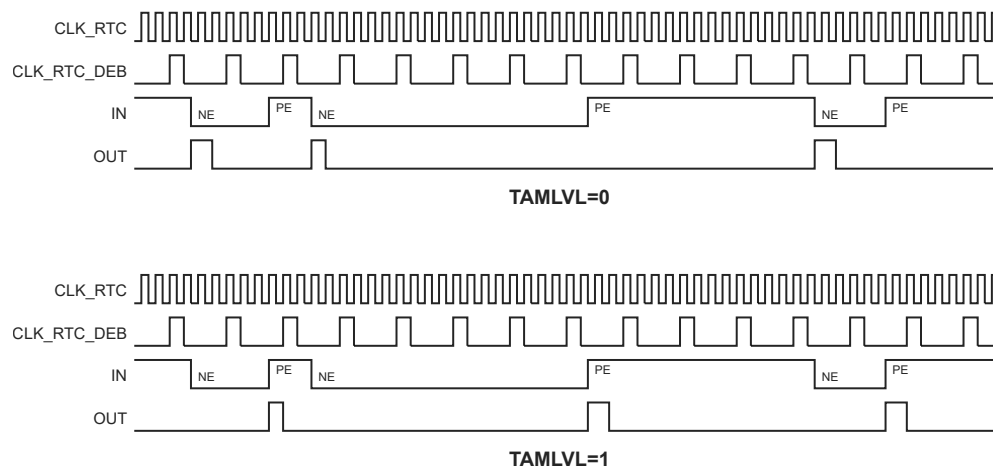
The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVTEI).

Up to four polarity external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register (CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See the following figure for an example.

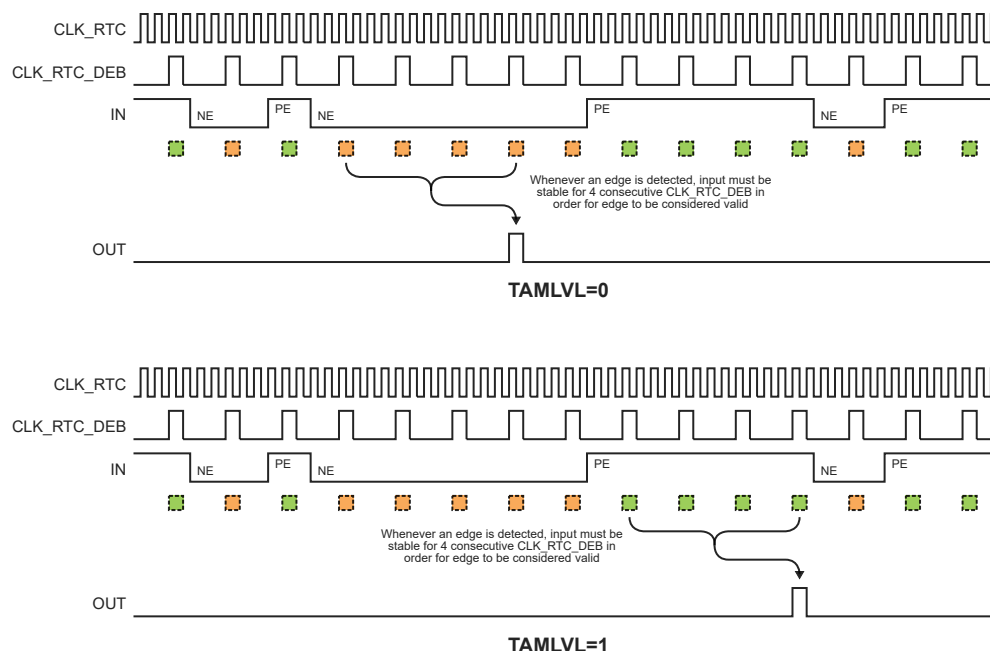
Figure 21-6. Edge Detection with Debouncer Disabled



When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. For more details, refer to the following table. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

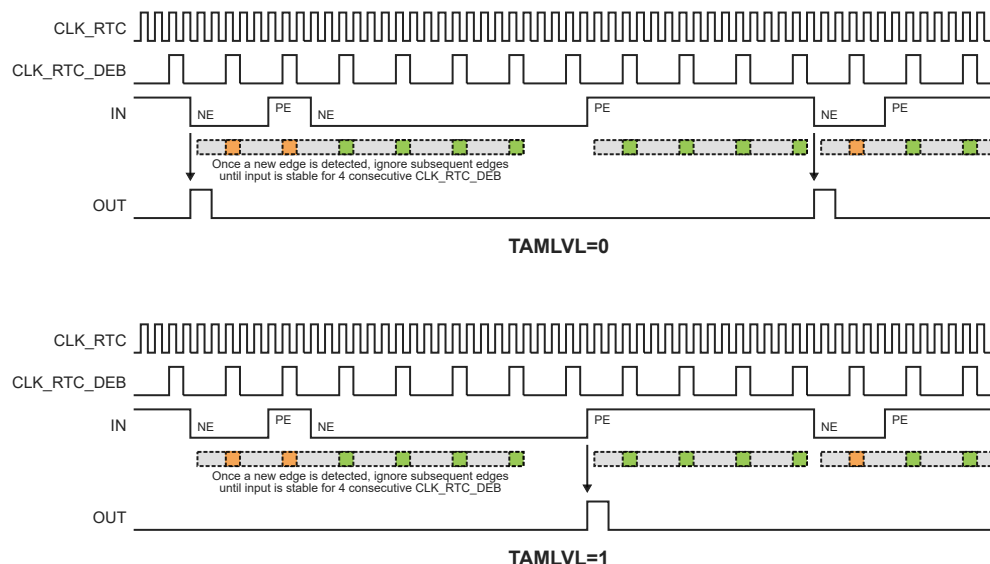
- Synchronous (CTRLB.DEBASYNC = 0): INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See the following figure for an example.

Figure 21-7. Edge Detection with Synchronous Stability Debouncing



- Asynchronous (`CTRLB.DEBASYN` = 1): The first edge on `INN` is detected. Further detection is blanked until `INN` remains stable for four `CLK_RTC_DEB` periods. See the following figure for an example.

Figure 21-8. Edge Detection with Asynchronous Stability Debouncing



Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (`CTRLB.DEBMAJ`). `INN` must be valid for two out of three `CLK_RTC_DEB` periods. See the following figure for an example.

Figure 21-9. Edge Detection with Majority Debouncing

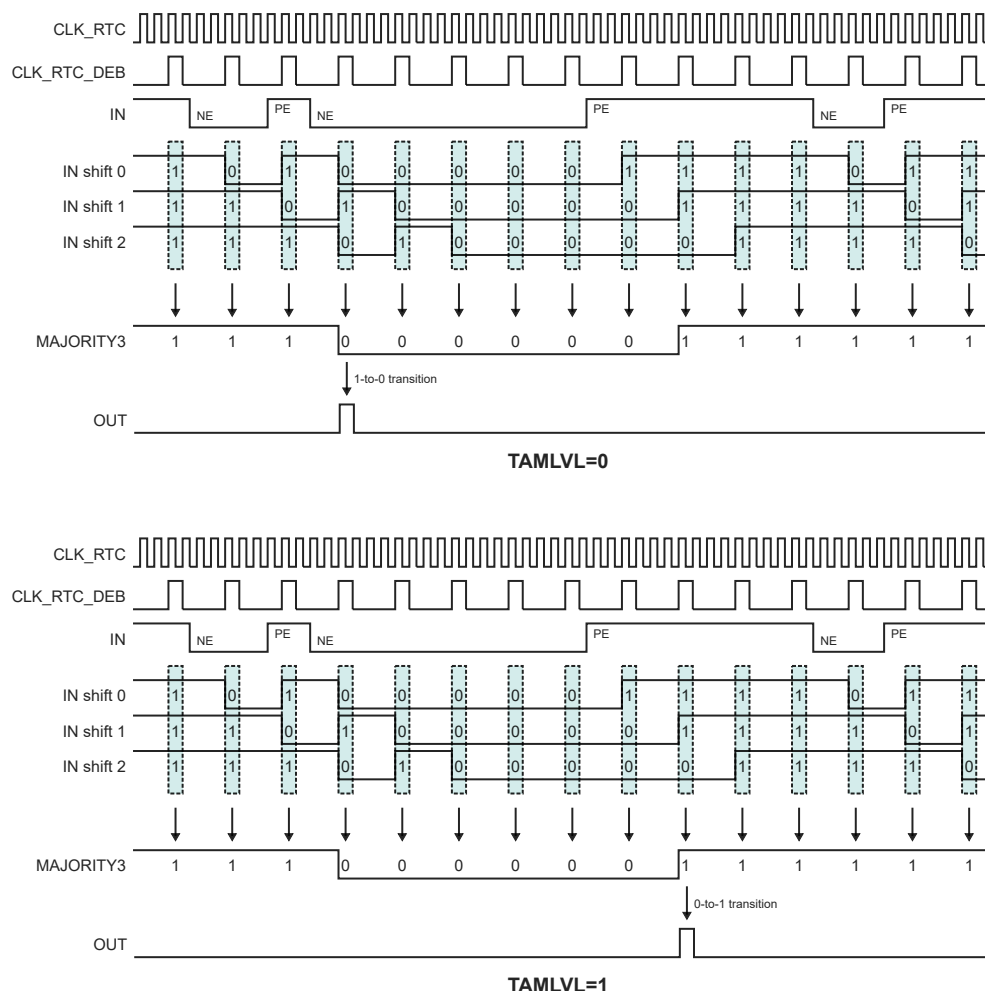


Table 21-3. Debouncer Configuration

TAMPCTRL. DEBNCn	CTRLB. DEBMAJ	CTRLB. DEBASYN	Description
0	X	X	Detect edge on INn with no debouncing. Every edge detected is immediately triggered.
1	0	0	Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	0	1	Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1').

21.6.8.5.1 Timestamp

As part of tamper detection the RTC can capture the counter value (COUNT/CLOCK) into the TIMESTAMP register. Three CLK_RTC periods are required to detect the tampering condition and capture the value. The TIMESTAMP value can be read once the Tamper flag in the Interrupt Flag

register (INTFLAG.TAMPER) is set. If the DMA Enable bit in the Control B register (CTRLB.DMAEN) is '1', a DMA request will be triggered by the timestamp. In order to determine which tamper source caused a capture, the Tamper ID register (TAMPID) provides the detection status of each tamper channel and the tamper input event. A DMA transfer can then read both TIMESTAMP and TAMPID in succession.

A new timestamp value cannot be captured until the Tamper flag is cleared, either by reading the timestamp or by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to '1'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.

Note: Once the value from the TIMESTAMP register is read, the INTFLAG.TAMPER bit must be cleared. The next value from this register must be read only after the INTFLAG.TAMPER bit is set again.

21.6.8.5.2 Active Layer Protection

The RTC provides a mean of detecting broken traces on the PCB, also known as Active layer Protection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using the RTC_OUT output pin to one RTC INn input pin. A tamper condition is detected if there is a mismatch on the generated RTC signal.

The Active Layer Protection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

Note: The Active Layer Protection works with one output pin (RTC_OUT) and multiple input pin INn. This is achieved by clearing the Separate Tamper Output bit CTRLB.SEPTO.

Enabling active layer protection requires the following steps:

- Enable the RTC prescaler output by writing a '1' to the RTC Out bit in the Control B register (CTRLB.RTCOUT). The I/O pins must also be configured to correctly route the signal to the external pins.
- Select the frequency of the output signal by configuring the RTC Active Layer Frequency field in the Control B register (CTRLB.ACTF).

$$\text{CLK_RTC_OUT} = \frac{\text{CLK_RTC}}{2^{\text{CTRLB.ACTF} + 1}}$$
- Enable the tamper input n (INn) in Active Layer mode by writing '3' to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled and INn and OUTn pin are used, the value of INn is sampled on the falling edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one half of a CLK_RTC period is available for propagation delay through the trace.
- Enable Active Layer Protection by setting CTRLB.RTCOUT bit.

21.7 Register Summary - Mode 0 - 32-Bit Counter

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC		BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
		15:8			ACTF[2:0]			DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO							
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER						
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0					COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16								
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15	Reserved									
...										
0x17										
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x1C	Reserved									
...										
0x1F										
0x20	COMP0	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
		23:16	COMP[23:16]							
		31:24	COMP[31:24]							
0x24	COMP1	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
		23:16	COMP[23:16]							
		31:24	COMP[31:24]							
0x28	Reserved									
...										
0x3F										
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x4C	GP3	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x50 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64 ... 0x67	Reserved									
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C ... 0x7F	Reserved									
0x80	BKUP0	7:0	BKUP[7:0]							
		15:8	BKUP[15:8]							
		23:16	BKUP[23:16]							
		31:24	BKUP[31:24]							

21.8 Register Description - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

21.8.1 Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC		BKTRST		PRESCALER[3:0]			
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable

The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match

This bit defines if the counter is cleared or not on a match.
This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm match
1	The counter is cleared on a Compare/Alarm match

Bits 3:2 – MODE[1:0] Operating Mode

This bit group defines the operating mode of the RTC.
This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

21.8.2 Control B in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: Enable-Protected

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYN	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 14:12 – ACTF[2:0] Active Layer Frequency

These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency

These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYN Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – **DEBMAJ** Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – **GPOEN** General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

21.8.3 Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	OVFEO							
Access	R/W							
Reset	0							
Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREO_n Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

21.8.4 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.8.5 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.8.6 Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper event

This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]

This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERn is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

21.8.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run

This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

21.8.8 Synchronization Busy in COUNT32 mode (CTRLA.MODE=0)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
Access					R	R	R	R
Reset					0	0	0	0

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

21.8.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 – 127	The RTC frequency is adjusted according to the value.

21.8.10 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value

These bits define the value of the 32-bit RTC counter in mode 0.

21.8.11 Compare 0 Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMP0
Offset: 0x20
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0] Compare Value

The 32-bit value of COMPn is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is one.

21.8.12 Compare 1 Value in COUNT32 mode (CTRLA.MODE=1)

Name: COMP1
Offset: 0x24
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0] Compare Value

The 32-bit value of COMPn is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is one.

21.8.13 General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose

These bits are for user-defined general purpose use, see *General Purpose Registers* from Related Links.

Related Links

[21.6.8.4. General Purpose Registers](#)

21.8.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNCn Debounce Enable of Tamper Input INn [n=0..3]

Note: Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..3]

Note: Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INnACT Tamper Channel n Action [n=0..3]

These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

21.8.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPIDn Tamper on Channel n Detected [n=0..3]

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

21.8.16 Backup0

Name: BKUP0
Offset: 0x80
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0] Backup

These bits are user-defined for general purpose use in the Backup domain.

21.9 Register Summary - Mode 1 - 16-Bit Counter

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC		BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASync	DEBMAJ				GP0EN
		15:8			ACTF[2:0]			DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO							
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0				PER	COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16								
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15	Reserved									
...										
0x17										
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x1A	Reserved									
...										
0x1B										
0x1C	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1E	Reserved									
...										
0x1F										
0x20	COMP0	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x22	COMP1	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x24	COMP2	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x26	COMP3	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x28	Reserved									
...										
0x30										
0x3F										
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	GP3	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x50 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64 ... 0x67	Reserved									
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C ... 0x7F	Reserved									
0x80	BKUP0	7:0	BKUP[7:0]							
		15:8	BKUP[15:8]							
		23:16	BKUP[23:16]							
		31:24	BKUP[31:24]							

21.10 Register Description - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

21.10.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC		BKTRST		PRESCALER[3:0]			
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0
Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable

The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

21.10.2 Control B in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: Enable-Protected

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 14:12 – ACTF[2:0] Active Layer Frequency

These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	$CLK_RTC_OUT = CLK_RTC / 2$
0x1	DIV4	$CLK_RTC_OUT = CLK_RTC / 4$
0x2	DIV8	$CLK_RTC_OUT = CLK_RTC / 8$
0x3	DIV16	$CLK_RTC_OUT = CLK_RTC / 16$
0x4	DIV32	$CLK_RTC_OUT = CLK_RTC / 32$
0x5	DIV64	$CLK_RTC_OUT = CLK_RTC / 64$
0x6	DIV128	$CLK_RTC_OUT = CLK_RTC / 128$
0x7	DIV256	$CLK_RTC_OUT = CLK_RTC / 256$

Bits 10:8 – DEBF[2:0] Debounce Frequency

These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	$CLK_RTC_DEB = CLK_RTC / 2$
0x1	DIV4	$CLK_RTC_DEB = CLK_RTC / 4$
0x2	DIV8	$CLK_RTC_DEB = CLK_RTC / 8$
0x3	DIV16	$CLK_RTC_DEB = CLK_RTC / 16$
0x4	DIV32	$CLK_RTC_DEB = CLK_RTC / 32$
0x5	DIV64	$CLK_RTC_DEB = CLK_RTC / 64$
0x6	DIV128	$CLK_RTC_DEB = CLK_RTC / 128$
0x7	DIV256	$CLK_RTC_DEB = CLK_RTC / 256$

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – **DEBMAJ** Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – **GPOEN** General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

21.10.3 Event Control in COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	OVFEO							
Access	R/W							
Reset	0							
Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREO_n Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

21.10.4 Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.10.5 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.10.6 Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]

This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

21.10.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run

This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

21.10.8 Synchronization Busy in COUNT16 mode (CTRLA.MODE=1)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access	COUNTSYNC							
Reset	R							
Bit	7	6	5	4	3	2	1	0
Access				PER	COUNT	FREQCORR	ENABLE	SWRST
Reset				R	R	R	R	R
				0	0	0	0	0

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bit 4 – PER Period Synchronization Busy Status

Value	Description
0	Write synchronization for PER register is complete.
1	Write synchronization for PER register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

21.10.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 – 127	The RTC frequency is adjusted according to the value.

21.10.10 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value

These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

21.10.11 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0] Counter Period

These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

21.10.12 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: $0x20 + n \cdot 0x02$ [$n=0..3$]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0] Compare Value

The 16-bit value of COMP_n is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP_n) is set on the next counter cycle.

21.10.13 General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose

These bits are for user-defined general purpose use, see *General Purpose Registers* from Related Links.

Related Links

[21.6.8.4. General Purpose Registers](#)

21.10.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNCn Debounce Enable of Tamper Input INn [n=0..3]

Note: Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..3]

Note: Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INnACT Tamper Channel n Action [n=0..3]

These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

21.10.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPIDn Tamper on Channel n Detected [n=0..3]

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

21.10.16 Backup0

Name: BKUP0
Offset: 0x80
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0] Backup

These bits are user-defined for general purpose use in the Backup domain.

21.11 Register Summary - Mode 2 - Clock/Calendar

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
		15:8	CLOCKSYNC		BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASync	DEBMAJ				GP0EN
		15:8			ACTF[2:0]				DEBF[2:0]	
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO						ALARMEO1	ALARMEO0
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0					CLOCK	FREQCORR	ENABLE	SWRST
		15:8	CLOCKSYNC							
		23:16								
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]			SECOND[5:0]				
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]	
		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM0	7:0	MINUTE[1:0]			SECOND[5:0]				
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]	
		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	MASK0	7:0						SEL[2:0]		
0x25 ... 0x27	Reserved									
0x28	ALARM1	7:0	MINUTE[1:0]			SECOND[5:0]				
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]	
		31:24	YEAR[5:0]						MONTH[3:2]	
0x2C	MASK1	7:0						SEL[2:0]		
0x2D ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	GP3	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x50 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
		15:8								
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64 ... 0x67	Reserved									
0x68	TAMPID	7:0					TAMPID3	TAMPID2	TAMPID1	TAMPID0
		15:8								
		23:16								
		31:24	TAMPEVT							
0x6C ... 0x7F	Reserved									
0x80	BKUP0	7:0	BKUP[7:0]							
		15:8	BKUP[15:8]							
		23:16	BKUP[23:16]							
		31:24	BKUP[31:24]							

21.12 Register Description - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

21.12.1 Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC		BKTRST		PRESCALER[3:0]			
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC CLOCK Read Synchronization Enable

The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match

This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation

This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete.

CTRLA.SWRST will be cleared when the reset is complete.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

21.12.2 Control B in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLB
Offset: 0x2
Reset: 0x0000
Property: Enable-Protected

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYN	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 14:12 – ACTF[2:0] Active Layer Frequency

These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency

These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Out Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYN Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GPOEN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0 disabled.
1	COMP0 compare function disabled. GP0 enabled.

21.12.3 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access	OVFEO						ALARMEO1	ALARMEO0
Reset	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
Access	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Reset	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 9 – ALARMEO1 Alarm 1 Event Output Enable

Value	Description
0	Alarm 1 event is disabled and will not be generated.
1	Alarm 1 event is enabled and will be generated for every compare match.

Bit 8 – ALARMEO0 Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREO_n Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

21.12.4 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.12.5 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

21.12.6 Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]

This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

21.12.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run

This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

21.12.8 Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
					CLOCK	FREQCORR	ENABLE	SWRST
Access					R	R	R	R
Reset					0	0	0	0

Bit 15 – CLOCKSYNC Clock Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bit 3 – CLOCK Clock Register Synchronization Busy Status

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

21.12.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 – 127	The RTC frequency is adjusted according to the value.

21.12.10 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]					MONTH[3:2]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]				HOUR[4]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year

The year offset with respect to the reference year (defined in software).
The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0] Month

1 – January
2 – February
...
12 – December

Bits 21:17 – DAY[4:0] Day

Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0] Hour

When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0] Minute

0 – 59

Bits 5:0 – SECOND[5:0] Second

0 – 59

21.12.11 Alarm n Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: ALARM
Offset: 0x20 + n*0x08 [n=0..1]
Reset: 0x00000000
Property: Write-Synchronized

The 32-bit value of ALARMn is continuously compared with the 32-bit CLOCK value, based on the masking set by MASKn.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARMn) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year

The alarm year. Years are only matched if MASKn.SEL is 6

Bits 25:22 – MONTH[3:0] Month

The alarm month. Months are matched only if MASKn.SEL is greater than 4.

Bits 21:17 – DAY[4:0] Day

The alarm day. Days are matched only if MASKn.SEL is greater than 3.

Bits 16:12 – HOUR[4:0] Hour

The alarm hour. Hours are matched only if MASKn.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0] Minute

The alarm minute. Minutes are matched only if MASKn.SEL is greater than 1.

Bits 5:0 – SECOND[5:0] Second

The alarm second. Seconds are matched only if MASKn.SEL is greater than 0.

21.12.12 Alarm n Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK
Offset: 0x24 + n*0x08 [n=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEL[2:0] Alarm Mask Selection

These bits define which bit groups of Alarm n are valid.

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years
0x7	-	Reserved

21.12.13 General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose

These bits are for user-defined general purpose use, see *General Purpose Registers* from Related Links.

Related Links

[21.6.8.4. General Purpose Registers](#)

21.12.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: Enable-Protected

Bit	31	30	29	28	27	26	25	24
					DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – DEBNCn Debounce Enable of Tamper Input INn [n=0..3]

Note: Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..3]

Note: Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7 – INnACT Tamper Channel n Action [n=0..3]

These bits determine the action taken by Tamper Channel n.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

21.12.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPIDn Tamper on Channel n Detected [n=0..3]

Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

21.12.16 Backup0

Name: BKUP0
Offset: 0x80
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0] Backup

These bits are user-defined for general purpose use in the Backup domain.

22. Direct Memory Access Controller (DMAC)

22.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter (see *DMAC Block Diagram* in the *Block Diagram* from Related Links). The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to SRAM.

All buses are AHB Manager interfaces except for the AHB/APB Bridge bus, which is an APB Subordinate interface.

Burst transfer options, buffered active channel to pre-fetch descriptors and advance quality of service features ensure low-latency transfers for high-speed peripherals or high-speed operations.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

Note: Traditional Direct Memory Access Controller (DMAC) documentation uses the terminology “Master” and “Slave”. The equivalent Microchip terminology used in this document is “Host” and “Client” respectively.

Related Links

[22.3. Block Diagram](#)

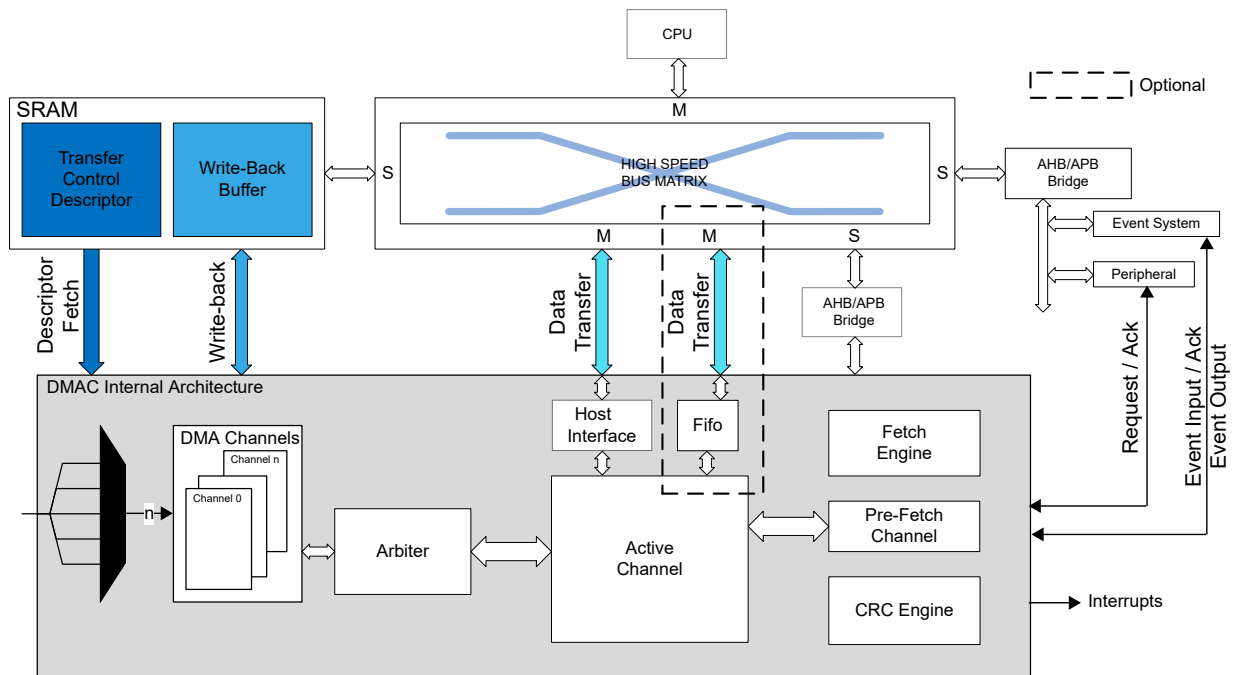
22.2 Features

- Data transfer from:
 - Peripheral to peripheral
 - Peripheral to memory
 - Memory to peripheral
 - Memory to memory
- Transfer trigger sources

- Software
 - Events from Event System
 - Dedicated requests from peripherals
- SRAM based transfer descriptors
 - Single transfer using one descriptor
 - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16 channels
 - Enable 16 independent transfers
 - Automatic descriptor fetch for each channel
 - Suspend/resume operation support for each channel
- Flexible arbitration scheme
 - 4 configurable priority levels for each channel
 - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
 - Static
 - Configurable increment scheme
- Optional interrupt generation
 - On block transfer complete
 - On error detection
 - On channel suspend
- 8 event inputs
 - One event input for each of the 8 least significant DMA channels
 - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
 - Can be selected to suspend or resume channel operation
- 4 event outputs
 - One output event for each of the 4 least significant DMA channels
 - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
 - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
 - CRC-16 (CRC-CCITT)
 - CRC-32 (IEEE® 802.3)

22.3 Block Diagram

Figure 22-1. DMAC Block Diagram



22.4 Signal Description

Not applicable.

22.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

22.5.1 I/O Lines

Not applicable.

22.5.2 Power Management

The DMAC will continue to operate in any Sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes. On hardware or software Reset, all registers are set to their Reset value.

22.5.3 DMA

Not applicable.

22.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

10.2. Nested Vector Interrupt Controller (NVIC)

22.5.5 Events

The events are connected to the event system.

22.5.6 Debug Operation

When the CPU is halted in Debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. See *DBGCTRL* from Related Links.

Related Links

[22.8.6. DBGCTRL](#)

22.5.7 Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

22.5.8 Analog Connections

Not applicable.

22.6 Functional Description

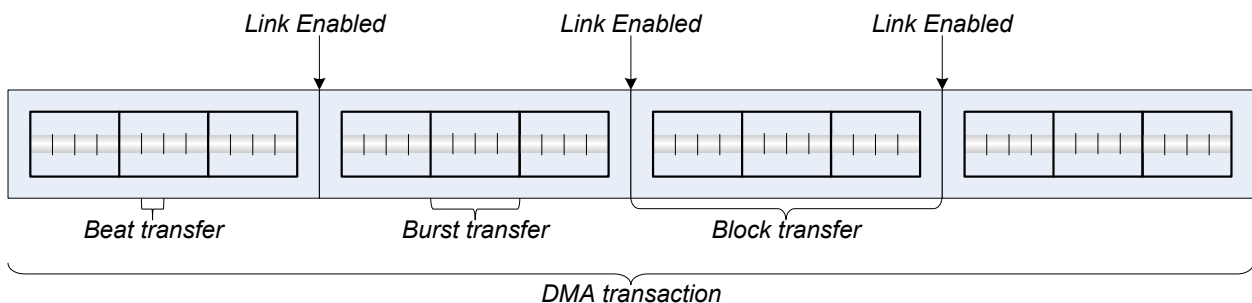
22.6.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

22.6.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

Figure 22-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer must be carried out by the DMAC, and it must remain in SRAM (see *Transfer Descriptors* from Related Links).

The figure above shows several block transfers linked together, which are called linked descriptors (see *Linked Descriptors* from Related Links).

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, depending on the configuration, the DMA channel will either be suspended or disabled.

Related Links

[22.6.2.3. Transfer Descriptors](#)

[22.6.3.1. Linked Descriptors](#)

22.6.1.2 CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. See *CRC Operation* from Related Links.

Related Links

[22.6.3.8. CRC Operation](#)

22.6.2 Basic Operation

22.6.2.1 Initialization

DMAC Initialization

Before DMAC is enabled, it must be configured as defined below:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register.
- The SRAM address of where the write-back section must be located must be written to the Write-Back Memory Base Address (WRBADDR) register.
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

DMA Channel Initialization

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as defined below:

- DMA Channel Configuration:
 - The channel number of the DMA channel to configure must be written to the Channel Control A (CHCTRLA) register.
 - Trigger action must be selected by writing the Trigger Action bit field in the Channel Control A (CHCTRLA.TRIGACT) register.
 - Trigger source must be selected by writing the Trigger Source bit field in the Channel Control A (CHCTRLA.TRIGSRC) register.
- Transfer Descriptor

- The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control (BTCTRL.BEATSIZE) register.
- The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control (BTCTRL.VALID) register.
- Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register.
- Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register.
- Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register.

CRC Calculation

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as described below:

- The CRC input source must be selected by writing the CRC Input Source bit group in the CRC Control (CRCCTRL.CRCSRC) register.
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control (CRCCTRL.CRCPOLY) register.
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control (CRCCTRL.CRCBEATSIZE) register.

Register Properties

The following DMAC registers are enable-protected, that is, they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- The Descriptor Base Memory Address (BASEADDR) register
- The Write-Back Memory Base Address (WRBADDR) register

The following DMAC bit is enable-protected, that is, it can only be written when the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CRCCTRL.CRCSRC=0):

- The Software Reset bit in the Control (CTRL.SWRST) register

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- The Channel Software Reset bit in the Channel Control A (CHCTRLA.SWRST) register

The following CRC registers are enable-protected, that is, they can only be written when the CRC is disabled (CRCCTRL.CRCSRC=0):

- The CRC Control (CRCCTRL) register
- CRC Checksum (CRCCHKSUM) register

Enable-protection is denoted by the 'Enable-Protected' property in the register description.

22.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to the CTRL.DMAENABLE bit.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after the corresponding channel ID to the channel is configured. A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLAn.SWRST), after writing the corresponding channel ID to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the Reset to take effect.

22.6.2.3 Transfer Descriptors

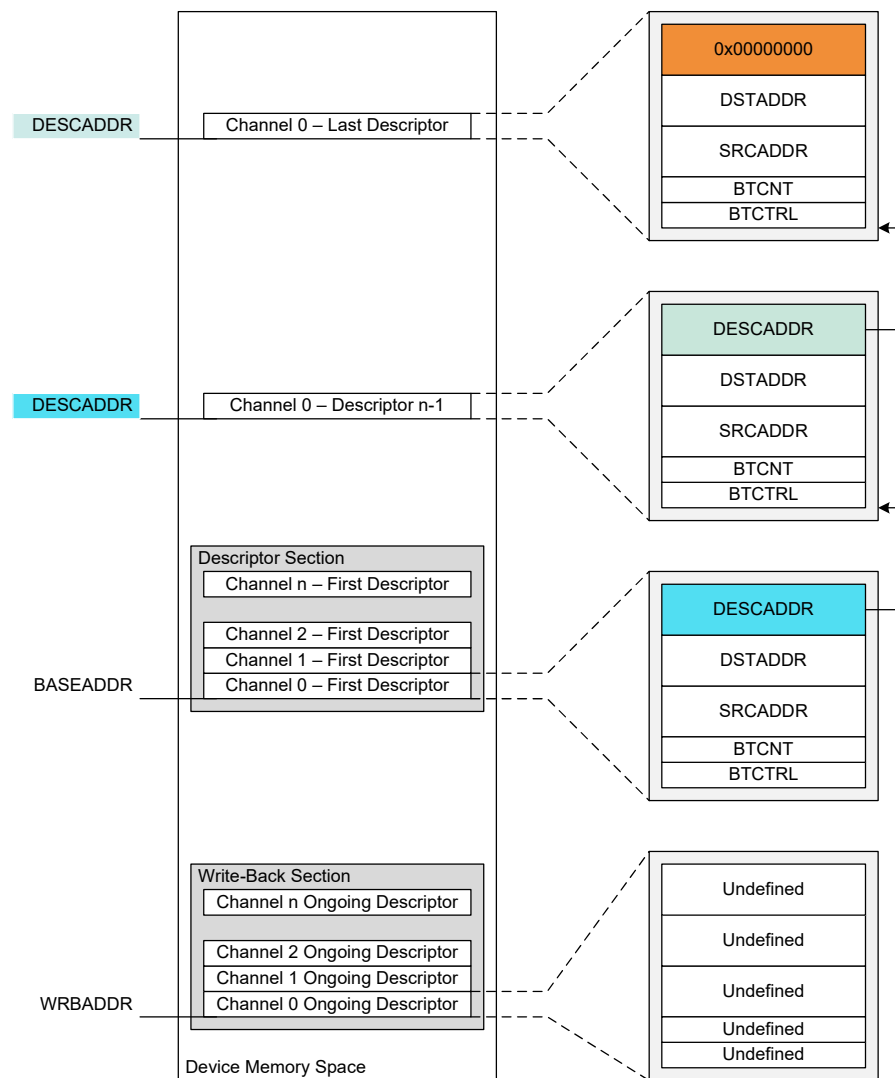
The transfer descriptors, together with the channel configurations, decide how a block transfer must be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one) and receives a transfer trigger, its first transfer descriptor must be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel '0' (see the following figure). All first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number (see *Linked Descriptors* from Related Links).

The write-back memory section is where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel '0'. All ongoing transfer descriptors are stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel '0' (see *Linked Descriptors* from Related Links).

Figure 22-3. Memory Sections



The size of the descriptor and write-back memory sections are dependent on the number of the most significant enabled DMA channel m , as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to use the less significant DMA channels, if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share a memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor.

Related Links

[22.6.3.1. Linked Descriptors](#)

22.6.2.4 Arbitration

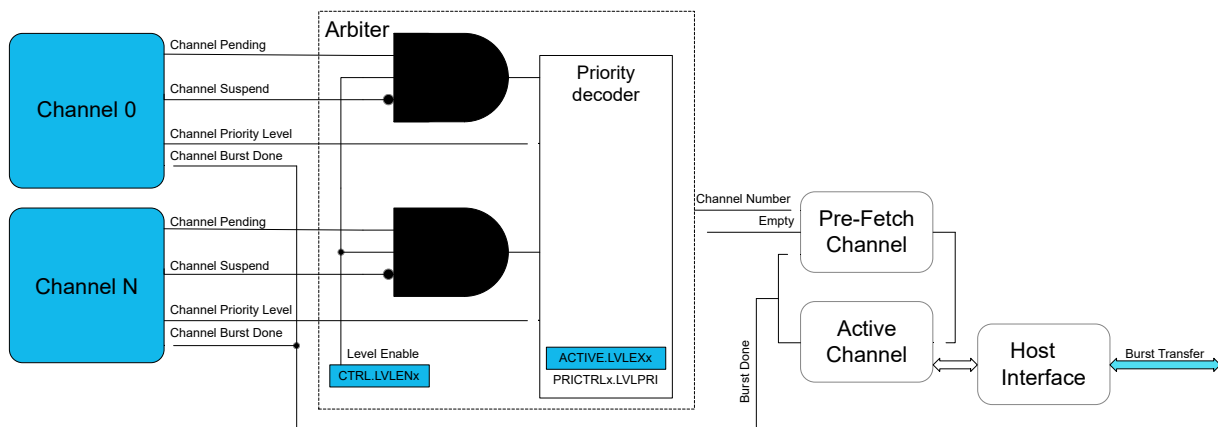
If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The next transfer descriptor will be fetched from SRAM memory and stored internally in the Pre-Fetch Channel. The active channel is the DMA channel being granted access to perform its next burst transfer. When the Active Channel has completed a burst transfer, the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel and a new burst will take place.

When the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel, the corresponding PENDCH.PENDCHx will be cleared. In the same way, depending on trigger action settings and if the upcoming burst transfer is the first for the transfer request or not, the corresponding Busy Channel x bit in the Busy Channels register (BUSYCH.BUSYCHx), will either be set or remain '1'. When the channel has performed its granted burst transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is set to wait for a new transfer trigger, suspended or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. The status will also be indicated in CHINTFLAGn.SUSP. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 22-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

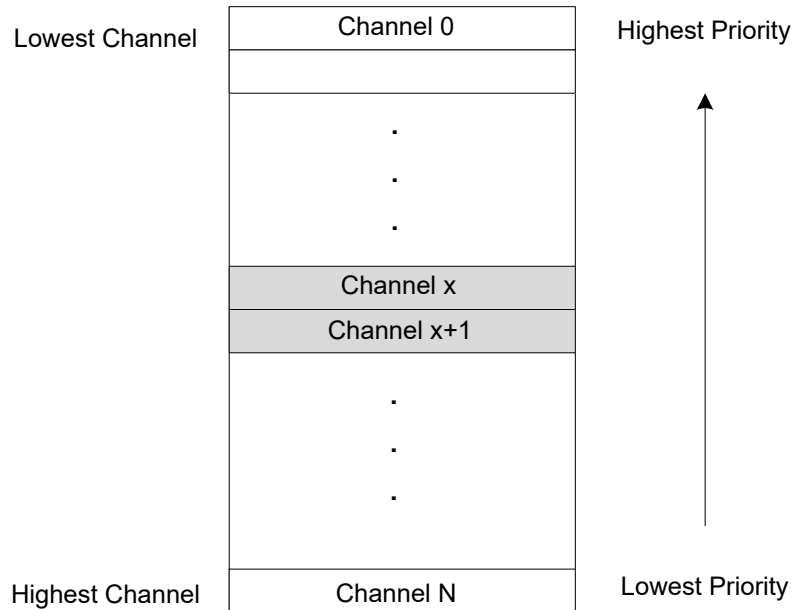
Each DMA channel supports up to 4-level priority scheme.

The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Priority Level register (CHPRILVL.PRILVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority

level number. A priority level is enabled by writing the Priority Level x Enable bit in the Control register (CTRL.LVLENx) to '1', for the corresponding level.

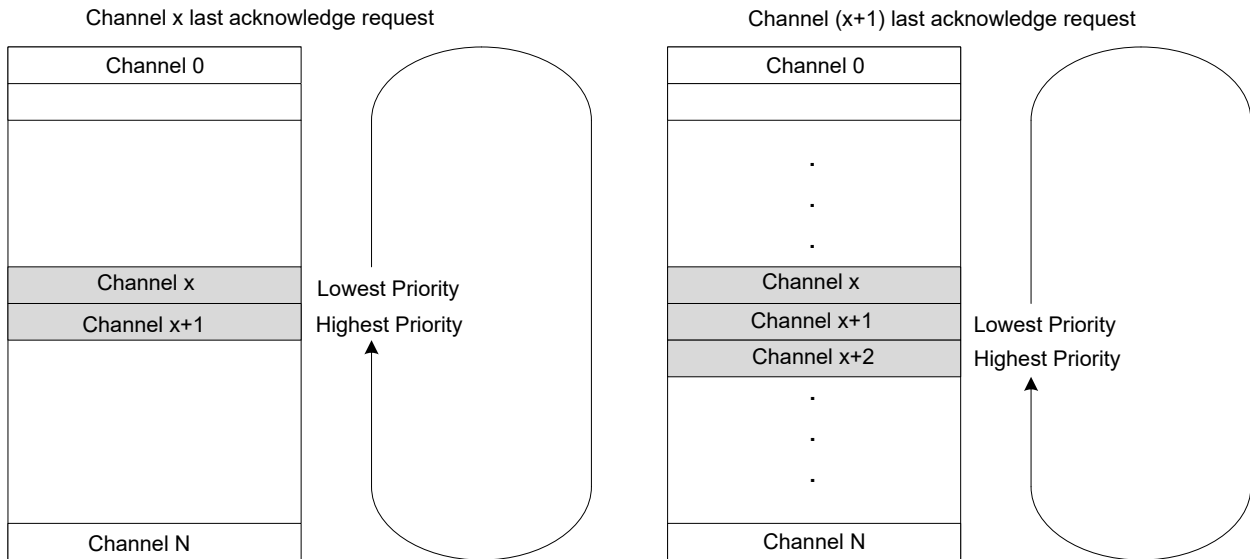
Within each priority level, the DMAC's arbiter can be configured to prioritize statically or dynamically. For the arbiter to perform static arbitration within a priority level, the Level X Round-Robin Scheduling Enable bit in the Priority Control x register (PRICTRL0.RRLVLENx) has to be written to '0'. When static arbitration is enabled (PRICTRL0.RRLVLENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown in the following figure. When using the static scheme, there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

Figure 22-5. Static Priority Scheduling



The dynamic arbitration scheme in the DMAC is round-robin. Round-robin arbitration is enabled by writing PRICTRL0.RRLVLEN to '1', for a given priority level x. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in the following figure. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.

Figure 22-6. Dynamic (Round-Robin) Priority Scheduling



22.6.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (see *DMAC Block Diagram* in the *Block Diagram* from Related Links) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated (see *Addressing* from the Related Links).

The arbitration procedure is performed after each transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

Related Links

[22.6.2.7. Addressing](#)

[22.3. Block Diagram](#)

22.6.2.6 Transfer Triggers and Actions

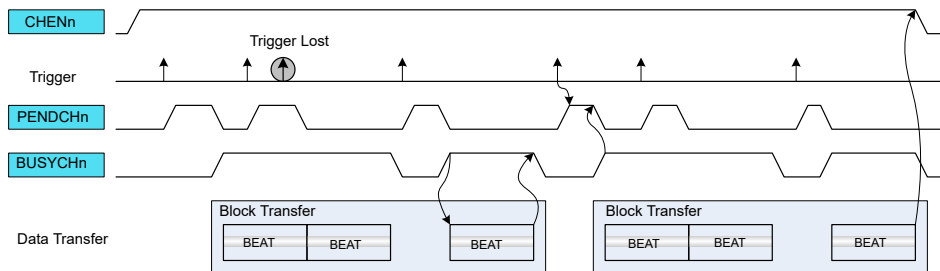
A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel n Control A (CHCTRLAn.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel n Control A register (CHCTRLAn.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. As long as the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a burst transfer (CHCTRLAn.TRIGACT=0x2) or transaction transfer (CHCTRLAn.TRIGACT=0x3) instead of a block transfer (CHCTRLAn.TRIGACT=0x0).

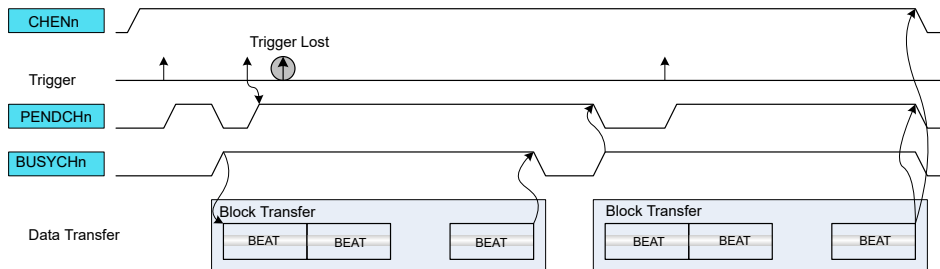
The following figure shows an example where triggers are used with two linked block descriptors.

Figure 22-7. Trigger Action and Transfers

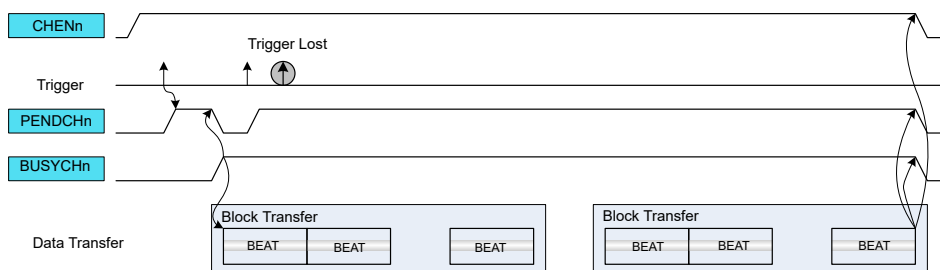
Beat Trigger Action



Block Trigger Action



Transaction Trigger Action



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUSn.PEND=1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel n Status register (CHSTATUSn.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

22.6.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (DSTADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1) \cdot 2^{\text{STEPSIZE}}$$

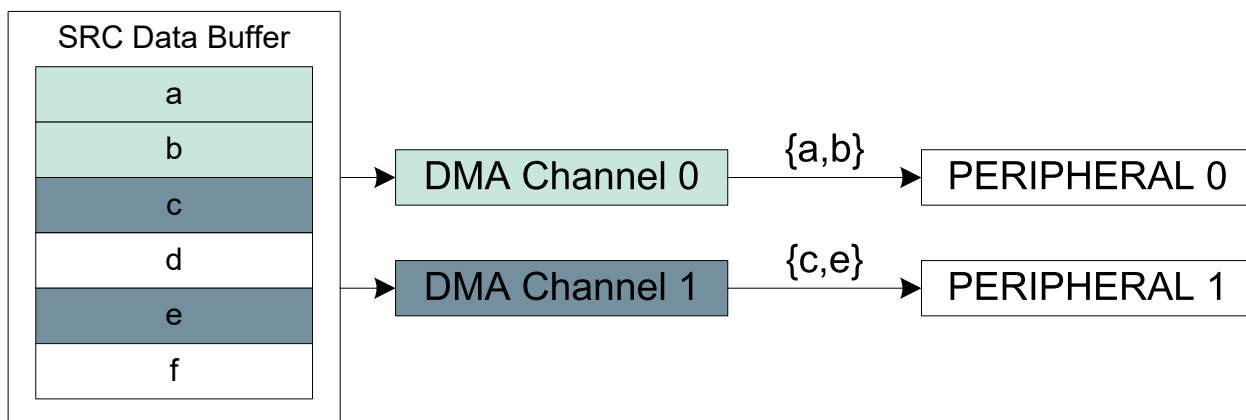
If BTCTRL.STEPSEL=0:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1)$$

- SRCADDR_{START} is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0×1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 22-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

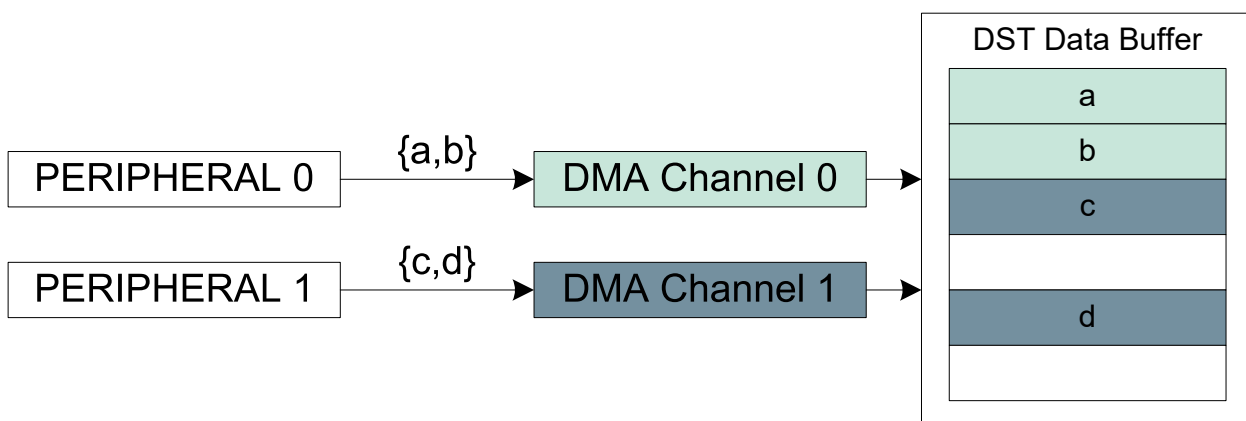
When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \cdot (BEATSIZE + 1) \cdot 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \cdot (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0, and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Figure 22-9. Destination Address Increment



22.6.2.8 Internal FIFO

To improve the bandwidth, the DMAC can support FIFO operation. When single-beat burst configuration is selected (CHCTRALx.BURSTLEN = SINGLE), the channel waits until the FIFO can transmit or accept a single beat transfer before it requests a bus access to write to the destination address. In all other cases, the channel waits until the FIFO threshold is reached before it requests a bus access to write to the destination address. The threshold is configurable and can be set by writing the THRESHOLD bits in the Channel x Control A register.

If the DMAC completes the read operations before the threshold is reached, the write to the destination is automatically enabled. If the FIFO is empty and the read from source is ongoing, the DMA will wait again until the FIFO threshold is reached before it requests a bus access to write the destination.

22.6.2.9 Error Handling

If a bus error is received from an AHB subordinate during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

22.6.3 Additional Features

22.6.3.1 Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consists of several block transfers it is done with the help of linked descriptors.

Memory Sections illustrates how linked descriptors work (see *Memory Sections* figure in the *Transfer Descriptors* from Related Links). When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor, which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR = 0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM (see *Data Transmission* from Related Links).

Related Links

[22.6.2.3. Transfer Descriptors](#)

[22.6.2.5. Data Transmission](#)

22.6.3.1.1 Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR = 0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

22.6.3.1.2 Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.

4. Configure the new descriptor:
 - Set the next descriptor address (DESCADDR)
 - Set the destination address (DESCADDR)
 - Set the source address (SRCADDR)
 - Configure the block transfer control (BTCTRL) including
 - Optionally enable the suspend block action
 - Set the descriptor VALID bit
5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the write-back memory.
 - If the DMA has not already fetched the descriptor that requires changes (in other words, DESCADDR is wrong):
 - Update the DESCADDR location of the descriptor from the list
 - Optionally clear the suspend block action
 - Set the descriptor VALID bit to '1'
 - Optionally enable the Resume Software command
 - If the DMA is executing the same descriptor as the one that requires changes:
 - Set the Channel Suspend Software command and wait for the suspend interrupt
 - Update the next descriptor address (DESCADDR) in the write-back memory
 - Clear the interrupt sources and set the Resume Software command
 - Update the DESCADDR location of the descriptor from the list
 - Optionally clear the suspend block action
 - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

22.6.3.1.3 Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
 - a. Set the descriptor A VALID bit to '0'.
 - b. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B (see *DESCADDR* in the *DMAC Register Summary* from Related Links).
 - c. Set the DESCADDR value of descriptor C to point to descriptor B (see *DESCADDR* in the *DMAC Register Summary* from Related Links).
 - d. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
 - a. Apply the software suspend command to the channel and
 - b. Perform steps 2.1 through 2.4.
 - c. Apply the software resume command to the channel.

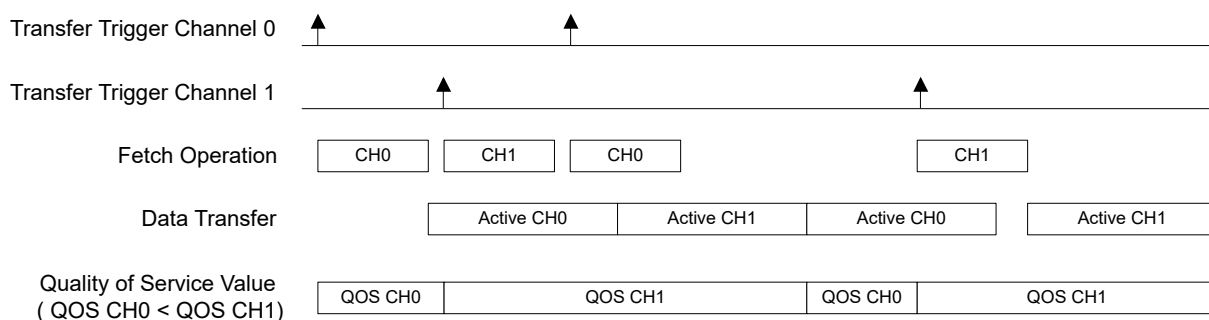
Related Links

[22.9. DMAC Register Summary \(SRAM\)](#)

22.6.3.2 Transfer Quality of Service

Each priority level group has dedicated quality of service settings. The setting can be written in the corresponding Quality of Service bit group in the Priority Control x register (PRICTRL0.QOSn).

Figure 22-10. Quality of Service



When a channel is stored in the Pre-Fetch or Active Channel, the corresponding PRICTRLx.QOS bits value is stored in the respective channel. As shown in Quality of Service, the DMAC will select the highest QOS value between Active and Pre-Fetch channels. This value will apply to all DMAC buses.

22.6.3.3 Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register (CHASTATUS.FERR) will be set.

Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors (see *Transfer Descriptors* from Related Links).

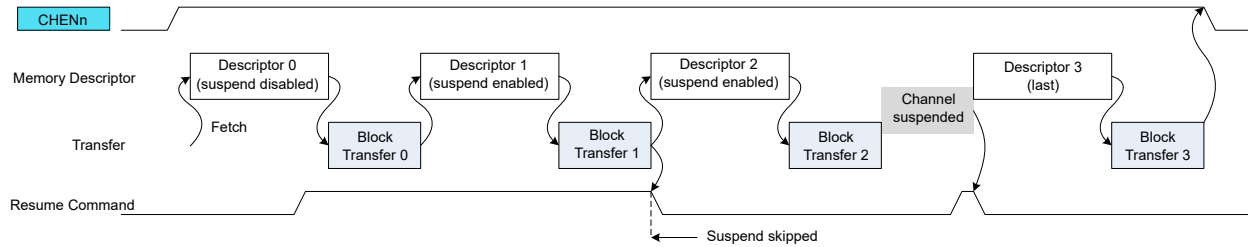
Related Links

[22.6.2.3. Transfer Descriptors](#)

22.6.3.4 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 22-11. Channel Suspend/Resume Operation



22.6.3.5 Event Input Actions

The event input actions are available only on the least significant DMA channels. For more details on channels with event input support (see *Event System (EVSYS)* from Related Links).

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Event Control register (CHEVCTRL.EVIE) must be written to '1'. See *Events* from Related Links.

Table 22-1. Event Input Action

Action	CHEVCTRL.EVACT	CHCTRLA.TRIGSRC
None	NOACT	—
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	Any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	
Increase priority	INCPRI	

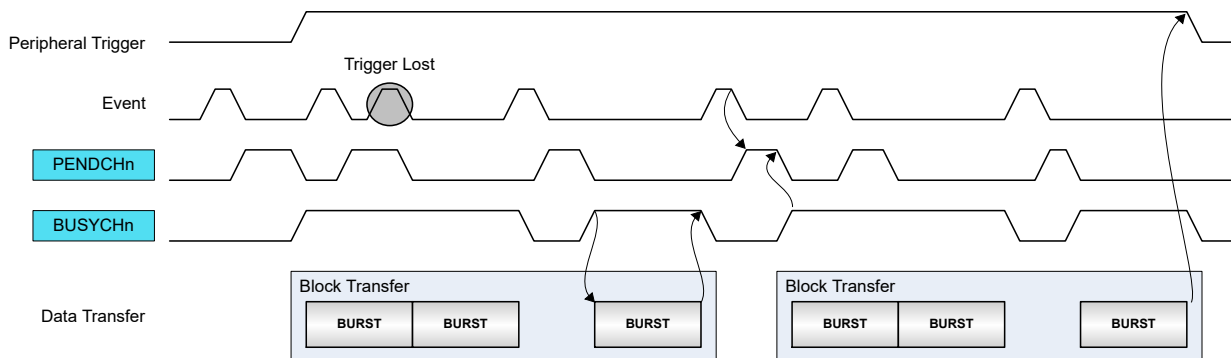
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The following figure shows an example where beat transfers are enabled by internal events.

Figure 22-12. Burst Event Trigger Action



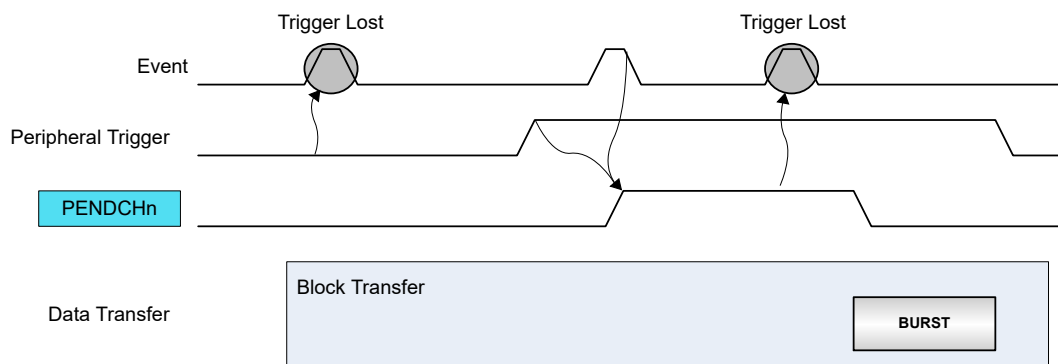
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, for example, for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (in other words, the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA waits for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The following figure shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 22-13. Periodic Event with Burst Peripheral Triggers



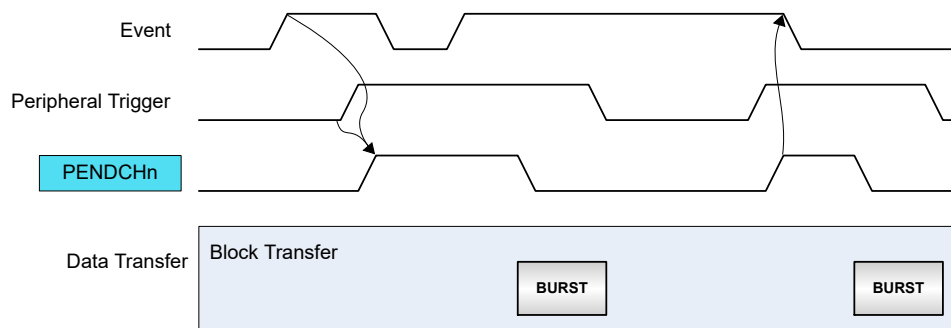
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The following figure shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 22-14. Conditional Event with Burst Peripheral Triggers



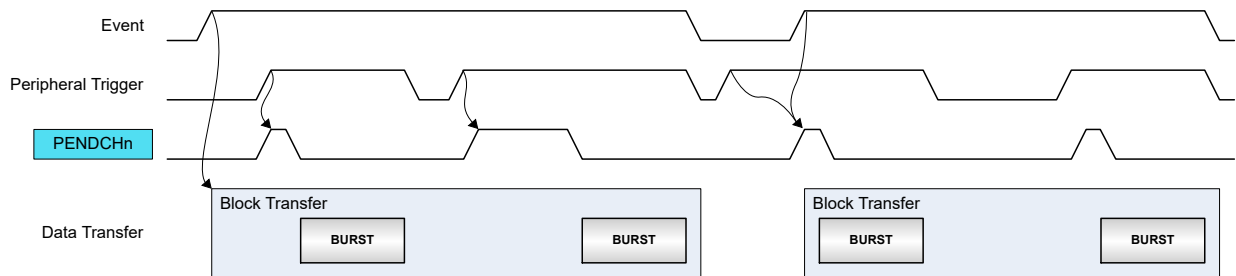
Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The following figure shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 22-15. Conditional Block Transfer with Burst Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For more details on Channel Suspend (see *Channel Suspend* from Related Links).

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. See *Channel Suspend* from Related Links.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

Increase priority

This event can be used to increase a channel priority and to request higher quality of service (QOS), when critical transfers must be done. When the event is detected, the channel will have the highest priority and the output Quality of Service value is internally forced to the maximum value. The event is acknowledged when the trigger action execution is completed. When acknowledged, the channel will recover its initial priority level and quality of service settings.

Related Links

[22.6.3.3. Channel Suspend](#)

[22.6.6. Events](#)

[28. Event System \(EVSYS\)](#)

22.6.3.6 Event Output Selection

The event output selections are available only for channels supporting event outputs.

The Channel Event Output Enable can be set in the corresponding Channel n Event Control register (CHEVCTRL.EVOE). The Event Output Mode bits in the Channel n Event Control register (CHEVCTRL.EVOMODE) selects the event type the channel will generate.

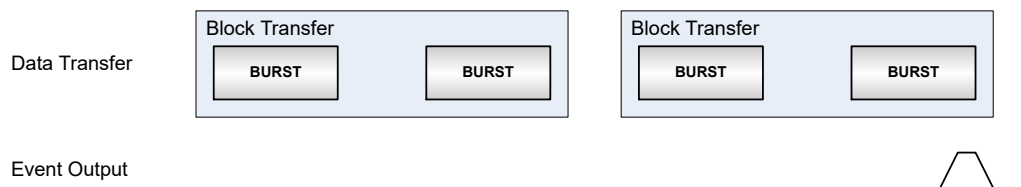
The transfer events (CHEVCTRL.EVOMODE = DEFAULT) are strobe events and their duration is one CLK_DMACH_AHB clock period. The transfer event type selection is available in each Descriptor Block Control location (BTCTRL.EVOSEL). Block or burst event output generation is supported.

The trigger action event (CHEVCTRL.EVOMODE = TRIGACT) is a level, active while the trigger action execution is not completed.

Block Event Output

When the block event output is selected, an event strobe is generated when the block transfer is completed. The pulse width of a block event output from a channel is one AHB clock cycle. It is also possible to use this event type to generate an event when the transaction is complete. For this type of application, the block event selection must be set in the last transfer descriptor only, as shown below.

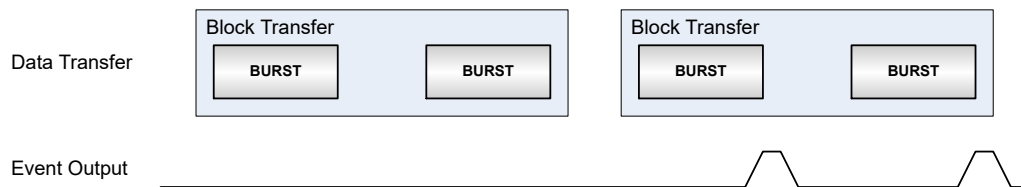
Figure 22-16. Block Event Output Generation



Burst Event Output

When the burst event output is selected, an event strobe is generated when each burst transfer within the corresponding block is completed. The pulse width of a burst event output from a channel is one AHB clock cycle. The figure below shows an example where the burst event output is set in the second descriptor of a linked list.

Figure 22-17. Burst Event Output Generation

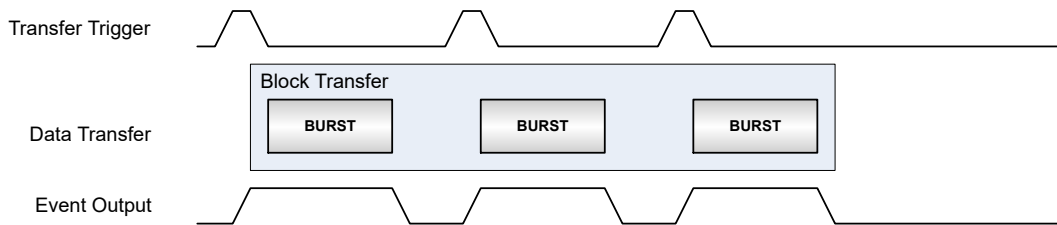


Trigger Action Event Output

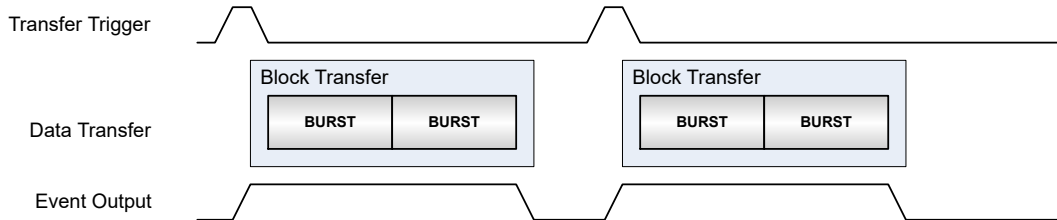
When the trigger action event output is selected, an event level is generated. The event output is set when the transfer trigger occurred, and cleared when the corresponding trigger action is completed. The following figure shows an example for each trigger action type.

Figure 22-18. Trigger Action Event Output Generation

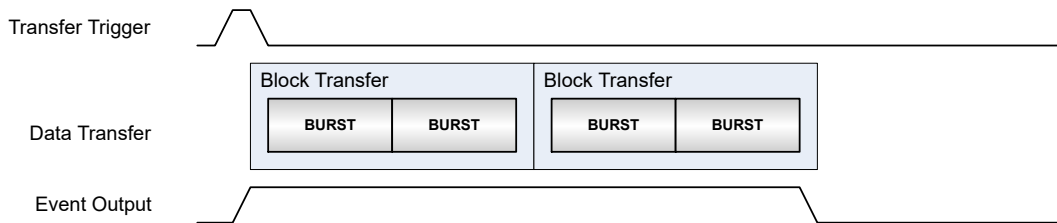
Burst Trigger Action Event Output



Block Trigger Action Event Output



Transaction Trigger Action Event Output



22.6.3.7 Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

22.6.3.8 CRC Operation

A Cyclic Redundancy Check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will, then, detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

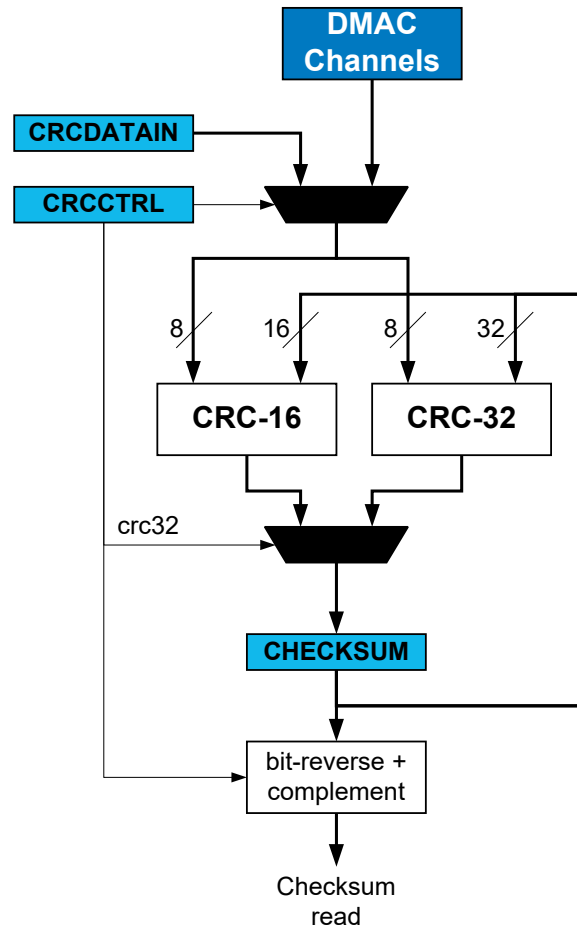
The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is $\leq n$ bits in length, and will detect the fraction $1-2^{-n}$ of all longer error bursts.

- CRC-16:
 - Polynomial: $x^{16} + x^{12} + x^5 + 1$
 - Hex value: 0x1021
- CRC-32:
 - Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
 - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When the CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in the following figure.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY); the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as a data source for the CRC engine, the DMA channel beat size setting will be used. When used with the APB bus interface, the application must select the CRC Beat Size bit field of the CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16- or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte-by-byte manner.

Figure 22-19. CRC Generator Block Diagram



CRC on DMA Data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC Using the I/O Interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register, the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when the CRCBUSY flag is not set.

22.6.3.9 Memory CRC Generation

When enabled, it is possible to automatically calculate a memory block checksum. When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT) stored in the Block Transfer Destination Address register (DSTADDR). The DMA read and calculate the checksum over the data from the source address. When the checksum calculation is completed, the CRC value is stored in the CRC Checksum register (CRCCHKSUM), the Transfer Complete interrupt flag is set (CHINTFLAGn.TCMPL) and optional interrupt is generated.

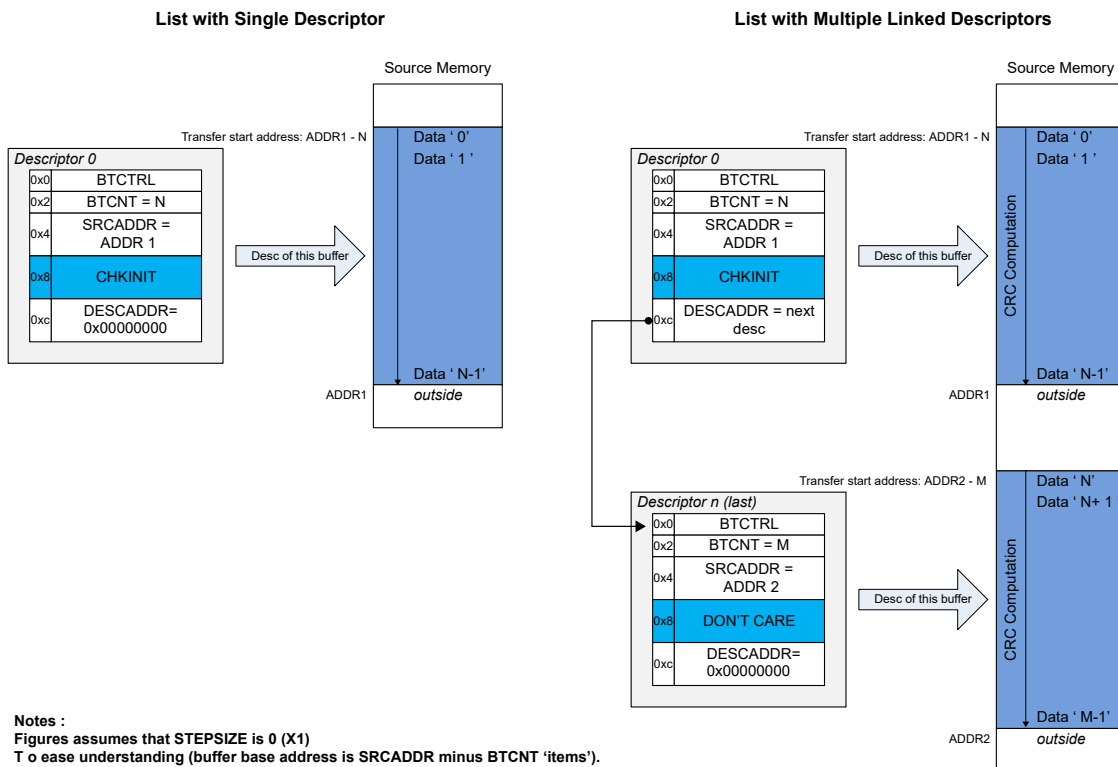
If the linked descriptor is in the list (DESCADDR !=0), the DMA will fetch the next descriptor and CRC calculation continues as described above. When the last list descriptor is executed, the channel is automatically disabled.

In order to enable the memory CRC generation, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor:
 - Set the next descriptor address (DESCADDR)
 - Set the destination address with the initial checksum value (DSTADDR = CHKINIT) in the first descriptor in a list
 - Set the transfer source address (SRCADDR)
 - Set the block transfer count (BTCNT)
 - Set the memory CRC generation operation mode (CRCCTRL.CRCMODE = CRCGEN)
 - Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLn.ENABLE)

The figure below shows the CRC computation slots and descriptor configuration when single or linked-descriptors transfers are enabled.

Figure 22-20. CRC Computation with Single Linked Transfers



22.6.3.10 Memory CRC Monitor

When enabled, it is possible to continuously check a memory block data integrity by calculating and checking the CRC checksum. The expected CRC checksum value must be located in the last memory block location, as shown in the table below:

CRCCTRL.CRCPOLY	CRCCTRL.CRCBEATSIZE	Last Memory Block Byte Locations Value (MSB Byte First)	CHECKSUM Result
CRC-16	Byte	Expected CRC[7:0]	0x00000000
	Half-word	Expected CRC[15:8]	
	Word	0x00	
		0x00	
CRC-32	Byte	Expected CRC[31:24]	CRC Magic Number (0x2144DF1C)
	Half-word	Expected CRC[23:16]	
	Word	Expected CRC[15:8]	
		Expected CRC[7:0]	

When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT), stored in the DSTADDR location of the first descriptor. The DMA read and calculate the checksum over the entire data from the source address. When the checksum calculation is completed the DMA read the last beat from the memory, the calculated CRC value from the CRC Checksum register is compared to zero or CRC magic number, depending on CRC polynomial selection.

If the CHECKSUM does not match the comparison value the DMA channel is disabled, and both the CRC Error bit in the Channel n Status register (CHSTATUSn.CRCERR) and Transfer Error interrupt flag (CHINTFLAGn.TERR) are set. If enabled, the Transfer Error interrupt is generated.

If the calculated checksum value matches the compare value, the Transfer Complete interrupt flag (CHINTFLAGn.TCML) is set, optional interrupt is generated and the DMA will perform the following actions, depending on the descriptor list settings:

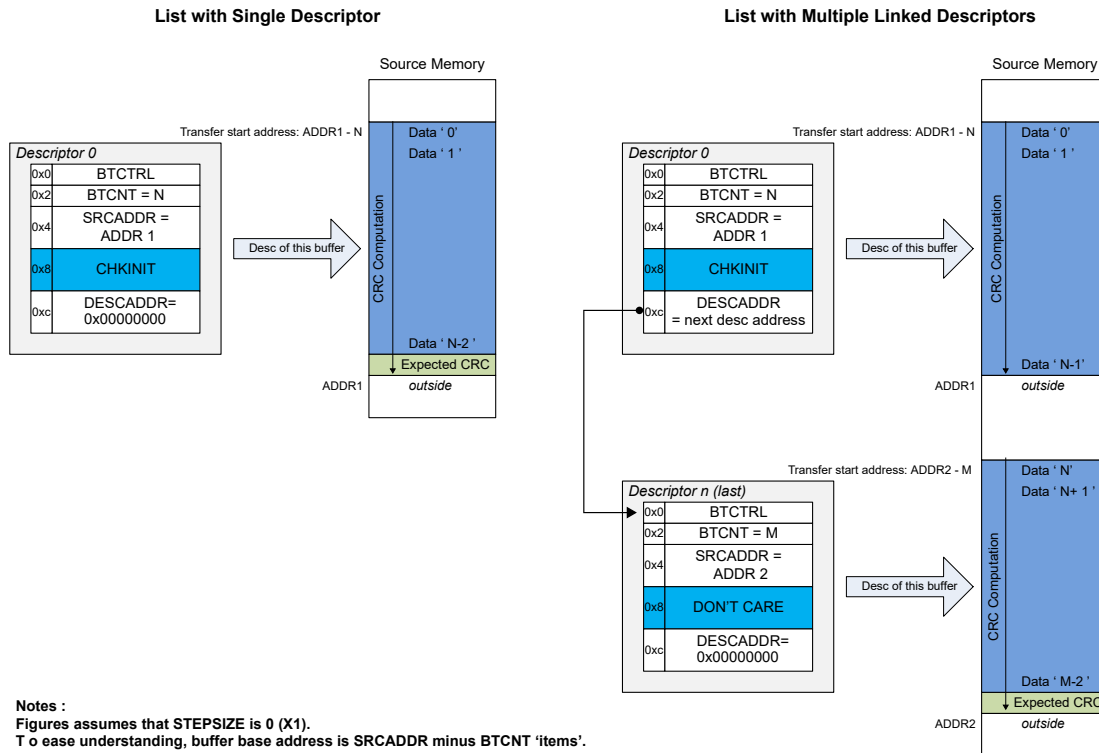
- If the list has only one descriptor, the DMA will re-fetch the descriptor
- If the current descriptor is the last descriptor from the list, the DMA will fetch the first descriptor from the list

When the fetch is completed, the DMA restarts the operations described above when new triggers are detected.

In order to enable the memory CRC monitor, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor
 - Set the next descriptor address (DESCADDR)
 - In the first list descriptor, set the destination address with the initial checksum value (DSTADDR = CHKINIT)
 - Set the transfer source address (SRCADDR)
 - Set the block transfer count (BTCNT)
 - Set the memory CRC monitor operation mode (CRCCTRL.CRCMODE = CRCMON)
 - Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE)

Figure 22-21. CRC Computation and Check with Single or Linked Transfers



22.6.4 DMA Operation

Not applicable.

22.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. See *Data Transmission* from Related Links.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. See *Error Handling* from Related Links.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. See *Channel Suspend* and *Data Transmission* from Related Links.

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear Interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register

to determine which Interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective Interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

[22.6.2.5. Data Transmission](#)

[22.6.3.3. Channel Suspend](#)

[10.2. Nested Vector Interrupt Controller \(NVIC\)](#)

22.6.6 Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. See *Event Output Selection* from Related Links.

Setting the Channel Event Output Enable bit (CHEVCTRLx.EVOE = 1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHEVCTRLx.EVOE = 0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition
- Increase Priority (INCPRI): increase channel priority

Setting the Channel Event Input Enable bit (CHEVCTRLx.EVIE = 1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. See *Event Input Actions* from Related Links for more details on event input actions.

Note: Event input and outputs are not available for every channel. See *Features* from Related Links.

Related Links

[22.6.3.6. Event Output Selection](#)

[22.6.3.5. Event Input Actions](#)

[22.2. Features](#)

22.6.7 Sleep Mode Operation

22.6.8 Synchronization

Not applicable.

22.7 DMAC Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0							DMAENABLE	SWRST
		15:8					LVLENx3	LVLENx2	LVLENx1	LVLENx0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
		15:8	CRCMODE[1:0]		CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
		15:8	CRCDATAIN[15:8]							
		23:16	CRCDATAIN[23:16]							
		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
		15:8	CRCCHKSUM[15:8]							
		23:16	CRCCHKSUM[23:16]							
		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0						CRCERR	CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E ... 0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIGn[7:0]							
		15:8	SWTRIGn[15:8]							
		23:16								
		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0	QOS00[1:0]		LVLPRIO[4:0]				
		15:8	RRLVLEN1	QOS01[1:0]		LVLPRIO[4:0]				
		23:16	RRLVLEN2	QOS02[1:0]		LVLPRIO[4:0]				
		31:24	RRLVLEN3	QOS03[1:0]		LVLPRIO[4:0]				
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0				ID[4:0]				
		15:8	PEND	BUSY	FERR	CRCERR		SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0								
		15:8								
		23:16								
		31:24								
0x28	BUSYCH	7:0								
		15:8								
		23:16								
		31:24								
0x2C	PENDCH	7:0								
		15:8								
		23:16								
		31:24								
0x30	ACTIVE	7:0								
		15:8	ABUSY							
		23:16	BTCNT[7:0]							
		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
		15:8	BASEADDR[15:8]							
		23:16	BASEADDR[23:16]							
		31:24	BASEADDR[31:24]							
0x38	WRBADDR	7:0	WRBADDR[7:0]							
		15:8	WRBADDR[15:8]							
		23:16	WRBADDR[23:16]							
		31:24	WRBADDR[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C ... 0x3F	Reserved									
0x40	CHCTRLA0	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x44	CHCTRLB0	7:0							CMD[1:0]	
0x45	CHPRILV0	7:0								
0x46	CHEVCTRL0	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x47 ... 0x4B	Reserved									
0x4C	CHINTENCLR0	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET0	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG0	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS0	7:0					CRCERR	FERR	BUSY	PEND
0x50	CHCTRLA1	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x54	CHCTRLB1	7:0							CMD[1:0]	
0x55	CHPRILV1	7:0								
0x56	CHEVCTRL1	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x57 ... 0x5B	Reserved									
0x5C	CHINTENCLR1	7:0						SUSP	TCMPL	TERR
0x5D	CHINTENSET1	7:0						SUSP	TCMPL	TERR
0x5E	CHINTFLAG1	7:0						SUSP	TCMPL	TERR
0x5F	CHSTATUS1	7:0					CRCERR	FERR	BUSY	PEND
0x60	CHCTRLA2	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x64	CHCTRLB2	7:0							CMD[1:0]	
0x65	CHPRILV2	7:0								
0x66	CHEVCTRL2	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x67 ... 0x6B	Reserved									
0x6C	CHINTENCLR2	7:0						SUSP	TCMPL	TERR
0x6D	CHINTENSET2	7:0						SUSP	TCMPL	TERR
0x6E	CHINTFLAG2	7:0						SUSP	TCMPL	TERR
0x6F	CHSTATUS2	7:0					CRCERR	FERR	BUSY	PEND
0x70	CHCTRLA3	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x74	CHCTRLB3	7:0							CMD[1:0]	
0x75	CHPRILV3	7:0								
0x76	CHEVCTRL3	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x77 ... 0x7B	Reserved									
0x7C	CHINTENCLR3	7:0						SUSP	TCMPL	TERR
0x7D	CHINTENSET3	7:0						SUSP	TCMPL	TERR
0x7E	CHINTFLAG3	7:0						SUSP	TCMPL	TERR
0x7F	CHSTATUS3	7:0					CRCERR	FERR	BUSY	PEND

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80	CHCTRLA4	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x84	CHCTRLB4	7:0							CMD[1:0]	
0x85	CHPRILVL4	7:0								
0x86	CHEVCTRL4	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x87	Reserved									
...										
0x8B										
0x8C	CHINTENCLR4	7:0						SUSP	TCMPL	TERR
0x8D	CHINTENSET4	7:0						SUSP	TCMPL	TERR
0x8E	CHINTFLAG4	7:0						SUSP	TCMPL	TERR
0x8F	CHSTATUS4	7:0					CRCERR	FERR	BUSY	PEND
0x90	CHCTRLA5	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x94	CHCTRLB5	7:0							CMD[1:0]	
0x95	CHPRILVL5	7:0								
0x96	CHEVCTRL5	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x97	Reserved									
...										
0x9B										
0x9C	CHINTENCLR5	7:0						SUSP	TCMPL	TERR
0x9D	CHINTENSET5	7:0						SUSP	TCMPL	TERR
0x9E	CHINTFLAG5	7:0						SUSP	TCMPL	TERR
0x9F	CHSTATUS5	7:0					CRCERR	FERR	BUSY	PEND
0xA0	CHCTRLA6	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xA4	CHCTRLB6	7:0							CMD[1:0]	
0xA5	CHPRILVL6	7:0								
0xA6	CHEVCTRL6	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xA7	Reserved									
...										
0xAB										
0xAC	CHINTENCLR6	7:0						SUSP	TCMPL	TERR
0xAD	CHINTENSET6	7:0						SUSP	TCMPL	TERR
0xAE	CHINTFLAG6	7:0						SUSP	TCMPL	TERR
0xAF	CHSTATUS6	7:0					CRCERR	FERR	BUSY	PEND
0xB0	CHCTRLA7	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xB4	CHCTRLB7	7:0							CMD[1:0]	
0xB5	CHPRILVL7	7:0								
0xB6	CHEVCTRL7	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xB7	Reserved									
...										
0xBB										
0xBC	CHINTENCLR7	7:0						SUSP	TCMPL	TERR
0xBD	CHINTENSET7	7:0						SUSP	TCMPL	TERR
0xBE	CHINTFLAG7	7:0						SUSP	TCMPL	TERR
0xBF	CHSTATUS7	7:0					CRCERR	FERR	BUSY	PEND
0xC0	CHCTRLA8	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xC4	CHCTRLB8	7:0							CMD[1:0]	

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC5	CHPRILVL8	7:0								
0xC6	CHEVCTRL8	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xC7	...									
0xCB	Reserved									
0xCC	CHINTENCLR8	7:0						SUSP	TCMPL	TERR
0xCD	CHINTENSET8	7:0						SUSP	TCMPL	TERR
0xCE	CHINTFLAG8	7:0						SUSP	TCMPL	TERR
0xCF	CHSTATUS8	7:0					CRCERR	FERR	BUSY	PEND
0xD0	CHCTRLA9	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xD4	CHCTRLB9	7:0							CMD[1:0]	
0xD5	CHPRILVL9	7:0								
0xD6	CHEVCTRL9	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xD7	...									
0xDB	Reserved									
0xDC	CHINTENCLR9	7:0						SUSP	TCMPL	TERR
0xDD	CHINTENSET9	7:0						SUSP	TCMPL	TERR
0xDE	CHINTFLAG9	7:0						SUSP	TCMPL	TERR
0xDF	CHSTATUS9	7:0					CRCERR	FERR	BUSY	PEND
0xE0	CHCTRLA10	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xE4	CHCTRLB10	7:0							CMD[1:0]	
0xE5	CHPRILVL10	7:0								
0xE6	CHEVCTRL10	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xE7	...									
0xEB	Reserved									
0xEC	CHINTENCLR10	7:0						SUSP	TCMPL	TERR
0xED	CHINTENSET10	7:0						SUSP	TCMPL	TERR
0xEE	CHINTFLAG10	7:0						SUSP	TCMPL	TERR
0xEF	CHSTATUS10	7:0					CRCERR	FERR	BUSY	PEND
0xF0	CHCTRLA11	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0xF4	CHCTRLB11	7:0							CMD[1:0]	
0xF5	CHPRILVL11	7:0								
0xF6	CHEVCTRL11	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xF7	...									
0xFB	Reserved									
0xFC	CHINTENCLR11	7:0						SUSP	TCMPL	TERR
0xFD	CHINTENSET11	7:0						SUSP	TCMPL	TERR
0xFE	CHINTFLAG11	7:0						SUSP	TCMPL	TERR
0xFF	CHSTATUS11	7:0					CRCERR	FERR	BUSY	PEND
0x0100	CHCTRLA12	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x0104	CHCTRLB12	7:0							CMD[1:0]	
0x0105	CHPRILVL12	7:0								
0x0106	CHEVCTRL12	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0107	...									
0x010B	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010C	CHINTENCLR12	7:0						SUSP	TCMPL	TERR
0x010D	CHINTENSET12	7:0						SUSP	TCMPL	TERR
0x010E	CHINTFLAG12	7:0						SUSP	TCMPL	TERR
0x010F	CHSTATUS12	7:0					CRCERR	FERR	BUSY	PEND
0x0110	CHCTRLA13	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x0114	CHCTRLB13	7:0							CMD[1:0]	
0x0115	CHPRILVL13	7:0								
0x0116	CHEVCTRL13	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0117	Reserved									
...										
0x011B										
0x011C	CHINTENCLR13	7:0						SUSP	TCMPL	TERR
0x011D	CHINTENSET13	7:0						SUSP	TCMPL	TERR
0x011E	CHINTFLAG13	7:0						SUSP	TCMPL	TERR
0x011F	CHSTATUS13	7:0					CRCERR	FERR	BUSY	PEND
0x0120	CHCTRLA14	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x0124	CHCTRLB14	7:0							CMD[1:0]	
0x0125	CHPRILVL14	7:0								
0x0126	CHEVCTRL14	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0127	Reserved									
...										
0x012B										
0x012C	CHINTENCLR14	7:0						SUSP	TCMPL	TERR
0x012D	CHINTENSET14	7:0						SUSP	TCMPL	TERR
0x012E	CHINTFLAG14	7:0						SUSP	TCMPL	TERR
0x012F	CHSTATUS14	7:0					CRCERR	FERR	BUSY	PEND
0x0130	CHCTRLA15	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]			BURSTLEN[3:0]		
0x0134	CHCTRLB15	7:0							CMD[1:0]	
0x0135	CHPRILVL15	7:0								
0x0136	CHEVCTRL15	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0137	Reserved									
...										
0x013B										
0x013C	CHINTENCLR15	7:0						SUSP	TCMPL	TERR
0x013D	CHINTENSET15	7:0						SUSP	TCMPL	TERR
0x013E	CHINTFLAG15	7:0						SUSP	TCMPL	TERR
0x013F	CHSTATUS15	7:0					CRCERR	FERR	BUSY	PEND

22.8 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description. See *Register Access Protection* from Related Links.

Some registers are enable protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

Related Links

[22.5.7. Register Access Protection](#)

22.8.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
					LVLENx3	LVLENx2	LVLENx1	LVLENx0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							DMAENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bits 8, 9, 10, 11 – LVLENxx Priority Level x Enable

When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.
For details on arbitration schemes, see *Arbitration* from Related Links.
These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled.
1	Transfer requests for Priority level x will be handled.

Bit 1 – DMAENABLE DMA Enable

Setting this bit will enable the DMA module.
Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit when the DMAC module is disabled (DMAENABLE bit set to '0'), resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

Related Links

[22.6.2.4. Arbitration](#)

22.8.2 CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	CRCMODE[1:0]		CRCSRC[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 15:14 – CRCMODE[1:0] CRC Operating Mode
These bits define the block transfer mode.

Value	Name	Description
0x0	DEFAULT	Default operating mode
0x1	-	Reserved
0x2	CRCMON	Memory CRC monitor operating mode
0x3	CRCGEN	Memory CRC generation operating mode

Bits 13:8 – CRCSRC[5:0] CRC Input Source

These bits select the input source for generating the CRC. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	DISABLE	No action
0x01	IO	I/O interface
0x02 – 0x1F	-	Reserved
0x20	CH0	DMA channel 0
0x21	CH1	DMA channel 1
0x22	CH2	DMA channel 2
0x23	CH3	DMA channel 3
0x24	CH4	DMA channel 4
0x25	CH5	DMA channel 5
0x26	CH6	DMA channel 6
0x27	CH7	DMA channel 7
0x28	CH8	DMA channel 8
0x29	CH9	DMA channel 9
0x2A	CH10	DMA channel 10
0x2B	CH11	DMA channel 11
0x2C	CH12	DMA channel 12
0x2D	CH13	DMA channel 13
0x2E	CH14	DMA channel 14
0x2F	CH15	DMA channel 15
0x30	CH16	DMA channel 16
0x31	CH17	DMA channel 17

Value	Name	Description
0x32	CH18	DMA channel 18
0x33	CH19	DMA channel 19
0x34	CH20	DMA channel 20
0x35	CH21	DMA channel 21
0x36	CH22	DMA channel 22
0x37	CH23	DMA channel 23
0x38	CH24	DMA channel 24
0x39	CH25	DMA channel 25
0x3A	CH26	DMA channel 26
0x3B	CH27	DMA channel 27
0x3C	CH28	DMA channel 28
0x3D	CH29	DMA channel 29
0x3E	CH30	DMA channel 30
0x3F	CH31	DMA channel 31

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type

These bits select the CRC polynomial type.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2–0x3	-	Reserved

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size

These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	WORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3	-	Reserved

22.8.3 CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0] CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

22.8.4 CRC Checksum

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: PAC Write Protection, Enable-Protected

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0] CRC Checksum

These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

22.8.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CRCERR	CRCZERO	CRCBUSY
Access						R	R	R/W
Reset						0	0	0

Bit 2 – CRCERR CRC Error

This bit is read '1' when the memory CRC monitor detects data corruption.

Bit 1 – CRCZERO CRC Zero

This bit is cleared when a new CRC source is selected.

This bit is set when the CRC generation is complete and the CRC Checksum is zero.

Bit 0 – CRCBUSY CRC Module Busy

When used with an I/O interface ([CRCCTRL.CRCSRC=0x1](#)):

- This bit is cleared by writing a '1' to it
- This bit is set when the CRC Data Input (CRCDATAIN) register is written
- Writing a '1' to this bit will clear the CRC Module Busy bit
- Writing a '0' to this bit has no effect

When used with a DMA channel ([CRCCTRL.CRCSRC=0x20..,0x3F](#)):

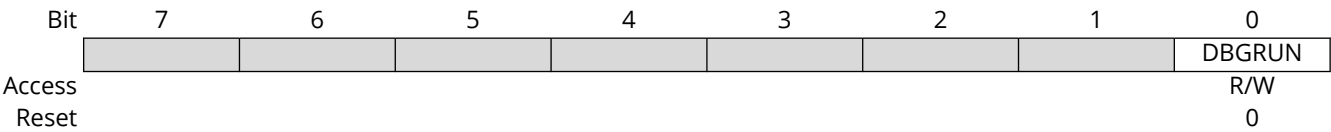
- This bit is cleared when the corresponding DMA channel is disabled
- This bit is set when the corresponding DMA channel is enabled
- Writing a '1' to this bit has no effect
- Writing a '0' to this bit has no effect

Related Links

[22.7. DMAC Register Summary](#)

22.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write Protection



Bit 0 – DBGRUN Debug Run
This bit is not reset by a Software Reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

22.8.7 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	SWTRIGn[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SWTRIGn[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SWTRIGn[15:0] Channel n Software Trigger [n = 15..0]

This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it. See *CHSTATUS* in the *DMAC Register Summary* from Related Links.

This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit. See *CHSTATUS* in the *DMAC Register Summary* from Related Links.

Writing a '0' to this bit will clear the bit.

Writing a '1' to this bit will generate a DMA software trigger on channel x, if CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared. See *CHSTATUS* in the *DMAC Register Summary* from Related Links.

Related Links

[22.7. DMAC Register Summary](#)

22.8.8 Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x40404040
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3	QOS03[1:0]		LVLPRI3[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RRLVLEN2	QOS02[1:0]		LVLPRI2[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RRLVLEN1	QOS01[1:0]		LVLPRI1[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RRLVLEN0	QOS00[1:0]		LVLPRI0[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bits 7, 15, 23, 31 – RRLVLEN Level Round-Robin Scheduling Enable

For details on arbitration schemes, see *Arbitration* from Related Links.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 5:6, 13:14, 21:22, 29:30 – QOS Level Quality of Service

0x0	DISABLE Background (no sensitive operation)
0x1	LOW Sensitive to bandwidth
0x2	MEDIUM Sensitive to latency
0x3	Critical Latency

Bits 0:4, 8:12, 16:20, 24:28 – LVLPRI Level Channel Priority Number

When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.

When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

Related Links

[22.6.2.4. Arbitration](#)

22.8.9 Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

This register allows the user to identify the lowest DMA channel with pending interrupt. An interrupt that handles several channels must consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR	CRCERR		SUSP	TCMPL	TERR
Access	R	R	R	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
						ID[4:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 15 – PEND Pending

This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy

This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error

This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 12 – CRCERR CRC Error

This bit will read '1' when the channel selected by Channel ID field (ID) has a CRC Error Status Flag bit set, and is set when the CRC monitor detects data corruption.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bit 10 – SUSP Channel Suspend

This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bit 9 – TCMPL Transfer Complete

This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bit 8 – TERR Transfer Error

This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

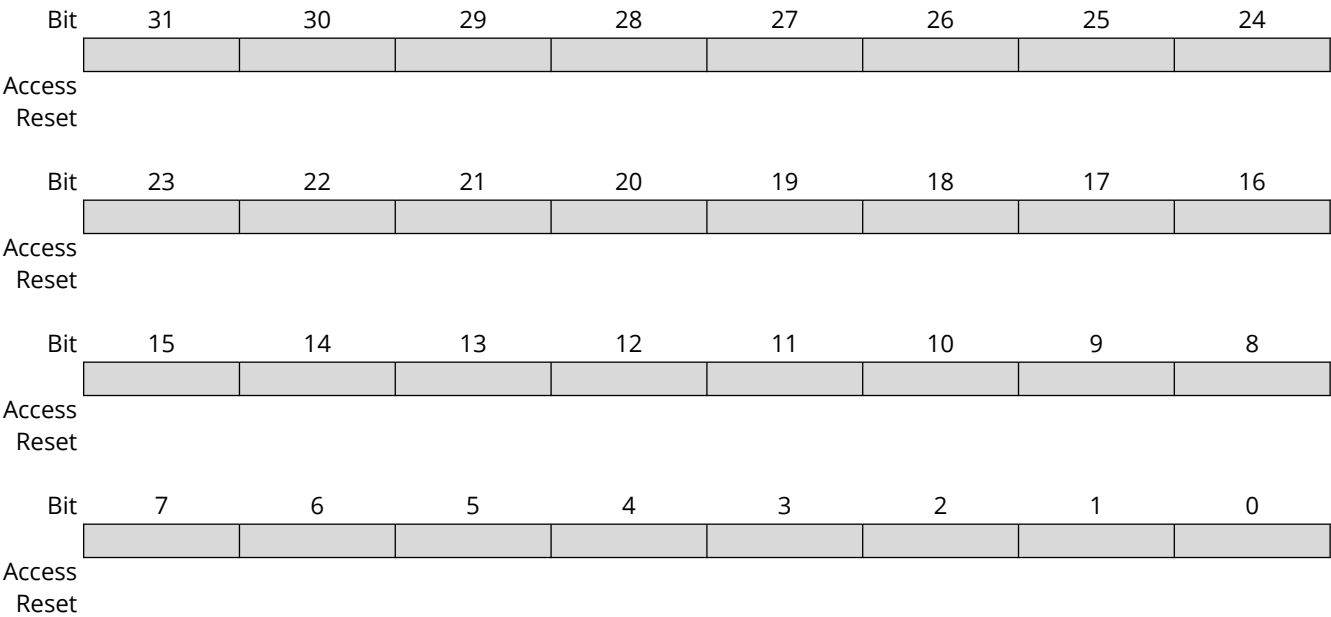
Bits 4:0 – ID[4:0] Channel ID

These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

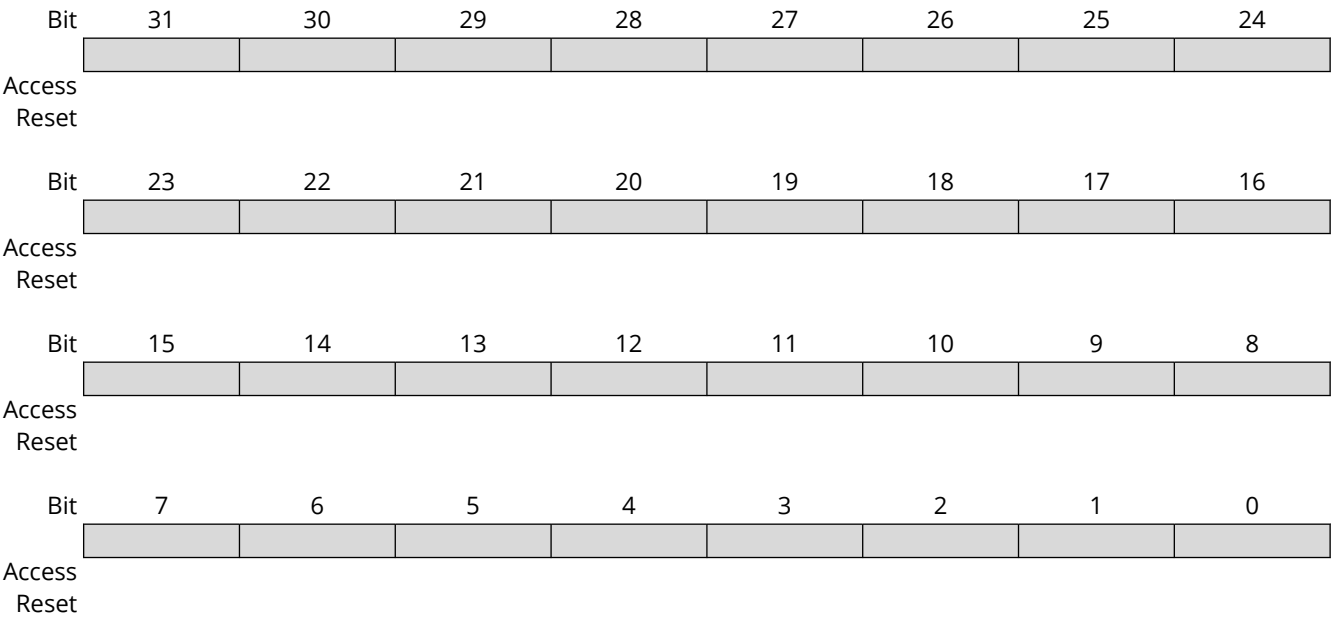
22.8.10 Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -



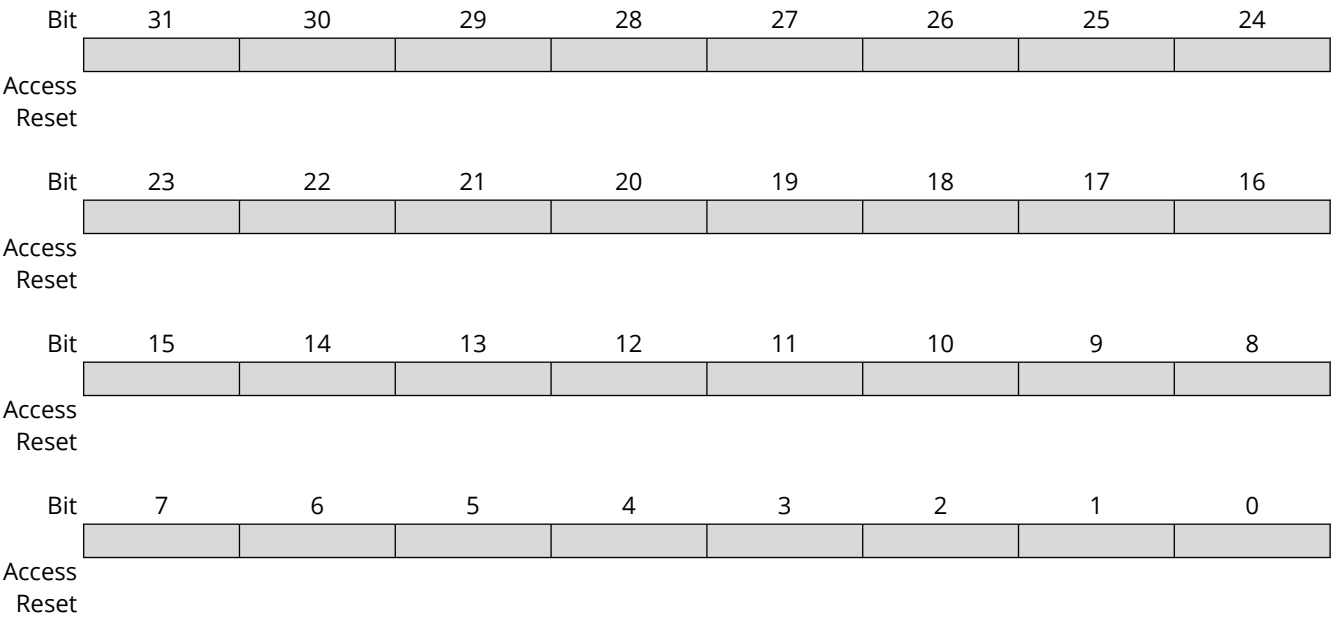
22.8.11 Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -



22.8.12 Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: -



22.8.13 Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ABUSY							
Access	R							
Reset	0							
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:16 – BTCNT[15:0] Active Channel Block Transfer Count

These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel Active Busy flag (ABUSY) is set.

Bit 15 – ABUSY Active Channel Busy

This bit is cleared when the active transfer count is written back in the write-back memory section. This bit is set when the next descriptor transfer count is read from the write-back memory section.

22.8.14 Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BASEADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASEADDR[31:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 64-bit aligned.

22.8.15 Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	WRBADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WRBADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WRBADDR[31:0] Write-Back Memory Base Address

These bits store the Write-Back memory base address. The value must be 64-bit aligned.

22.8.16 Channel Control A

Name: CHCTRLA
Offset: 0x40 + n*0x10 [n=0..15]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
			THRESHOLD[1:0]		BURSTLEN[3:0]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TRIGACT[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit	15	14	13	12	11	10	9	8
	TRIGSRC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bits 29:28 – THRESHOLD[1:0] FIFO Threshold

These bits define the threshold from which the DMA starts to write to the destination. These bits have no effect in the case of single beat transfers.

These bits are not enable-protected.

Value	Name	Description
0x0	1BEAT	Destination write starts after each beat source address read
0x1	2BEATS	Destination write starts after 2-beats source address read
0x2	4BEATS	Destination write starts after 4-beats source address read
0x3	8BEATS	Destination write starts after 8-beats source address read

Bits 27:24 – BURSTLEN[3:0] Burst Length

These bits define the burst mode.

These bits are not enable-protected.

Value	Name	Description
0x0	SINGLE	Single-beat burst
0x1	2BEAT	2-beats burst length
0x2	3BEAT	3-beats burst length
0x3	4BEAT	4-beats burst length
0x4	5BEAT	5-beats burst length
0x5	6BEAT	6-beats burst length
0x6	7BEAT	7-beats burst length
0x7	8BEAT	8-beats burst length
0x8	9BEAT	9-beats burst length
0x9	10BEAT	10-beats burst length
0xA	11BEAT	11-beats burst length
0xB	12BEAT	12-beats burst length
0xC	13BEAT	13-beats burst length

Value	Name	Description
0xD	14BEAT	14-beats burst length
0xE	15BEAT	15-beats burst length
0xF	16BEAT	16-beats burst length

Bits 21:20 – TRIGACT[1:0] Trigger Action

These bits define the trigger action used for a transfer.
These bits are not enable-protected.

Value	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1		Reserved
0x2	BURST	One trigger required for each burst transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 15:8 – TRIGSRC[7:0] Trigger Source

These bits define the peripheral that will be the source of a trigger.

Table 22-2. Triggers Map

Number	Name
0	Unused (Tied to 1'b0)
1	RTC_DMAMC_ID_TIMESTAMP
2	DSU_DMAMC_ID_DCC0
3	DSU_DMAMC_ID_DCC1
4	SERCOM0_DMAMC_ID_RX
5	SERCOM0_DMAMC_ID_TX
6	SERCOM1_DMAMC_ID_RX
7	SERCOM1_DMAMC_ID_TX
8	SERCOM2_DMAMC_ID_RX
9	SERCOM2_DMAMC_ID_TX
10	SERCOM3_DMAMC_ID_RX
11	SERCOM3_DMAMC_ID_TX
12	TCC0_DMAMC_ID_OVF
13	TCC0_DMAMC_ID_MC_0
14	TCC0_DMAMC_ID_MC_1
15	TCC0_DMAMC_ID_MC_2
16	TCC0_DMAMC_ID_MC_3
17	TCC0_DMAMC_ID_MC_4
18	TCC0_DMAMC_ID_MC_5
19	TCC1_DMAMC_ID_OVF
20	TCC1_DMAMC_ID_MC_0
21	TCC1_DMAMC_ID_MC_1
22	TCC1_DMAMC_ID_MC_2
23	TCC1_DMAMC_ID_MC_3
24	TCC1_DMAMC_ID_MC_4
25	TCC1_DMAMC_ID_MC_5
26	TCC2_DMAMC_ID_OVF
27	TCC2_DMAMC_ID_MC_0
28	TCC2_DMAMC_ID_MC_1
29	TC0_DMAMC_ID_OVF
30	TC0_DMAMC_ID_MC_0
31	TC0_DMAMC_ID_MC_1
32	TC1_DMAMC_ID_OVF
33	TC1_DMAMC_ID_MC_0
34	TC1_DMAMC_ID_MC_1
35	TC2_DMAMC_ID_OVF

.....continued	
Number	Name
36	TC2_DMACH_ID_MC_0
37	TC2_DMACH_ID_MC_1
38	TC3_DMACH_ID_OVF
39	TC3_DMACH_ID_MC_0
40	TC3_DMACH_ID_MC_1
41	AES_DMACH_ID_WR
42	AES_DMACH_ID_RD
43	QSPI_DMACH_ID_RX
44	QSPI_DMACH_ID_TX

Bit 6 – RUNSTDBY Channel run in standby

This bit is used to keep the DMAC channel running in standby mode.

This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE Channel Enable

Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

Writing a '1' to this bit will enable the DMA channel.

This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset

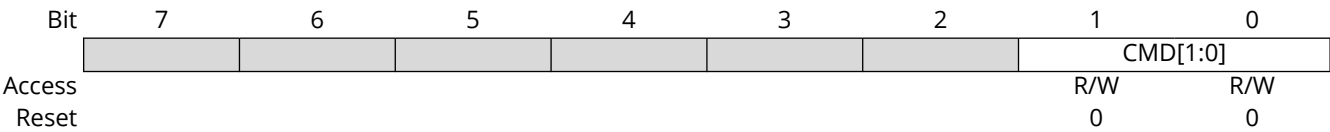
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

22.8.17 Channel Control B

Name: CHCTRLB
Offset: 0x44 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection



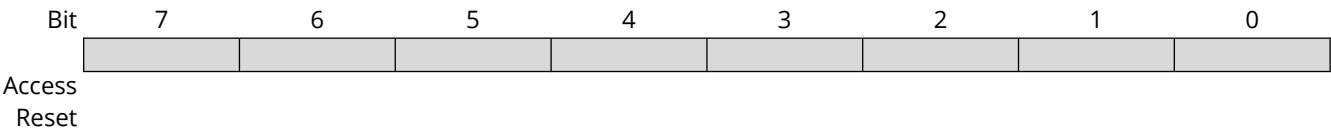
Bits 1:0 – CMD[1:0] Software Command
These bits define the software commands. See *Channel Suspend* and *Channel Resume and Next Suspend Skip* from Related Links.
These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Related Links
[22.6.3.3. Channel Suspend](#)
[22.6.3.4. Channel Resume and Next Suspend Skip](#)

22.8.18 Channel Priority Level

Name: CHPRILVL
Offset: 0x45 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection



22.8.19 Channel Event Control

Name: CHEVCTRL
Offset: 0x46 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – EVOE Channel Event Output Enable

This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the Channel Event Output Selection bits (CHEVCTRL.EVOMODE).

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 6 – EVIE Channel Event Input Enable

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 5:4 – EVOMODE[1:0] Channel Event Output Mode

These bits define the channel event output selection. For more details on event output generation, see *Event Output Selection* from Related Links.

Value	Name	Description
0x0	DEFAULT	Block event output selection. See BTCTRL.EVOSEL for available selections.
0x1	TRIGACT	Ongoing trigger action
0x2–0x3		Reserved

Bits 2:0 – EVACT[2:0] Channel Event Input Action

These bits define the event input action. The action is executed only if the corresponding EVIE bit in the CHEVCTRL register of the channel is set. For more details on event actions, see *Event Input Actions* from Related Links. These bits are available only for channels with event input support.

Value	Name	Description
0x0	NOACT	No action
0x1	TRIG	Transfer and periodic transfer trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	INCPRI	Increase priority

Related Links

[22.6.3.5. Event Input Actions](#)

[22.6.3.6. Event Output Selection](#)

22.8.20 Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

22.8.21 Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

22.8.22 Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E + n*0x10 [n=0..15]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend

This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.

For details on available event input actions, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.

For details on available block actions, see *BTCTRL* in the *DMAC Register Summary (SRAM)* from Related Links.

Bit 1 – TCMPL Channel Transfer Complete

This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR Channel Transfer Error

This flag is cleared by writing a '1' to it.

This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

Related Links

[22.9. DMAC Register Summary \(SRAM\)](#)

[22.7. DMAC Register Summary](#)

22.8.23 Channel Status

Name: CHSTATUS
Offset: 0x4F + n*0x10 [n=0..15]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					CRCERR	FERR	BUSY	PEND
Access					R/W	R	R	R
Reset					0	0	0	0

Bit 3 – CRCERR Channel CRC Error

This bit is set when the CRC monitor detects data corruption. This bit is cleared by writing '1' to it, or by clearing the CRC Error bit in the INTPEND register (INTPEND.CRCERR). See *INTPEND* in the *DMAC Register Summary* from Related Links.

Bit 2 – FERR Channel Fetch Error

This bit is cleared when a software resume command is executed.
This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY Channel Busy

This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.
This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND Channel Pending

This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

Related Links

[22.7. DMAC Register Summary](#)

22.9 DMAC Register Summary (SRAM)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
		15:8	SRCADDR[15:8]							
		23:16	SRCADDR[23:16]							
		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
		15:8	DSTADDR[15:8]							
		23:16	DSTADDR[23:16]							
		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
		15:8	DESCADDR[15:8]							
		23:16	DESCADDR[23:16]							
		31:24	DESCADDR[31:24]							

22.10 Register Description - SRAM

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the “PAC Write-Protection” property in each individual register description. See *Register Access Protection* from Related Links.

Some registers are enable protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

Related Links

[22.5.7. Register Access Protection](#)

22.10.1 Block Transfer Control

Name: BTCTRL
Offset: 0x00
Reset: 0x0000
Property: -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size

These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL Step Selection

This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable

Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled
1	The Destination Address Increment is enabled

Bit 10 – SRCINC Source Address Increment Enable

Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled
1	The Source Address Increment is enabled

Bits 9:8 – BEATSIZE[1:0] Beat Size

These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	WORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
other		Reserved

Bits 4:3 – BLOCKACT[1:0] Block Action

These bits define what actions the DMAC must take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection

These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID Descriptor Valid

Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid
1	The descriptor is valid

22.10.2 Block Transfer Count

Name: BTCNT
Offset: 0x02
Property: -

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	BTCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BTCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BTCNT[15:0] Block Transfer Count

This bit group holds the 16-bit block transfer count.
During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

22.10.3 Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Property: -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SRCADDR[31:0] Transfer Source Address

This bit field holds the block transfer source address.

When source address incrementation is disabled (BTCTRL.SRCINC=0), SRCADDR corresponds to the last beat transfer address in the block transfer.

When source address incrementation is enabled (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL = 1:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1) \cdot 2^{\text{STEPsize}}$$

If BTCTRL.STEPSEL= 0:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1)$$

- SRCADDR_{START} is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPsize is the configured number of beats for each incrementation

22.10.4 Block Transfer Destination Address

Name: DSTADDR
Offset: 0x08
Property: -

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DSTADDR[31:0] Transfer Destination Address

This bit field holds the block transfer destination address.

When destination address incrementation is disabled (BTCTRL.DSTINC = 0), DSTADDR corresponds to the last beat transfer address in the block transfer.

When destination address incrementation is enabled (BTCTRL.DSTINC = 1), DSTADDR is calculated as follows:

If BTCTRL.STEPSEL = 1:

$$DSTADDR = DSTADDR_{START} + BTCNT \cdot (BEATSIZE + 1)$$

If BTCTRL.STEPSEL = 0:

$$DSTADDR = DSTADDR_{START} + BTCNT \cdot (BEATSIZE + 1) \cdot 2^{STEPsize}$$

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPsize is the configured number of beats for each incrementation

22.10.5 Next Descriptor Address

Name: DESCADDR
Offset: 0x0C
Property: -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access	-	-	-	-	-	-	-	-
Reset								
Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset								
Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset								
Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset								

Bits 31:0 – DESCADDR[31:0] Next Descriptor Address

This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

23. External Interrupt Controller (EIC)

23.1 Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Also, each external pin can be configured to be asynchronous in order to wake-up the device from Sleep modes where all clocks have been disabled. External pins can generate an event.

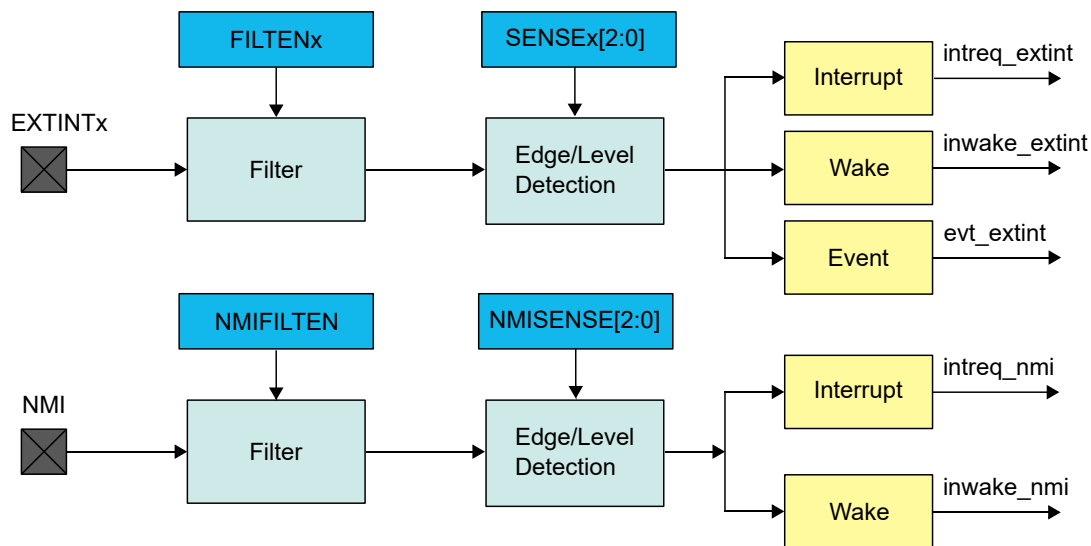
A separate Non-Maskable Interrupt (NMI) is supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other Interrupt mode.

23.2 Features

- Up to four external pins (EXTINTx), plus one non-maskable pin (NMI)
- Dedicated, Individually Maskable Interrupt for Each Pin
- Interrupt on Rising, Falling, or Both Edges
- Interrupt on High or Low Levels
- Asynchronous Interrupts for Sleep Modes Without Clock
- Filtering of External Pins
- Event Generation from EXTINTx

23.3 Block Diagram

Figure 23-1. EIC Block Diagram



23.4 Signal Description

Signal Name	Type	Description
EXTINT[3..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal may be available on several pins.

23.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

23.5.1 I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

23.5.2 Power Management

All interrupts are available down to STANDBY Sleep mode, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any Sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from Sleep modes. Events connected to the Event System can trigger other operations in the system without exiting Sleep modes.

23.5.3 Clocks

The EIC bus clock (PB1_CLK) can be enabled and disabled by the CRU, the default state of PB1_CLK can be found in the CRU and PMD registers.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low-Power 32 KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral.

GCLK_EIC is configured and enabled in the CRU registers (see *Clock and Reset (CRU)* from Related Links).

CLK_ULP32K is provided by the internal Ultra Low-Power (OSCULP32K) Oscillator in the CRU module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (PB1_CLK). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

Related Links

[13. Clock and Reset Unit \(CRU\)](#)

23.5.4 DMA

Not applicable.

23.5.5 Interrupts

There are several interrupt request lines, some (the number depends on the product variant) for the external interrupts (EXTINT) and one for Non-Maskable Interrupt (NMI).

Each EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is connected to the interrupt controller, but does not require the interrupt to be configured.

23.5.6 Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

[28. Event System \(EVSYS\)](#)

23.5.7 Debug Operation

When the CPU is halted in Debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

23.5.8 Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

23.5.9 Analog Connections

Not applicable.

23.6 Functional Description

23.6.1 Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

23.6.2 Basic Operation

23.6.2.1 Initialization

The EIC must be initialized in the following order:

1. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL).
2. Enable GCLK_EIC or CLK_ULP32K when one of the following configurations is selected:
 - The NMI uses edge detection or filtering
 - One EXTINT uses filtering
 - One EXTINT uses edge detection
 - One EXTINT uses debouncing

GCLK_EIC is used when a frequency higher than 32 KHz is required for filtering.

CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K, write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).

3. Configure the EIC input sense and filtering by writing the Configuration register (CONFIG).
4. Enable the EIC by writing a '1' to CTRLA.ENABLE.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE=0):

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Configuration register (CONFIG)
- External Interrupt Asynchronous Mode register (ASYNCH)
- Debouncer Enable register (DEBOUNCEN)
- Debounce Prescaler register (DPRESCALER)

Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the “Enable-Protected” property in the register description.

See *NMICTRL*, *CTRLA*, *CONFIG*, *ASYNCH*, *DEBOUNCEN*, *DPRESCALER*, *EVCTRL* registers in the *EIC Register Summary* from Related Links.

Related Links

[23.7. EIC Register Summary](#)

23.6.2.2 Enabling, Disabling and Resetting

The EIC is enabled by writing a ‘1’ to the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to ‘0’.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

23.6.3 External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIG.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met.

In level-sensitive mode, when the interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if the bit Filter Enable x in the Configuration n register (CONFIG.FILTENx) is written to ‘1’. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

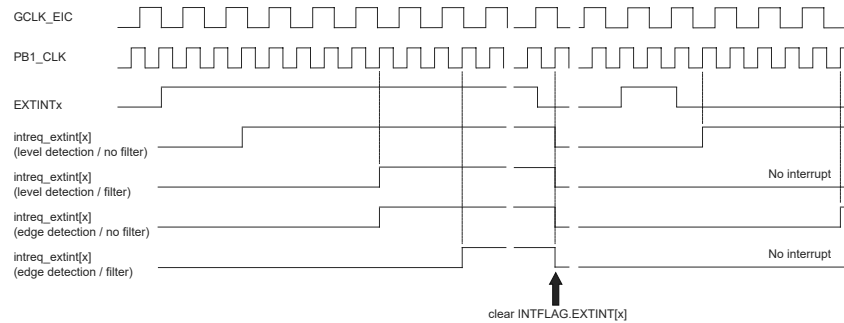
Table 23-1. Majority Vote Filter

Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

If filtering or edge detection is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the CRU. In these modes the external pin is sampled at the EIC clock rate, thus pulses with duration lower than two EIC clock periods may not be properly detected.

Figure 23-2. Interrupt Detection Latency by Modes (Rising Edge)



The detection latency depends on the detection mode.

Table 23-2. Detection Latency

Detection Mode	Latency (Worst Case)
Level without filter	Five PB1_CLK periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five PB1_CLK periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five PB1_CLK periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five PB1_CLK periods

23.6.4 Additional Features

23.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC module is not required to be enabled.

When an NMI is detected, the Non-maskable Interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

23.6.4.2 Asynchronous Edge Detection Mode

The EXTINT edge detection operates synchronously or asynchronously, as selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In Synchronous Edge Detection Mode, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. The EIC clock is needed in this mode.

The Synchronous Edge Detection Mode can be used in Idle and Standby sleep modes.

In Asynchronous Edge Detection Mode, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. The EIC clock is not needed in this mode.

Note: The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

23.6.4.3 Interrupt Pin Debouncing

The external interrupt pin (EXTINT) edge detection can use a debouncer to improve input noise immunity. When selected, the debouncer can work in the synchronous mode or the asynchronous mode, depending on the configuration of the ASYNCH.ASYNCH[x] bit for the pin. The debouncer uses the EIC clock as defined by the bit CTRLA.CKSEL to clock the debouncing circuitry. The debouncing time frame is set with the debouncer prescaler DPRESALER.PRESCALERn, which provides the *low frequency clock* tick that is used to reject higher frequency signals.

The debouncing mode for pin EXTINT x can be selected only if the Sense bits in the Configuration y register (CONFIG.SENSEx) are set to RISE, FALL or BOTH. If the debouncing mode for pin EXTINT x is selected, the filter mode for that pin (CONFIG.FILTENx) can not be selected.

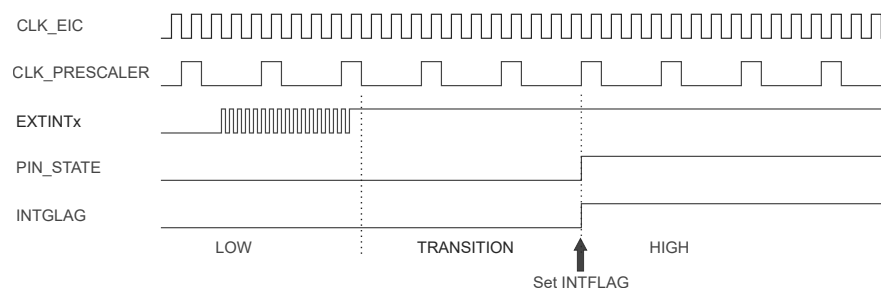
The debouncer manages an internal “valid pin state” that depends on the external interrupt (EXTINT) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continuously on EIC clock. The sampled value is evaluated on each *low frequency clock* tick to detect a transitional edge when the sampled value is different of the current valid pin state. The sampled value is evaluated on each EIC clock when DPRESALER.TICKON=0 or on each *low frequency clock* tick when DPRESALER.TICKON=1, to detect a bounce when the sampled value is equal to the current valid pin state. Transitional edge detection increments the transition counter of the EXTINT pin, while bounce detection resets the transition counter. The transition counter must exceed the transition count threshold as defined by the DPRESALER.STATESn bitfield. In the synchronous mode the threshold is 4 when DPRESALER.STATESn=0 or 8 when DPRESALER.STATESn=1. In the asynchronous mode the threshold is 4.

The valid pin state for the pins can be accessed by reading the register PINSTATE for both synchronous or asynchronous debouncing mode.

Synchronous edge detection In this mode the external interrupt (EXTINT) pin is sampled continuously on EIC clock.

1. A pin edge transition will be validated when the sampled value is consistently different of the current valid pin state for 4 (or 8 depending on bit DPRESALER.STATESn) consecutive ticks of the low frequency clock.
2. Any pin sample, at the *low frequency clock* tick rate, with a value opposite to the current valid pin state will increment the transition counter.
3. Any pin sample, at EIC clock rate (when DPRESALER.TICKON=0) or the *low frequency clock* tick (when DPRESALER.TICKON=1), with a value identical to the current valid pin state will return the transition counter to zero.
4. When the transition counter meets the count threshold, the pin edge transition is validated and the pin state PINSTATE.PINSTATE[x] is changed to the detected level.
5. The external interrupt flag (INTFLAG.EXTINT[x]) is set when the pin state PINSTATE.PINSTATE[x] is changed.

Figure 23-3. EXTINT Pin Synchronous Debouncing (Rising Edge)

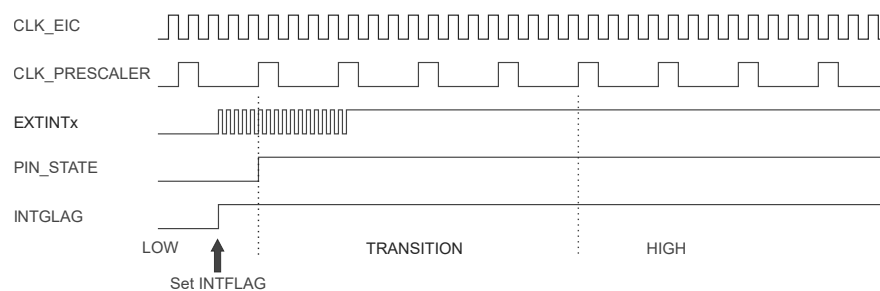


In the synchronous edge detection mode, the EIC clock is required. The synchronous edge detection mode can be used in Idle and Standby sleep modes.

Asynchronous edge detection In this mode, the external interrupt (EXTINT) pin directly drives an asynchronous edges detector which triggers any rising or falling edge on the pin:

1. Any edge detected that indicates a transition from the current valid pin state will immediately set the valid pin state PINSTATE.PINSTATE[x] to the detected level.
2. The external interrupt flag (INTFLAG.EXTINT[x]) is immediately changed.
3. The edge detector will then be idle until no other rising or falling edge transition is detected during 4 consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the idle state will return the transition counter to 0.
5. After 4 consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

Figure 23-4. EXTINT Pin Asynchronous Debouncing (Rising Edge)



In this mode, the EIC clock is requested. The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

23.6.5 DMA Operation

Not applicable.

23.6.6 Interrupts

The EIC has the following interrupt sources:

- External interrupt (EXTINTx) pins. See *Basic Operation* from Related Links.
- Non-maskable interrupt (NMI) pin. See *Non-Maskable Interrupt (NMI)* from Related Links

Each interrupt source has an associated Interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an Interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET = 1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR = 1). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear Interrupt flags. The EIC has one interrupt request line for each external interrupt (EXTINTx) and one line for NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which Interrupt condition is present.

Notes:

1. Interrupts must be globally enabled for interrupt requests to be generated.
2. If an external interrupt (EXTINT) is common on two or more I/O pins, only one will be active (the first one programmed).

Related Links

[23.6.2. Basic Operation](#)

[23.6.4.1. Non-Maskable Interrupt \(NMI\)](#)

23.6.7 Events

The EIC can generate the following output events:

- External event from pin (EXTINT0-3)

Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. For more details on configuring the event system, see *Event System (EVSYS)* from Related Links.

When the condition on pin EXTINTx matches the configuration in the CONFIG register, the corresponding event is generated, if enabled.

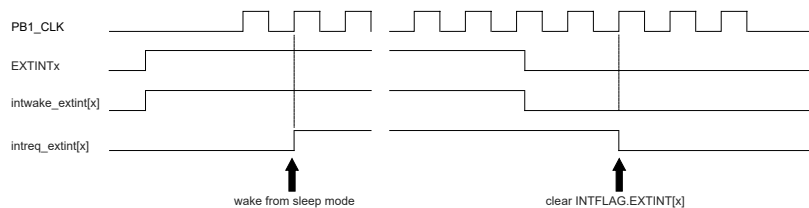
Related Links

[28. Event System \(EVSYS\)](#)

23.6.8 Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 23-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



23.6.9 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

See CTRLA from Related Links.

Related Links

[23.8.1. CTRLA](#)