PIC32CX-BZ3 and WBZ35x Family Data Sheet

PIC32CX-BZ3 and WBZ35x Family

Introduction

The PIC32CX-BZ3 family of SoCs are general purpose, low-cost, 32-bit Microcontrollers (MCUs) with Cortex® M4F ARM® processor, secure Bluetooth® Low Energy 5.2, IEEE® 802.15.4 wireless connectivity, an ultra-low power 2.4 GHz transceiver, a hardware-based security accelerator, capacitive touch capabilities and secure boot functionality.

The WBZ35x modules, which are fully RF-certified wireless modules, are built around PIC32CX-BZ3 SoC. The WBZ35x modules are available with a PCB antenna or U.FL connector for an external antenna.

The PIC32CX-BZ3 family supports a comprehensive range of peripherals for low-cost Internet of Things (IoT) solutions in automotive, smart factory and smart home applications.

- Analog-to-Digital Converter (ADC)
- Capacitive Voltage Divider (CVD) Controller for Touch Sensing
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)
- Quad I/O Serial Peripheral Interface (QSPI)
- Universal Asynchronous Receiver-Transmitter (UART)
- Timer/Counter for Control Applications (TCC)
- Digital-to-Analog Converter (DAC)
- Local Interconnect Network (LIN)
- Serial Wire Debug (SWD)

The secure boot ROM has the capability to check the integrity and authenticity of the application prior to executing it. This verification mechanism is a key element to ensure the system root of trust for the deployment and execution of the secure firmware.

PIC32CX-BZ3 SoC Family Features

Operating Conditions

- 1.9-3.6V, -40°C to +125°C, DC to 64 MHz
  - AEC Q100 Grade 1 qualified with reduced parameters
- 1.9-3.6V, -40°C to +105°C, DC to 64 MHz
  - AEC Q100 Grade 2 qualified

Core: 64 MHz ARM Cortex-M4F

- 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
- CoreSight™ Embedded Trace Buffer (ETB)
- Trace Port Interface Unit (TPIU)
- IEEE 754-Compliant Floating Point Unit (FPU)

Memories

- 64-KB ROM for Secure Boot
  - Support for asymmetric secure boot
- 3072-Bit of eFuse
  - Secure boot key storage
  - Debug lock
- 512-KB On-Chip Self-Programmable Flash With:
  - Error Correction Code (ECC)
  - Prefetch module to speed up Flash accesses
  - 20k cycles endurance (100k cycles with erase retry option) and 20 years of data retention support
- 32-KB Boot Flash Memory (Eight Pages)
  - User boot code configuration
  - Flexible device configuration
- 96-KB Multi-Port Programmable QoS SRAM Main Memory
  - 32-KB of ECC RAM option
  - 32-KB of RAM space for CoreSight ETB debug usage when enabled
  - Up to 32-KB of SRAM can be retained in Backup mode
- Up to 4-KB of Tightly Coupled Memory (TCM)

System

- Power-on Reset (POR) and Brown-out Reset (BOR)
- Internal and External Clock Options
- External Interrupt Controller (EIC)
  - Up to four external interrupts
  - One non-maskable interrupt
- Extensive Debug and Trace Capabilities
  - 2-pin Serial Wire Debug (SWD) programming and debugging interface
  - ETM trace interface pins for serial wire trace

Supported Connectivity Standards

- Complies with:
  - Bluetooth Low Energy 5.2
  - IEEE 802.15.4, Zigbee® 3.0

Power Supply

- Integrated PMU with:
  - MLDO (linear) mode
- Integrated On-Chip 1.5V Low Dropout (LDO) Regulator for eFuse
- Integrated On-Chip 1.2V Core Low Dropout Regulator (CLDO)

- Integrated 16 MHz ±20 ppm Crystal Oscillator (External Low Cost Crystal)
- POR and BOR on 3.3V and 1.2V Rails
- Run, Idle, Standby Sleep, Deep Sleep and Extreme Deep Sleep Modes
- Sleep Walking Peripherals

2.4 GHz RF Transceiver

- Integrated 2.4 GHz Ultra Low-Power RF Transceiver Shared Between Bluetooth Low Energy and IEEE 802.15.4 Modems and Link (MAC) Controllers
- Integrated TRX Switch and Balun with One Single-Ended RFIO for TX/RX
- High Efficiency Switching Power Amplifier (PA)
- Programmable Transmit Output Power Ranges from -24 dBm to +11 dBm with 1 dB Step Size
- Supported Data Rates:
  - Bluetooth Low Energy 5.2: 2 Mbps, 1 Mbps, 500 kbps and 125 kbps
  - IEEE 802.15.4: 250 kbps
  - Proprietary 2.4 GHz: 2 Mbps, 1 Mbps and 500 kbps

Bluetooth

- Bluetooth Low Energy 5.2 Certified
- Up to +11 dBm Programmable Transmit Output Power
- Typical Receiver Power Sensitivity:
  - -95 dBm for Bluetooth Low Energy 2 Mbps
  - -98 dBm for Bluetooth Low Energy 1 Mbps
  - -108 dBm for Bluetooth Low Energy 125 Kbps
  - -102 dBm for Bluetooth Low Energy 500 Kbps
- Bluetooth Low Energy Supported Features:
  - Low energy power control
  - 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
  - Robust to interference with wideband RSSI detector
- ECDH P256 Hardware Engine for Link Key Generation, when Bluetooth Pairing
- AES128 Hardware Module for Real-Time Bluetooth Payload Data Encryption
- Bluetooth Qualification Test Facility (BQTF) Certification
- Supports SIG Defined/Custom Bluetooth Low Energy Profiles and Services
- 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

  - Alert notification service (Apple Notification Center Service (ANCS))
  - Proximity reporter and time information
  - 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

IEEE 802.15.4

- PLCP Service Data Unit (PSDU) Data Rate: 250 Kbps
  - Proprietary data rates: 500 Kbps, 1 Mbps and 2 Mbps
- Programmable RX Mode
  - -103 dBm RX sensitivity for 250 Kbps in Continuous mode
  - -100 dBm RX 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 +11 dBm
- Hardware Assisted MAC
  - Auto acknowledge
  - Auto retry
  - Channel access back-off
  - Automated FCS check
  - Automatic Address filtering
- Zigbee 3.0 Specification Compliant
- SFD Detection; Spreading; De-spreading; Framing; CRC-16 Computation
- Independent TX/RX Buffers for Improved CPU Offloading
  - 128-byte TX and 128-byte RX frame buffer

High Performance Peripherals

- 16-Channel Direct Memory Access Controller (DMAC)
  - Built-in CRC with memory CRC generation and monitors hardware support
- One 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
- Two Serial Communication Interfaces (SERCOM), each Configurable to Operate as:
  - Universal Synchronous Asynchronous Receiver Transmitter (USART) with full duplex and single-wire half duplex configuration
  - ISO7816
  - I^2C up to 1 MHz
  - LIN Commander/Responder
  - RS485
  - SPI inter-byte space
- One SERCOM Configured as I^2C-Only Interface
- Eight 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 by paring two TCs
- 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 to increase resolution up to 5 bits and reduce 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 four wake-up pins with tamper detection and debouncing filter
- Watchdog Timer (WDT) with Window Mode
- Deadman Timer (DMT)
- Frequency Meter (FREQM)
- Configurable Custom Logic (CCL) with Two Look-up Table (LUT)
- One 7-Bit General Purpose Digital-to-Analog Converter (DAC)
- One 12-Bit, 2 Msps ADC SAR Core with Up to Eight Channels:
  - Differential and single-ended input
  - Automatic offset and gain error compensation
- Up to Two Analog Comparator (AC) with Window Compare Function
- Up to Eight Capacitive Voltage Divider (CVD) Channels for Touch Button Support (Using Shared ADC SAR Core)
- One Temperature Sensor (Die Temperature) Built into Wireless Subsystem

Cryptography

- High Performance Cryptographic Accelerators for Advanced Encryption Standard (AES), Secure Hash Algorithm (SHA), RSA and ECC
- NIST 800-90B Compliant True Random Number Generator (TRNG)
- Integrated Scatter Gather DMA

Oscillators

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

I/O

- 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 (PIC32CX5109BZ31048)
- Up to 14 Programmable I/O Pins (PIC32CX5109BZ31032)

DC Specification

- Electrostatic Discharge (ESD) Protection
  - Human Body Model (HBM): 2 kV (typical)
  - Charged Device Model (CDM): 250V (typical)
- Boot time(1)(2): ~7.1 ms

Package

- PIC32CX5109BZ31048
  - 48-pin QFN
  - Size: 6 mm x 6 mm x 1 mm
- PIC32CX5109BZ31032
  - 32-pin QFN
  - Size: 5 mm x 5 mm x 1 mm

Notes:

1. The numbers are design targets only and subject to change after device characterization is complete.
2. The time from Reset release until control is transferred to the Flash without code authentication.

WBZ35x Module Features

The following section lists the WBZ35x Module-related features, which complement SoC features.

WBZ35x Module Variants

- WBZ351 Based on PIC32CX5109BZ31048 SoC
  - WBZ351PE (PCB antenna)
  - WBZ351UE (u.FL)
- WBZ350 Based on PIC32CX5109BZ31032 SoC
  - WBZ350PE (PCB antenna)
  - WBZ350UE (u.FL)

Antenna

- On-Board PCB Antenna
- External Antenna

Security

- Integrated Trust&GO

Clock Management

- Integrated 16 MHz POSC

Advanced Analog

- 12-Bit ADC
  - Up to eight analog channels (PIC32CX5109BZ31048)
  - Up to four analog channels (PIC32CX5109BZ31032)
- CVD Touch Support Using Shared ADC
  - Up to eight CVD channels (PIC32CX5109BZ31048)
  - Up to four CVD channels (PIC32CX5109BZ31032)
- 7-Bit General Purpose DAC

Input/Output

- Up to 27 GPIO Pins (WBZ351)
- Up to 14 GPIO Pins (WBZ350)

Package and Operating Conditions

- WBZ351 Module Package:
  - 39-pin SMD package with Shield CAN
  - Size: 15.5 mm x 20.7 mm x 2.8 mm
- WBZ350 Module Package:
  - 30-pin SMD package with Shield CAN
  - Size: 13.4 mm x 18.7 mm x 2.8 mm
- Operating Conditions:
  - 1.9-3.6V for variants without Trust&GO
  - Operating temperature: -40ºC to +85ºC

Certifications

- WBZ351 Module Certified to FCC, ISED, CE, UKCA, MIC, KCC, NCC and SRRC Radio Regulations
- WBZ350 Module Certified to FCC, ISED and CE Radio Regulations
- Bluetooth SIG
- RoHS and REACH Compliant

Acronyms and Abbreviations

For complete list of acronyms and abbreviations, refer to 46. Appendix B: Acronyms and Abbreviations.

Table of Contents

1. Ordering Information

This chapter provides the ordering information of the PIC32CX-BZ3 SoC and the WBZ35 Module.

1.1 PIC32CX-BZ3 SoC Ordering Information

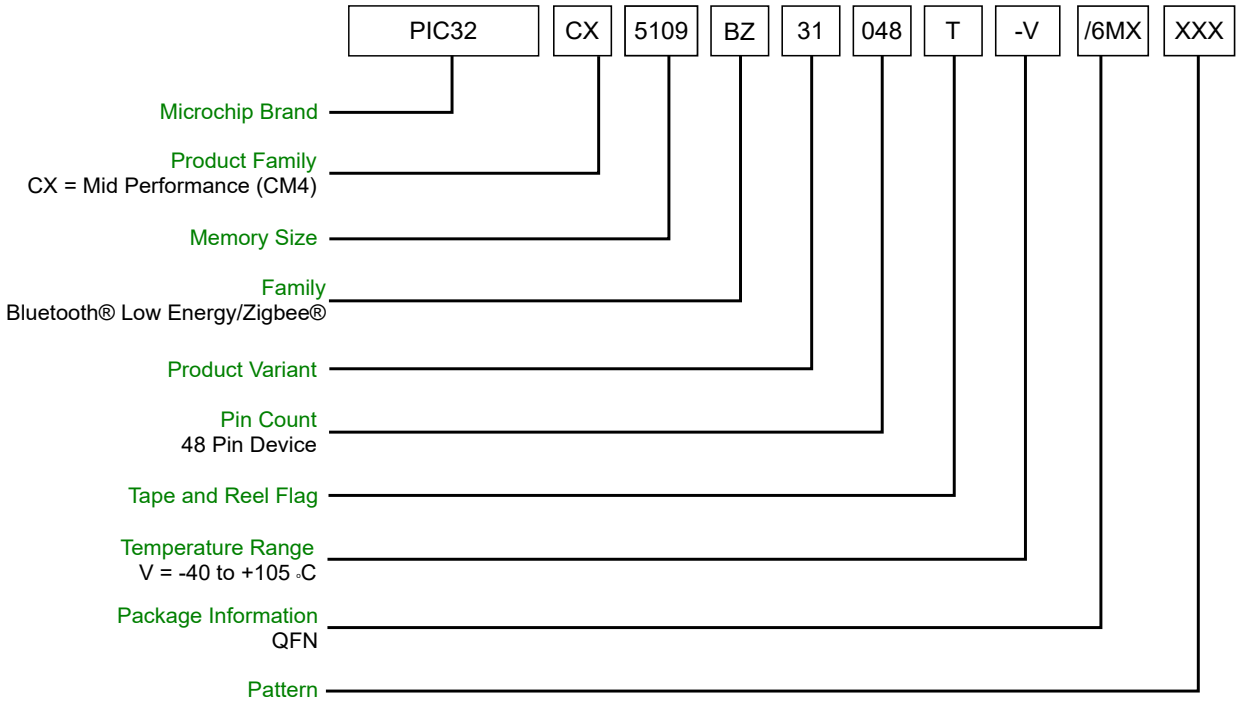
The following table describes the ordering information of the PIC32CX-BZ3 SoC.

Table 1-1. PIC32CX-BZ3 SoC Ordering Details

SoC Name	Pin and Package	Description	Ordering Code
PIC32CX5109BZ31048	48-pin and QFN (6 mm x 6 mm x 1 mm)	32-bit ARM®Cortex™-M4 with MCU Bluetooth®/Zigbee® Connectivity	PIC32CX5109BZ31048-V/ZWX
PIC32CX5109BZ31032	32-pin and QFN (5 mm x 5 mm x 1 mm)		PIC32CX5109BZ31032-V/ZWX

The following figure illustrates the details of the PIC32CX-BZ3 ordering information.

Figure 1-1. PIC32CX-BZ3 Ordering Information



1.2 WBZ35x Module Ordering Information

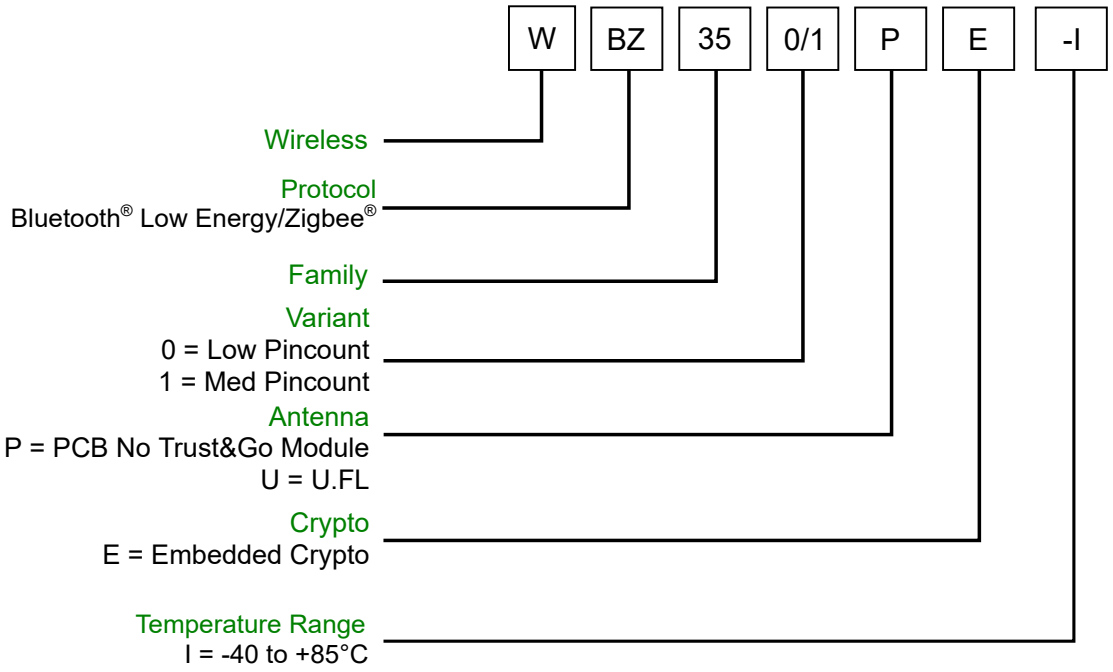
The following table describes the ordering information of the WBZ35x Module.

Table 1-2. WBZ35x Module Ordering Details

Model Name	Module SoC	Description	Regulatory Certification	Ordering Code
WBZ351PE	PIC32CX5109BZ31048-V/ZWX	Wireless module Bluetooth® LE/ZIGBEE WBZ351PE-I, with PCB Antenna, 39-LGA, 15.5 mm x 20.7 mm	FCC, ISED, CE, UKCA, MIC, KCC, NCC, SRRC	WBZ351PE-I
WBZ351UE		Wireless module Bluetooth LE/ZIGBEE WBZ351UE-I, with U.FL connector, 39-LGA, 15.5 mm x 20.7 mm	FCC, ISED, CE, UKCA, MIC, KCC, NCC, SRRC	WBZ351UE-I
WBZ350PE	PIC32CX5109BZ31032-V/ZWX	Wireless module Bluetooth LE/ZIGBEE WBZ350PE-I, with PCB Antenna, 30-LGA, 13.4 mm x 18.7 mm	FCC, ISED, CE, UKCA, MIC, KCC, NCC, SRRC	WBZ350PE-I
WBZ350UE		Wireless module Bluetooth LE/ZIGBEE WBZ350UE-I, with U.FL connector, 30-LGA, 13.4 mm x 18.7 mm	FCC, ISED, CE, UKCA, MIC, KCC, NCC, SRRC	WBZ350UE-I

The following figure illustrates the details of the WBZ35x Module ordering information.

Figure 1-2. WBZ35x Module Ordering Information



2. Configuration Summary

Table 2-1. PIC32CX-BZ3 and WBZ35x Family Features

Device	Boot ROM Memory (KB)	Program Memory (KB)	Data Memory (KB)	EFUSE (Bits)	Pins	Package	Peripherals												Analog					Security						Wireless		
							SERCOM	TC (16-bit)/Compare (Channels)	TCC (24-bit/16-bit)	QSPI	DMA Channels	RTCC	CCL/LUT	WDT	DMT	Frequency Measurement	Event System (Channels)	External Interrupt Lines	GPIO Pins	Analog Comparators/Channels	ADC (Channels)	CVD	DAC	Temperature Sensor	AES	TRNG	Public Key Cryptography	Secure Hash Function	Root-of-Trust	Max TX Power (dBm)	Bluetooth® Low Energy5.2	802.15.4/Zigbee® 3.0
PIC32CX5109BZ31048	64	512	96	3072	48	QFN	3	8/2	2/1	Y	16	Y	1/2	Y	Y	Y	32	4	27	2/4	8	8	1	Y	Y	Y	Y	Y	Y	10	Y	Y
PIC32CX5109BZ31032					32	QFN	2												14	0/0	4	4									
WBZ351					39	LGA	3												27	2/4	8	8									
WBZ350					30	LGA	2												14	0/0	4	4									

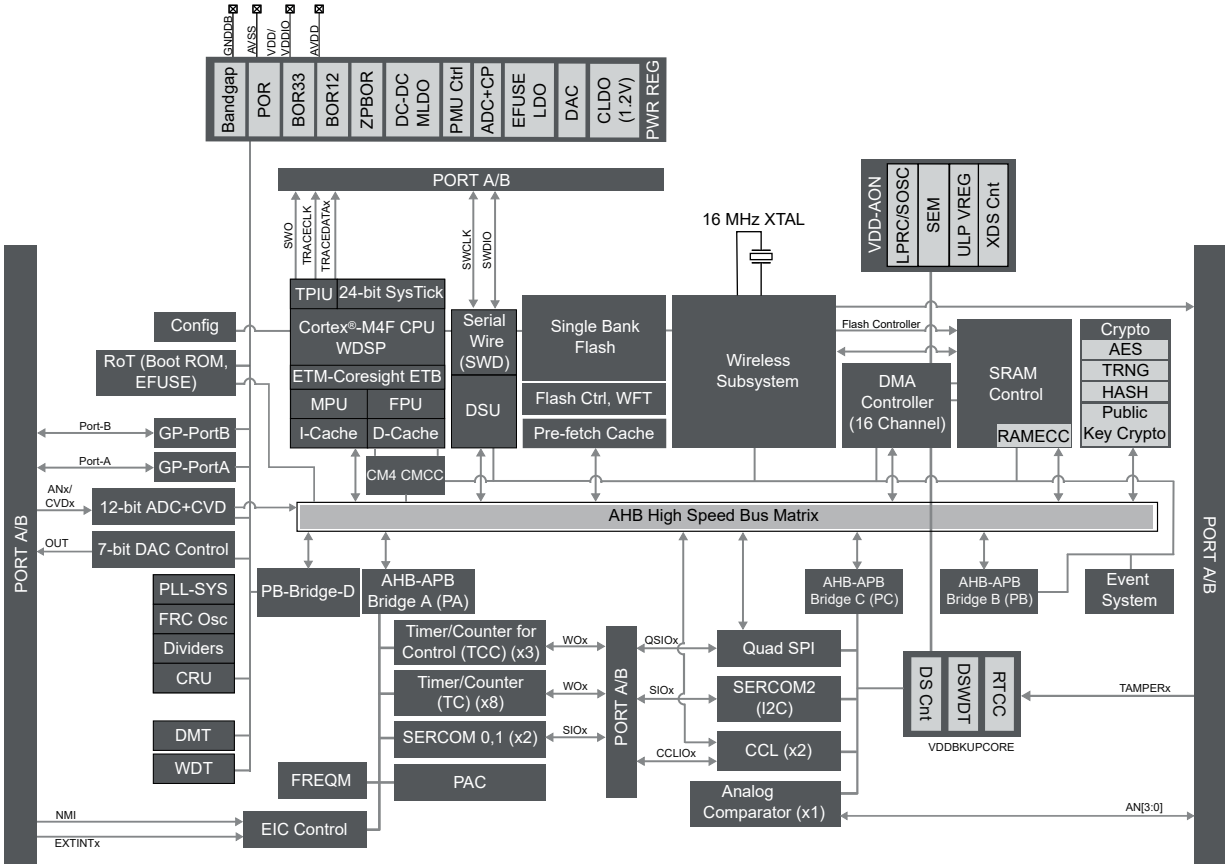
3. PIC32CX-BZ3 SoC Description

This chapter illustrates the block diagram of the PIC32CX-BZ3 SoC.

3.1 PIC32CX-BZ3 SoC Block Diagram

The following figure illustrates the block diagram of the core and peripheral modules in the PIC32CX-BZ3 SoC.

Figure 3-1. PIC32CX-BZ3 SoC Block Diagram



3.2 Pinout Diagram

This section provides details on pin diagrams and signal names along with the device pinout for each variant of PIC32CX5109BZ31048 and PIC32CX5109BZ31032 SoC.

Figure 3-2. PIC32CX5109BZ31048 SoC Pinout Diagram

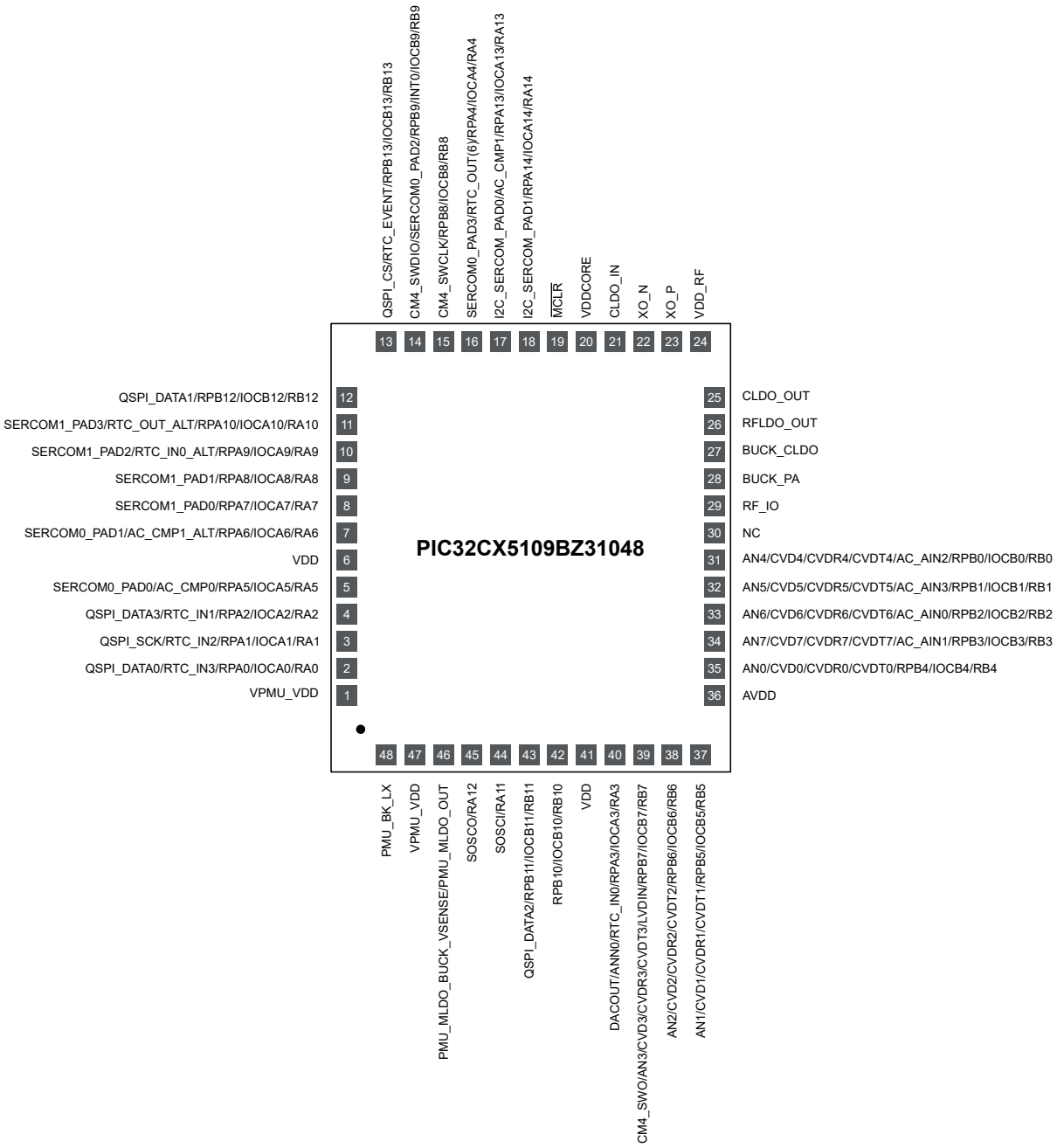
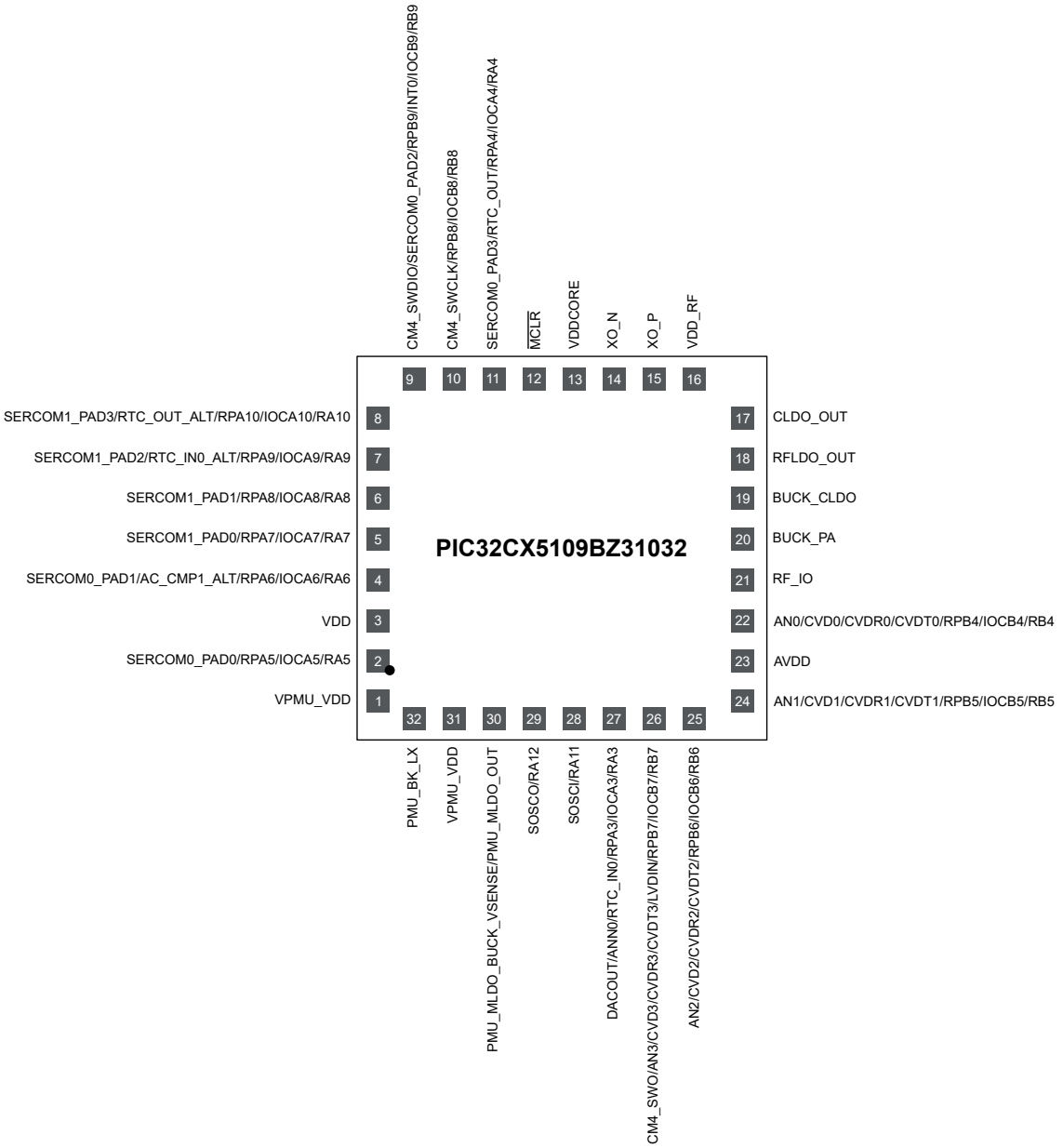


Figure 3-3. PIC32CX5109BZ31032 SoC Pinout Diagram



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

4. WBZ35x Module Description

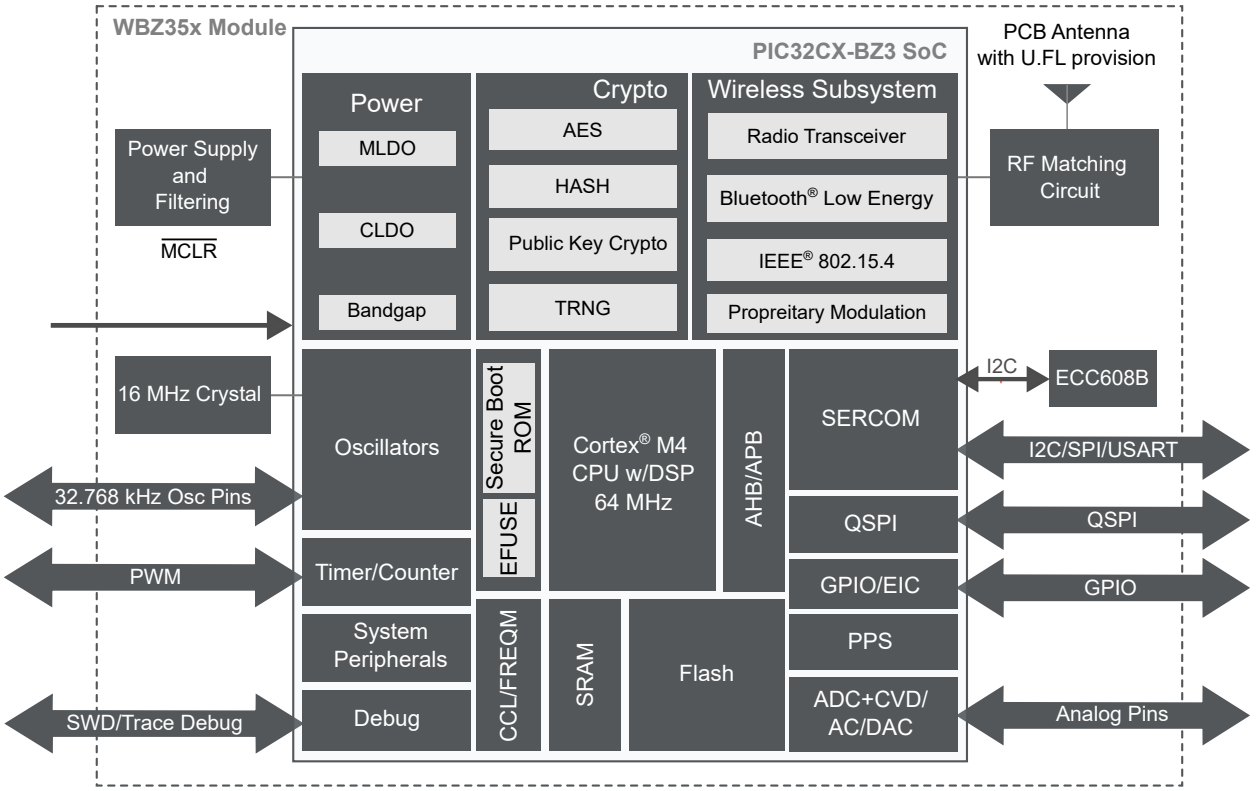
The WBZ35x modules are fully certified wireless modules built around PIC32CX-BZ3 SoC. The WBZ35x modules integrate a 16 MHz crystal and Trust&Go, in addition to the circuits for power supply decoupling, RF matching and the following antenna options:

- PCB antenna (WBZ35xPE)
- U.FL connector for external antenna (WBZ35xUE)

The Trust&GO is a pre-configured and pre-provisioned secure element of Microchip's family of security-focused devices.

The operating voltage range for the WBZ35x Module is 1.9-3.6V. The following figure illustrates the WBZ35x Module block diagram and various peripherals supported by the module.

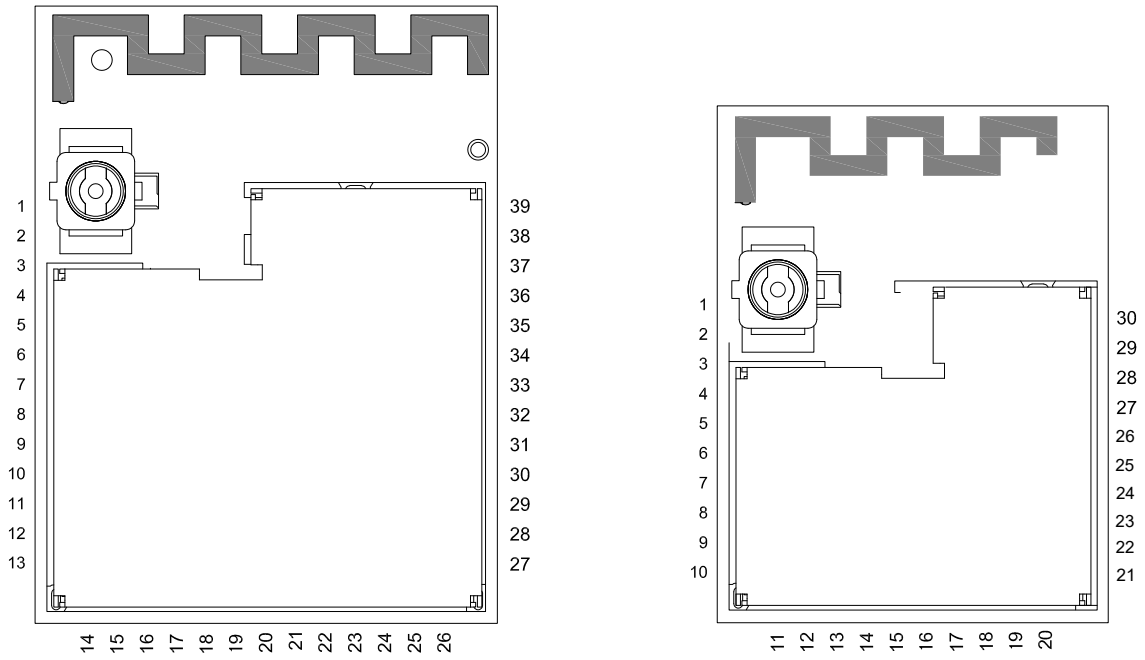
Figure 4-1. WBZ35x Module Block Diagram



4.1 Pinout Diagram

The following figure illustrates the modules pinout diagrams.

Figure 4-2. WBZ351 and WBZ350 Module Pin Diagram (Top View)



Note: Connect the exposed paddle on the bottom of the module to ground in the PCB.

See *Pinout and Signal Descriptions List* from Related Links.

Related Links

5. Pinout and Signal Descriptions List

4.2 Basic Connection Requirement

The WBZ35x modules require attention to a minimal set of device pin connections before proceeding with development.

Figure 4-3. Module Basic Connection and Interface Diagram for WBZ351 Module

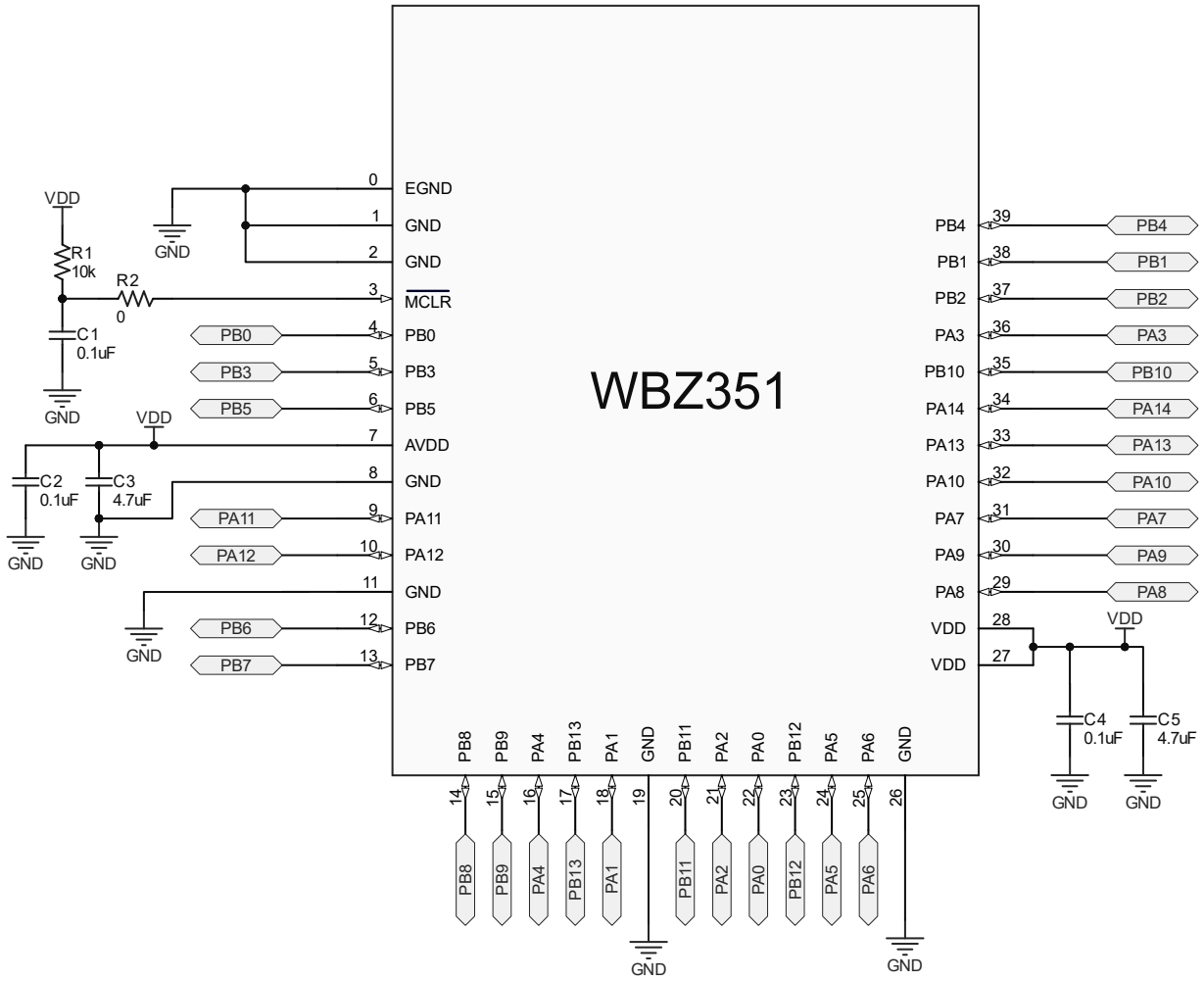
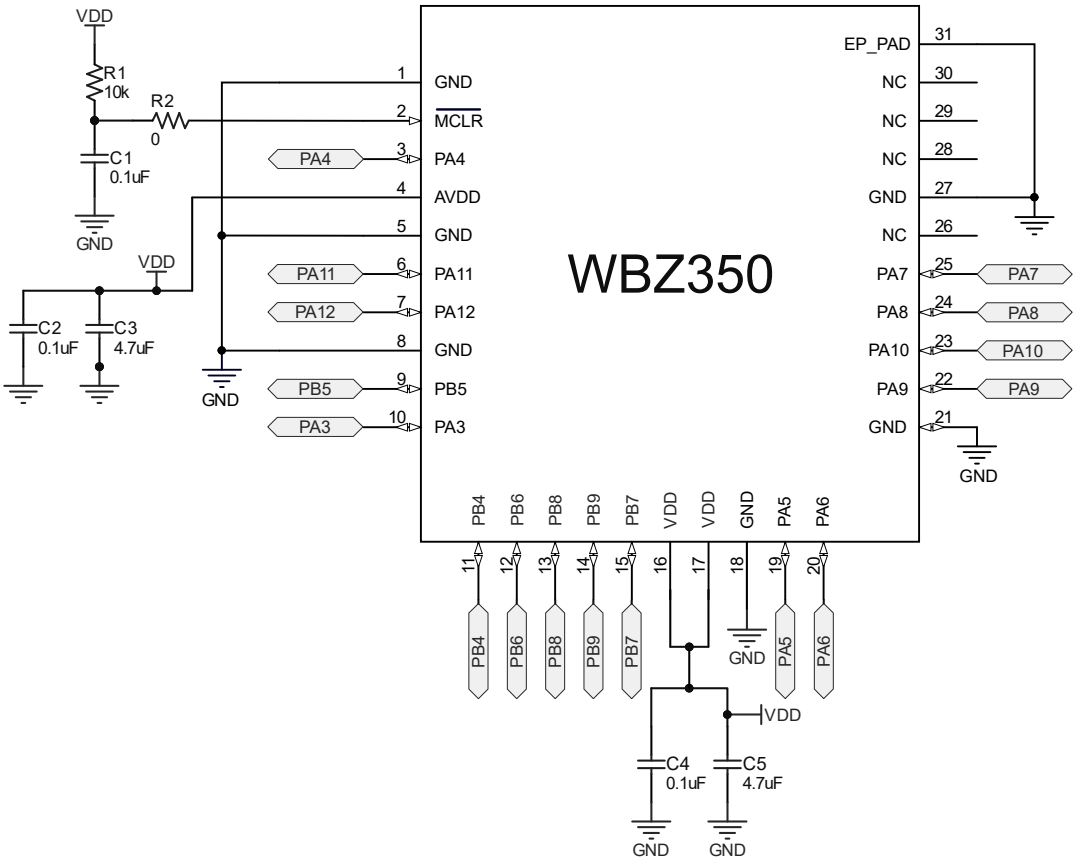


Figure 4-4. Module Basic Connection and Interface Diagram for WBZ350 Module

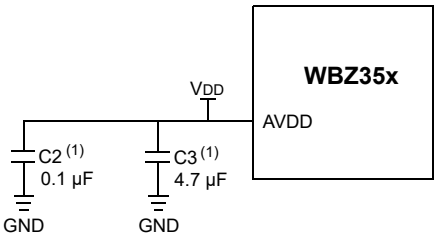


4.2.1 Power Pins

It is recommended that a bulk and a decoupling capacitor be added at the input supply pin (V_{DD}, A_{VDD} and GND pins) of the WBZ35x Module.

- The recommendation is to have a 4.7 µF on the A_{VDD} pin, 4.7 µF and a 0.1 µF on the V_{DD} pin.
- 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 user can adjust the values of the capacitor to provide a clean supply to the module.
- Place all the capacitors close to the module power supply pins.

Figure 4-5. Recommended Module Power Supply Connections



Notes:

1. The value of the C2 and C3 capacitors can vary based on the application requirement.
2. Place the C2 and C3 capacitors close to the module pin.

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

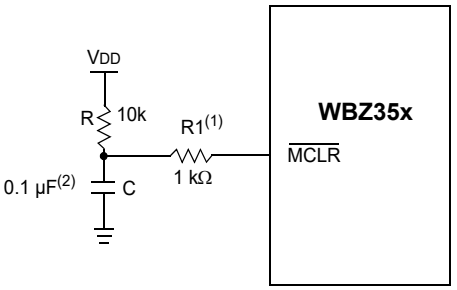
The $\overline{MCLR}$ pin provides two specific device functions:

- Device Reset
- Device programming and debugging

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

The WBZ35x module has sufficient filtering (0.1 µF) and pull-up (10k) on the Reset line. On a typical application, there is no need for extra filtering on this pin.

Figure 4-6. Example of $\overline{MCLR}$ Pin Connections



Notes:

1. 470Ω ≤ R1 ≤ 1 kΩ limits any current flowing into $\overline{MCLR}$ from the external capacitor C, in the event of $\overline{MCLR}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{MCLR}$ pin V_{IH} and V_{IL} specifications are met without interfering with the Debug/ Programmer tools.
2. The capacitor can be sized to prevent unintentional Resets from brief glitches or to extend the device Reset period during POR.

Related Links

4.2. Basic Connection Requirement

4.2.3 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 WBZ35x Module for SWD as the default configuration (CM4_SWO can be optional).

Keep the trace length between the SWD pins of the WBZ35x 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 Ωs, not to exceed 100Ω.

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

4.2.4 Unused I/O Pins

The recommendation is not to allow the unused I/O pins to float as inputs. The user can configure them as inputs and pulled up. Alternatively, depending on the application, they can be pulled down as well.

4.2.4.1 GPIO Pins/PPS Functions

Most of the WBZ35x 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 *WBZ35x Module Routing Guidelines* from Related Links.

Related Links

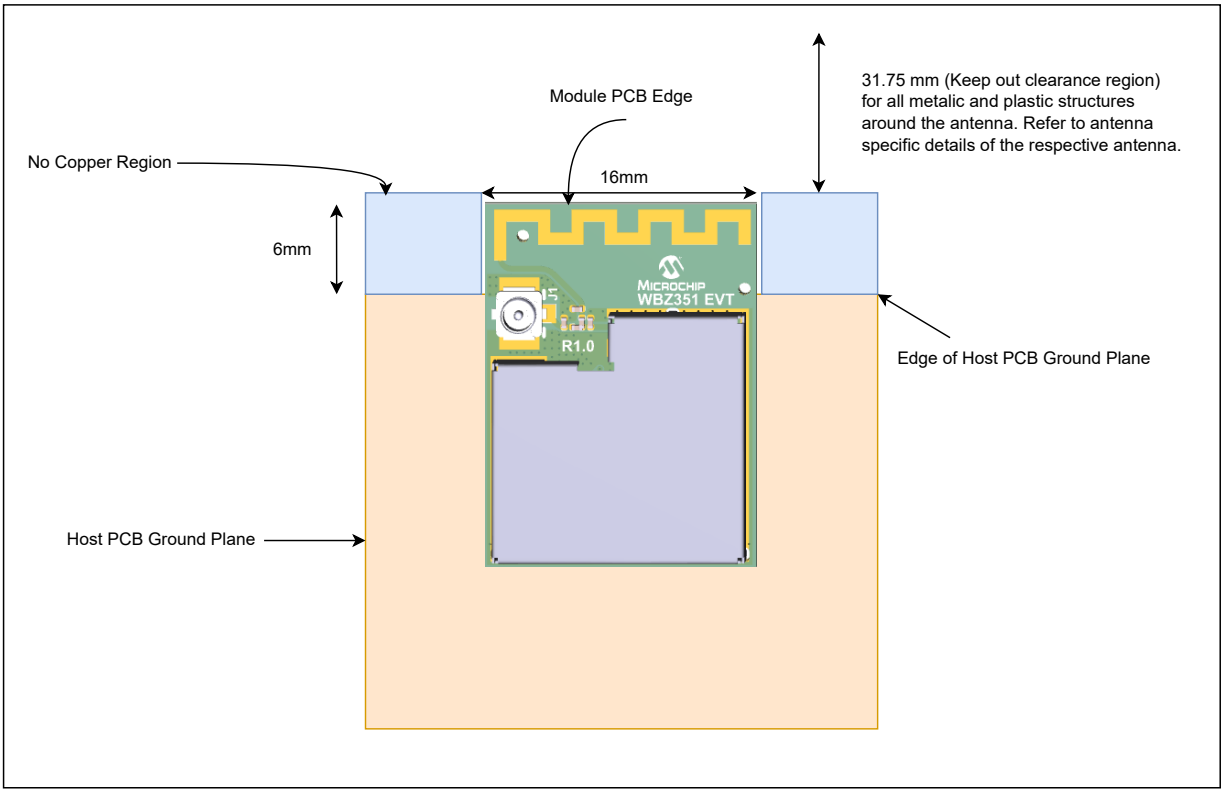
4.4. WBZ35x Module Routing Guidelines
6. I/O Ports and Peripheral Pin Select (PPS)

4.3 WBZ35x 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 ensure it is not surrounded by the ground plane. Thus, for the best PCB antenna performance, the WBZ35x Module must be placed at the edge of the host board.
- The WBZ35x Module ground outline edge must be aligned with the edge of the host board ground plane (see the following figure).
- A low-impedance ground plane for the WBZ35x 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 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 (see the following figure).
- Do not place the antenna on the WBZ35x Module in direct contact with or close proximity to plastic casing or objects. Be sure to keep a clearance of 10 mm in all directions around the PCB antenna (see 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 WBZ35x Module must be soldered to the host board (see *Example of Host Board on Top Layer* figure in the *WBZ35x Module Routing Guidelines* from Related Links).
- A PCB cutout or a copper keepout is required under RF test point. See *WBZ35x Module Packaging Information* from Related Links.
- Copper keepout areas are required on the top layer under voltage test points. See *WBZ35x Module Packaging Information* from Related Links.
- On the other hand, the entire region except the exposed ground paddle can be solder-masked.

The following figure illustrates the examples of the WBZ35x Module placement on a host board with a ground plane. Refer to the following figure for placement-specific guidance.

Figure 4-7. Module Placement Guidelines



Related Links

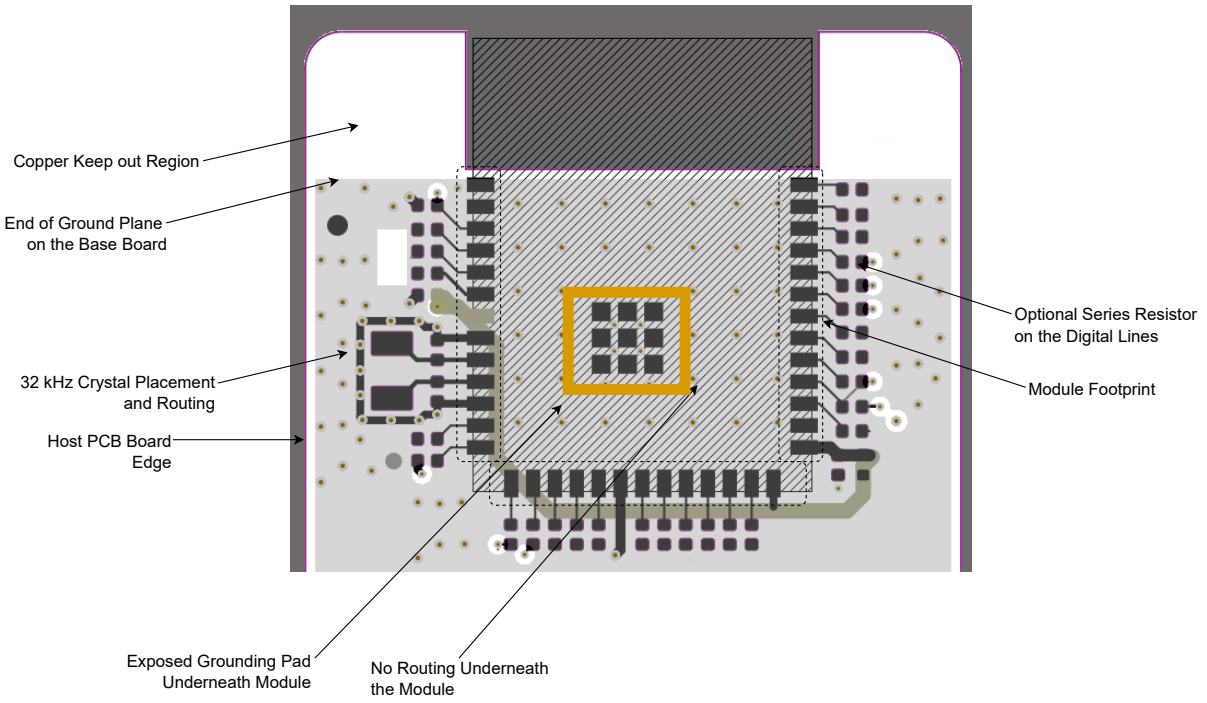
4.4. WBZ35x Module Routing Guidelines

44.2. WBZ35x Module Packaging Information

4.4 WBZ35x 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 WBZ35x Module.
- For a better GND connection to the WBZ35x Module, solder the exposed GND pads of the WBZ35x 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 WBZ35x 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 WBZ35x Module and follow the shortest trace routing length with no vias (see the following figure).

Figure 4-8. Example of Host Board on Top Layer



4.5 WBZ35x 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 WBZ35x 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 WBZ35x Module includes RF shielding on top of the board as a standard feature.

4.6 WBZ35x Module Antenna Considerations

4.6.1 PCB Antenna

For the WBZ35 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 WBZ35 module mounted on an Evaluation Board.

Table 4-1. PCB Antenna Specification for WBZ350 Module

Parameter	Specification
Operating frequency	2400 to 2500 MHz
Peak gain	3.5 dBi at 2420 MHz

.....continued	
Parameter	**Specification**
Efficiency	65%

Table 4-2. PCB Antenna Specification for WBZ351 Module

Parameter	Specification
Operating frequency	2400 to 2500 MHz
Peak gain	3.7 dBi at 2440 MHz
Efficiency	75%

PCB Antenna Radiation Pattern

The following figures illustrate the antenna radiation patterns for WBZ350 and WBZ351 modules.

Figure 4-9. WBZ350 Antenna Radiation Pattern when Phi = 0°

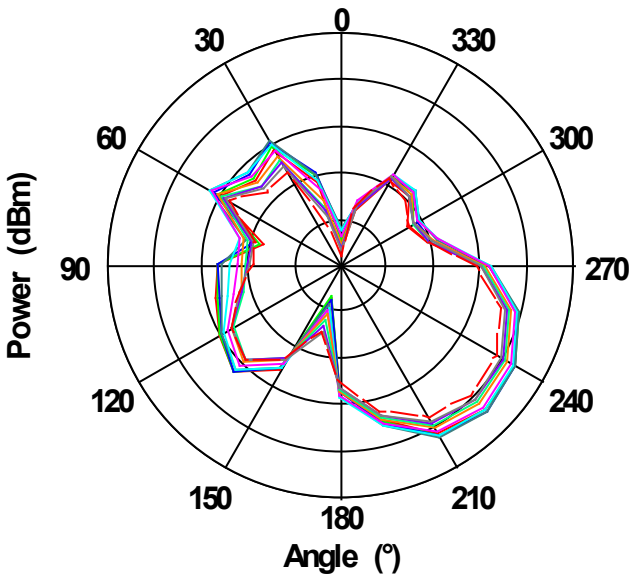


Figure 4-10. WBZ350 Antenna Radiation Pattern when Phi = 90°

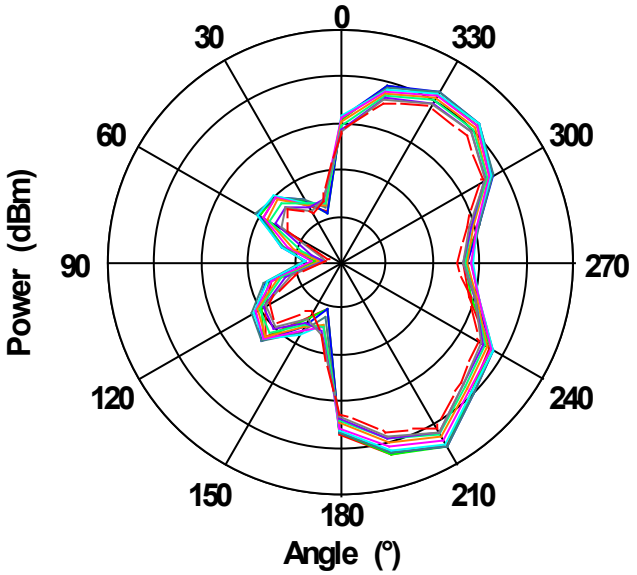


Figure 4-11. WBZ350 Antenna Radiation Pattern when Theta = 90°

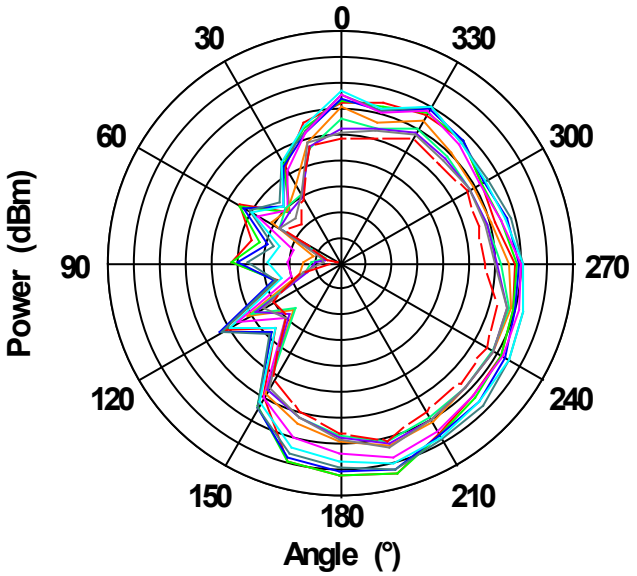


Figure 4-12. WBZ350 Antenna Radiation 3D Pattern

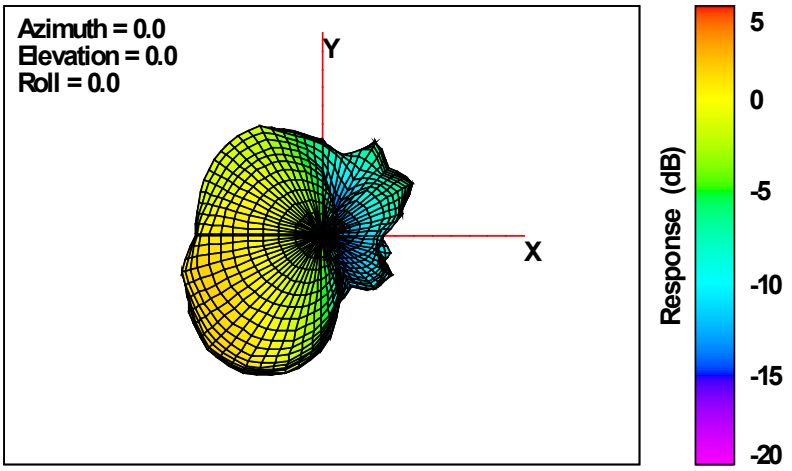


Figure 4-13. WBZ351 Antenna Radiation Pattern when Phi = 0°

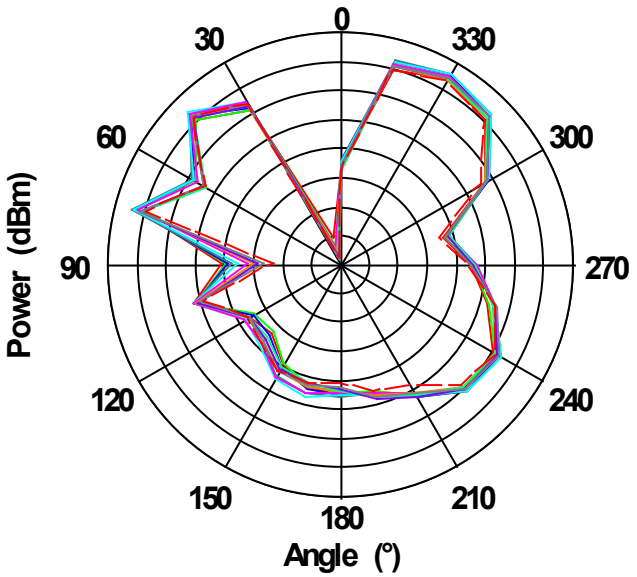


Figure 4-14. WBZ351 Antenna Radiation Pattern when Phi = 90°

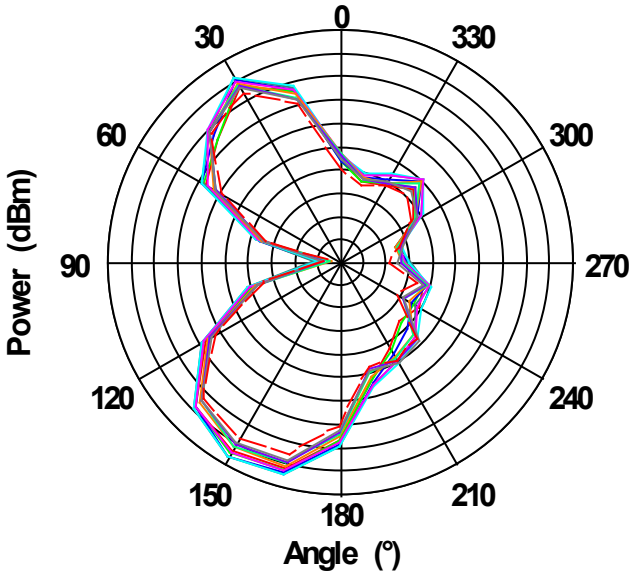


Figure 4-15. WBZ351 Antenna Radiation Pattern when Theta = 90°

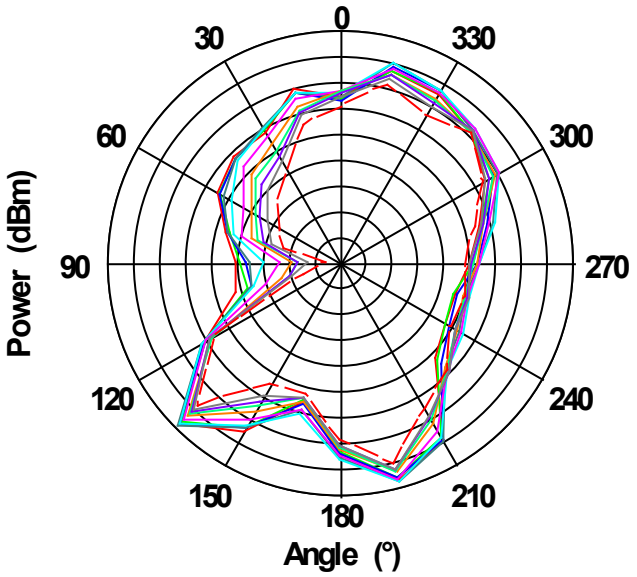
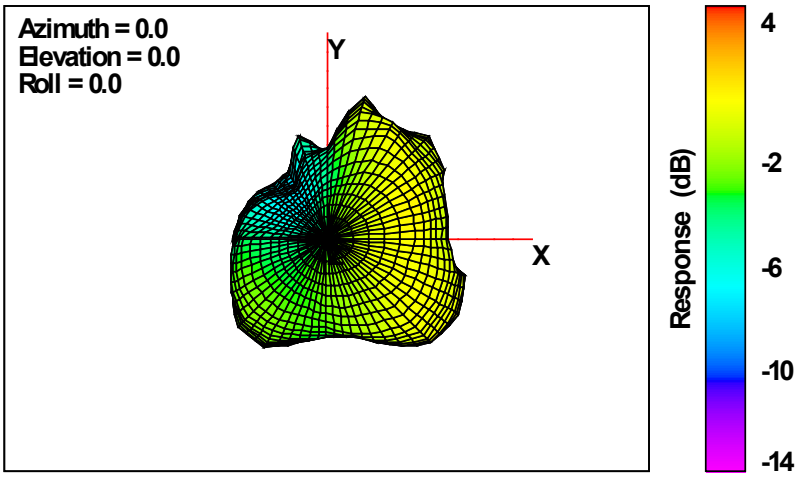


Figure 4-16. WBZ351 Antenna Radiation 3D Pattern



4.6.2 External Antenna Placement Recommendations

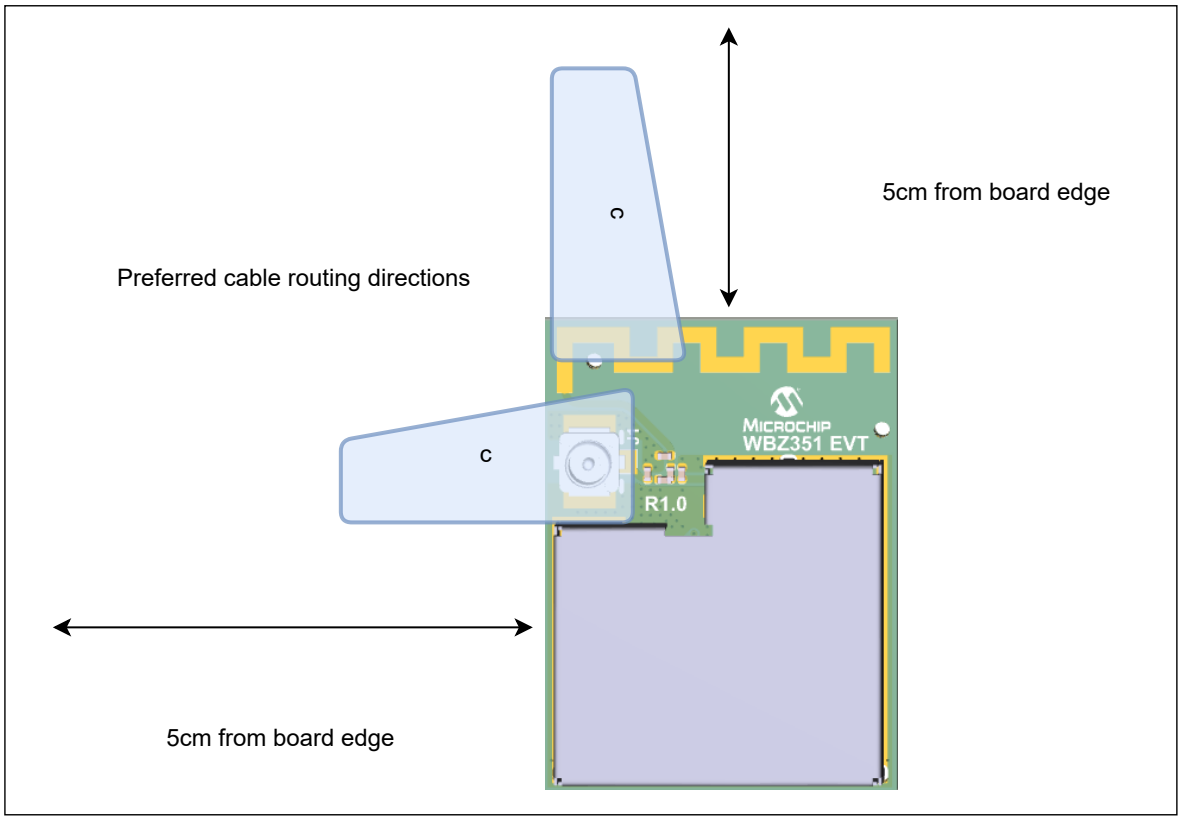
The user must apply the following recommendations for the placement of the antenna and its cable:

- Do not route the antenna cable over circuits generating electrical noise on the host board or alongside or underneath the module. The recommendation is to route the cable straight out of the module.
- Do not place the antenna in direct contact with or in close proximity to the plastic casing/objects. (Except when the selected antenna specifically recommends it.)
- Do not enclose the antenna within a metal shield.
- The user must keep any components capable of radiating noise, signals or harmonics in the 2.4-2.5 GHz frequency range away from the antenna, and, if feasible, provide shielding for such components. Any noise radiated from the host board in this frequency band degrades the sensitivity of the module.
- Place the antenna at a distance greater than 5 cm away from the module. The following figure illustrates the antenna keep-out 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 WBZ35x Module.

The following figure illustrates an indication on how to route the antenna cable depending on the location of the antenna with respect to the WBZ35x PCB. There are two possible options for the optimum routing of the cable.

Figure 4-17. WBZ35x Antenna Placement Guidelines



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

4.7 WBZ35x Module Reflow Profile Information

The WBZ35x Module was assembled using the IPC/JEDEC J-STD-020 standard lead-free reflow profile. The WBZ35x 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 WBZ35x Module Assembly Considerations

The WBZ35x 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

This following table provides detailed signal names classified by the peripherals along with the device pinout for each variant of the PIC32CX-BZ3 and the WBZ35x Module.

Table 5-1. Pinout and Signal Descriptions List

SoC		Module		Pin Name(1)(2)(5)
PIC32CX5109BZ31032	PIC32CX5109BZ31048	WBZ351	WBZ350	
0(3)	0(3)	0(3), 1, 2, 8, 11, 19, 26	0(3), 1, 5, 8, 18, 21, 27	GND
1	1	—	—	VPMU_VDD (PMU Core Circuit Power (1.9V-3.6V) Filtered version of main supply input); connect 100 nF decoupling capacitor
—	2	22	—	QSPI_DATA0/RTC_IN3/RPA0/IOCA0/RA0
—	3	18	—	QSPI_SCK/RTC_IN2/RPA1/IOCA1/RA1
—	4	21	—	QSPI_DATA3/RTC_IN1/RPA2/IOCA2/RA2
2	5	24	19	SERCOM0_PAD0/AC_CMP0/RPA5/IOCA5/RA5
3	6	27, 28	16, 17	VDD (filtered version of main supply); connect 100 nF decoupling capacitor
4	7	25	20	TRD3/SERCOM0_PAD1/AC_CMP1_ALT/RPA6/IOCA6/RA6
5	8	31	25	TRCLK/SERCOM1_PAD0/RPA7/IOCA7/RA7
6	9	29	24	SERCOM1_PAD1/RPA8/IOCA8/RA8
7	10	30	22	SERCOM1_PAD2/RTC_IN0_ALT/RPA9/IOCA9/RA9
8	11	32	23	SERCOM1_PAD3/RTC_OUT_ALT/RPA10/IOCA10/RA10
—	12	23	—	QSPI_DATA1/RPB12/IOCB12/RB12
—	13	17	—	QSPI_CS/RTC_EVENT(5)/RPB13/IOCB13/RB13
9	14	15	14	CM4_SWDIO/SERCOM0_PAD2/RPB9/INT0/IOCB9/RB9
10	15	14	13	CM4_SWCLK/RPB8/IOCB8/RB8
11	16	16	3	SERCOM0_PAD3/RTC_OUT(5)/RPA4/IOCA4/RA4
—	17	33	—	SERCOM2_PAD0(6)/AC_CMP1/RPA13/IOCA13/RA13
—	18	34	—	SERCOM2_PAD1(6)/RPA14/IOCA14/RA14
12	19	3	2	NMCLR; device reset
13	20	—	—	VDDCORE (Secondary digital power supply) (1.2V ± 5%); connect to CLDO_OUT with 1 uF to ground
—	21	—	—	CLDO_IN (Secondary digital power supply) (1.2V ± 5%); connect to CLDO_OUT with 1 uF to ground
14	22	—	—	XO_N
15	23	—	—	XO_P

.....continued

SoC		Module		Pin Name(1)(2)(5)
PIC32CX5109BZ31032	PIC32CX5109BZ31048	WBZ351	WBZ350	
16	24	—	—	VDD_RF (RF block power supply, to be connected to RFLDO_OUT pin); with 1 uF to ground
17	25	—	—	CLDO_OUT (Core LDO power supply 1.2V ± 5%); connect 1 uF decoupling capacitor
18	26	—	—	RFLDO_OUT (RF LDO power supply (1.2V), to be connected to VDD_RF pin; connect 1 uF decoupling capacitor
19	27	—	—	BUCK_CLDO (RF/DIG CLDO supply); filtered version of PMU output; connect 1 uF decoupling capacitor
20	28	—	—	BUCK_PA (RF PA supply); filtered version of PMU output; connect 1 uF decoupling capacitor
21	29	—	—	RF_IO
—	30	—	—	NC
—	31	4	—	AN4/CVD4/CVDR4/CVDT4/AC_AIN2/RPB0/IOCB0/RB0
—	32	38	—	AN5/CVD5/CVDR5/CVDT5/AC_AIN3/RPB1/IOCB1/RB1
—	33	37	—	AN6/CVD6/CVDR6/CVDT6/AC_AIN0/RPB2/IOCB2/RB2
—	34	5	—	AN7/CVD7/CVDR7/CVDT7/AC_AIN1/RPB3/IOCB3/RB3
22	35	39	11	AN0/CVD0/CVDR0/CVDT0/RPB4/IOCB4/RB4(7)
23	36	7	4	AVDD (VDDA)
24	37	6	9	AN1/CVD1/CVDR1/CVDT1/RPB5/IOCB5/RB5(7)
25	38	12	12	AN2/CVD2/CVDR2/CVDT2/RPB6/IOCB6/RB6(7)
26	39	13	15	CM4_SWO/AN3/CVD3/CVDR3/CVDT3/LVDIN/RPB7/IOCB7/RB7(7)
27	40	36	10	SCLKI/DACOUT/ANN0/RTC_IN0/RPA3/IOCA3/RA3
—	41	—	—	VDD; filtered version of main supply; connect 100 nF decoupling capacitor
—	42	35	—	RPB10/IOCB10/RB10
—	43	20	—	QSPI_DATA2/RPB11/IOCB11/RB11
28	44	9	6	SOSCI/RA11(4)
29	45	10	7	SOSCO/RA12(4)
30	46	—	—	PMU_MLDO_BUCK_VSENSE/PMU_MLDO_OUT (PMU Output – MLDO_O)
31	47	—	—	VPMU_VDD; filtered version of main supply input; connect 100 nF decoupling capacitor
32	48	—	—	PMU_BK_LX; connect 4.7 uH shielded inductor with 10 uF and 100 nF decoupling capacitor

.....continued

SoC		Module		Pin Name(1)(2)(5)
PIC32CX5109BZ31032	PIC32CX5109BZ31048	WBZ351	WBZ350	

Notes:

1. The remappable peripherals can use all GPIOs (RAn and RBn) via PPS.
2. All GPIOs (RAn and RBn) can be used as I/O Change Notification (IOCAn and IOCBn).
3. Connect the metal paddle at the bottom of the device to the system ground.
4. If not using SOSC, use this pin as an input-only pin.
5. The RTC_OUT and RTC_EVENT signals are multiplexed. Any one of the signals can be out at a time in pin-limited variants (32-pin variant).
6. The SERCOM2 has only I^2C functionality. The SERCOM2_PAD0 is I2C_SERCOM_SDA and the SERCOM2_PAD1 is I2C_SERCOM_SCL.
7. JTAG is the default functionality on these pins. The recommendation is to write '0' to the CFGCON0.JTAGEN register during application initialization to use these pins for regular GPIO functionality. See the *CFGCON0* register from Related Links.
8. A pull-up resistor (1K) on the SWCLK pin is critical for reliable operation.
9. For more details on power supply pins connections and the filtering components, refer to the Design Package available at the product page.

Related Links

22.9.1. CFGCON0(L)

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

6.1 Overview

General purpose I/O (GPIO) pins allow the PIC32CX-BZ3 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, do not use that pin 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.
Note: The fuse values stored in Boot Flash memory (NVR) can be used to alter the power-on default function for certain GPIO pin. See *System Configuration and Register Locking (CFG)* from Related Links.

Related Links

4.2.4.1. GPIO Pins/PPS Functions
6.7. Function Priority for Device Pins
22. System Configuration and Register Locking (CFG)

6.2 Features

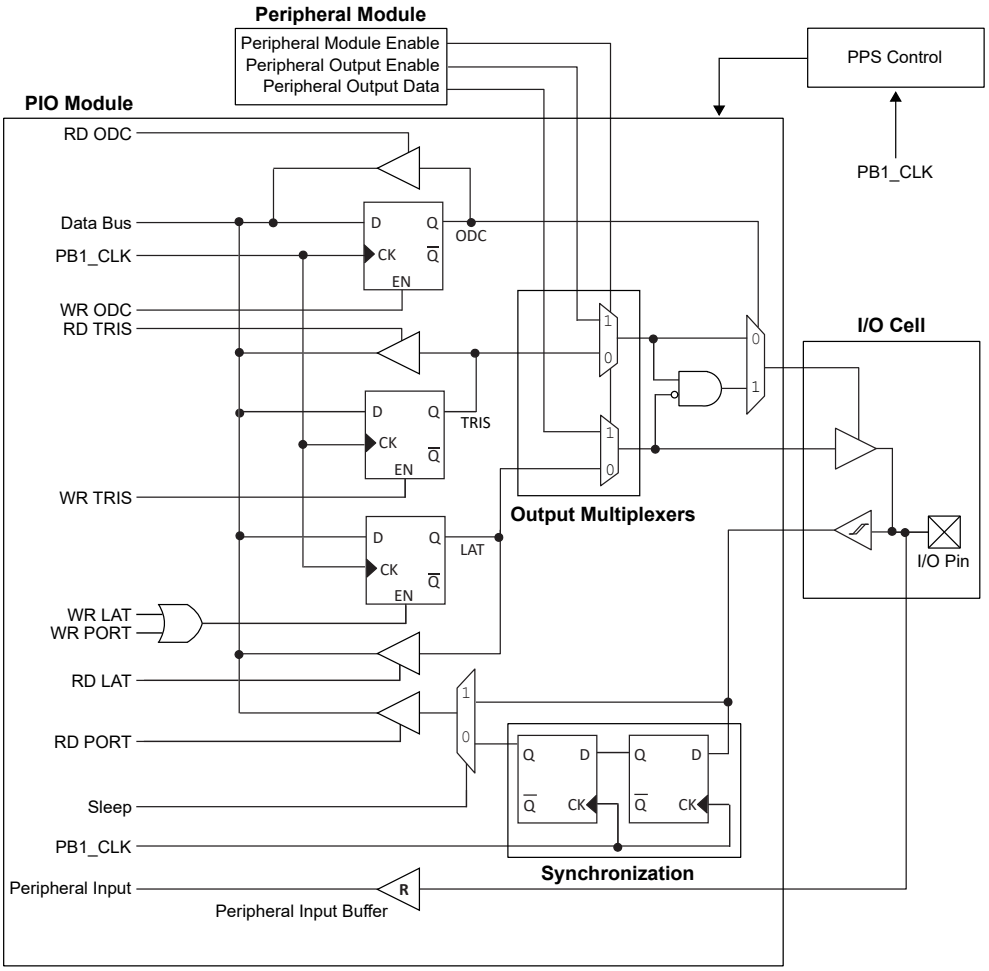
The following are some of the key features of the I/O ports:

- Individual Output Pin Open-Drain Enable/Disable
- Individual Input Pin Weak Pull-Up and Pull-Down
- Monitoring Selective Inputs and Generate Interrupt After Detecting Change in Pin State
- Operation During Sleep and Idle Modes
- Fast Bit Manipulation Using CLR, SET and INV Registers
- Slew Rate Control (Not Available on All Devices)
- Flexibility for Peripheral Pin Remapping Through PPS

6.3 Block Diagram

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

Figure 6-1. Typical Multiplexed Port Structure Block Diagram



6.4 Parallel I/O (PIO) Ports

Each I/O port has 14 registers directly associated with the operation of the port. Each I/O port pin has a corresponding bit in these registers. The data direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. 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 are not valid for a particular device are disabled and read as zeros.

6.4.1 I/O Ports Configuration

6.4.1.1 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.

6.4.1.2 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.

6.4.1.3 Open-Drain Configuration (ODCx)

The user can individually configure each I/O pin 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', configure the pin 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 can 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 I^2C pins). I^2C pins are already open-drain pins; therefore, the ODCx settings do not influence the I^2C pins.

6.4.1.4 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.

6.4.1.5 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).

6.4.1.6 I/O Port Write/Read Timing

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically this instruction is an NOP.

6.4.1.7 Input Change Notification (CN)

The Input Change Notification (CN) function of the I/O ports allows PIC32CX-BZ3 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.

6.4.1.7.1 CN Configuration and Operation

The CN pins are configured as follows:

1. Disable the CPU interrupts.
2. Set the desired CN I/O pin as an input by setting the corresponding TRISx register bits = `1`.
3. Enable the CN Module by setting the ON bit (CNCONx[15]) = `1`.
4. Enable the individual CN input pins, and optional pull-ups or pull-downs.
5. Read the corresponding PORTx registers to clear the CN interrupt.
6. Configure the CN Interrupt Priority bits, NVIC.IP register.
7. Clear the CN Interrupt Flag bit, by setting the corresponding CLRPEND bit in the NVIC.IPCR register.
8. Configure the CN pin interrupt for a specific edge detect using the EDGEDETECT bit in the CNCONx register, and set up edge control using the CNENx/CNNEx bits.
9. Enable the CN Interrupt Enable bit by setting the corresponding enable bit in the NVIC.ISER register.
10. Enable CPU interrupts.

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. When a CN interrupt occurs in the Mismatch mode, the user must read the PORTx register associated with the CN pins. This will clear the mismatch condition and set up the CN logic to detect the next pin change. The current PORTx value can be compared to the PORTx read value obtained at the last CN interrupt or during initialization and used to determine which pin changed. The CN pins have a minimum input pulse-width specification. See *Electrical Characteristics* from Related Links.

Related Links

43. Electrical Characteristics

6.4.1.8 Slew Rate Control

Some I/O pins can be configured for various types of slew rate control on their 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.

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

Note:
1. By default, all the port pins are set to the fastest edge rate.

6.4.1.9 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`.

6.5 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.

6.5.1 Remappable Pin Groupings

The remappable pins, as well as the available input and output functions, are divided into five groups. The remappable pins of Group k can be assigned pin functions only from Group k, k = 1, 2, 3, 4, 5. The pins used by each peripheral are spread across all five groups when possible to maximize flexibility.

6.5.2 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

6.5.3 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 I^2C 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. 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.

6.5.4 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.

6.5.5 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.

6.5.5.1 Remappable Inputs

6.5.5.1.1 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*, Input Pin Selection Group 5 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. In general, all functions other than GPIO are considered higher priority than remap pins. For the list and priority of pin functions on each pin, see *Function Priority for Device Pins* from Related Links.

When transitioning the pin usage from one peripheral to another, it is mandatory to make sure the pins are not configured for any other peripheral. Therefore, 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 be safely performed after any device Reset. After Reset, all inputs are mapped to a default value and all outputs are disabled, the user can safely perform the mapping after any device Reset.

Related Links

6.5.5.1.3. Remappable Input Example

6.14.1. [pin name]R

6.7. Function Priority for Device Pins

6.5.5.1.2 Remappable Input Priority

The user can select only one single pin for any of the remappable peripheral inputs. There is no need for priority encoding 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 a different pin variant, repeat the same input functions in multiple groups. Therefore, to differentiate between them, the 'Off' code is provided in the *Input Pin Selection Group* tables. See these tables in the *Remappable Input Example* from Related Links.

For proper operation, the software must ensure that the unused group register offset of a (repeated) input function is programmed to `4'h0`. Failure to do so leads to unknown behavior.

Related Links

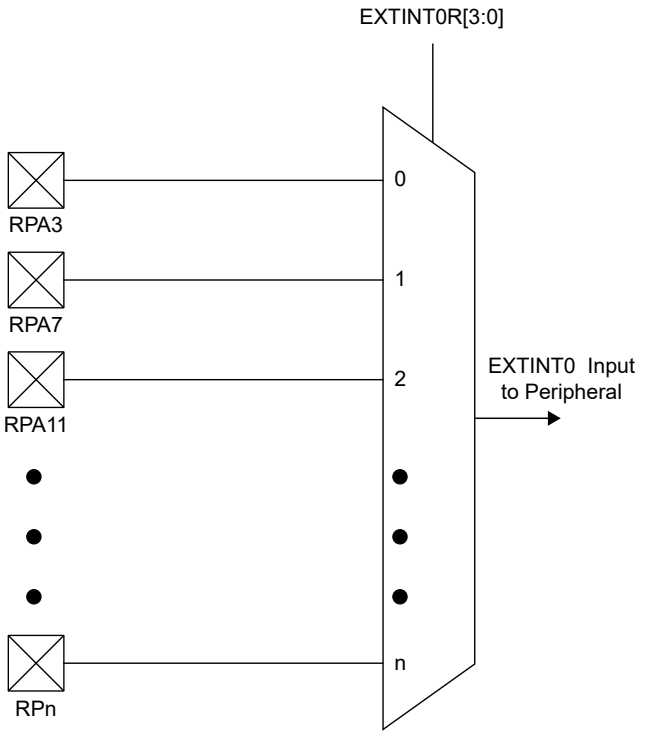
6.5.5.1.3. Remappable Input Example

6.5.5.1.3 Remappable Input Example

The following figure illustrates the remappable pin selection for the EXTINT0 input. In order to remap the EXTINT0 input to a particular pin, program the EXTINT0R remap register. 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 EXTINT0 to RPB0, program the value 4 (`4'b0100`) into the EXINTR0R SFR register. See the *Input Pin Selection Group 1* table in the *Input Mapping in PIC32CX-BZ3 Family of Devices* from Related Links.

Figure 6-2. EXTINT0 Remappable Pin Selection



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

Related Links

6.5.5.1.4. Input Mapping in PIC32CX-BZ3 Family of Devices

6.5.5.1.4 Input Mapping in PIC32CX-BZ3 Family of Devices

The following tables provide input mapping in PIC32CX-BZ3 family of devices

Table 6-2. Input Pin Selection Group 1

Peripheral Pin (pin name)	[*pin name*]R SFR	[*pin name*]R bits	[*pin name*]R Value to RPn Pin Selection
EXTINT0	EXTINT0R	EXTINT0R[3:0]	0000 = OFF
SERCOM0_PAD3	SCOM0P3R	SCOM0P3R[3:0]	0001 = RPA3
SERCOM1_PAD2	SCOM1P2R	SCOM1P2R[3:0]	0010 = RPA7
QD1	QD1R	QD1R[3:0]	0011 = RPA11
CCLIN0	CCLIN0R	CCLIN0R[3:0]	0100 = RPB0(1)
CCLIN3	CCLIN3R	CCLIN3R[3:0]	0101 = RPB4
TC0_WO0G1	TC0WO0G1R	TC0WO0G1R[3:0]	0110 = RPB8
TC1_WO0G1	TC1WO0G1R	TC1WO0G1R[3:0]	0111 = RPB12(1)
TC2_WO0G1	TC2WO0G1R	TC2WO0G1R[3:0]	1000 = RPA2(1)
TC3_WO0G1	TC3WO0G1R	TC3WO0G1R[3:0]	1001 = RPA6
TC4_WO0G1	TC4WO0G1R	TC4WO0G1R[3:0]	1010 = RPA10
TC5_WO0G1	TC5WO0G1R	TC5WO0G1R[3:0]	1011 = RPA14
TC6_WO0G1	TC6WO0G1R	TC6WO0G1R[3:0]	1100 = RPB3
TC7_WO0G1	TC7WO0G1R	TC7WO0G1R[3:0]	1101 = RPB7
			1110 = RPB11
			1111 = RPA9

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-3. Input Pin Selection Group 2

Peripheral Pin (pin name)	[*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
QD2	QD2R	QD2R[3:0]	0011 = RPA12
CCLIN1	CCLIN1R	CCLIN1R[3:0]	0100 = RPB1(1)
CCLIN4	CCLIN4R	CCLIN4R[3:0]	0101 = RPB5
TC0_WO0G2	TC0WO0G2R	TC0WO0G2R[3:0]	0110 = RPB9
TC1_WO1G2	TC1WO1G2R	TC1WO1G2R[3:0]	0111 = RPB13(1)
TC2_WO1G2	TC2WO1G2R	TC2WO1G2R[3:0]	1000 = RPA3
TC3_WO1G2	TC3WO1G2R	TC3WO1G2R[3:0]	1001 = RPA7
TC4_WO1G2	TC4WO1G2R	TC4WO1G2R[3:0]	1010 = RPA11
TC5_WO1G2	TC5WO1G2R	TC5WO1G2R[3:0]	1011 = RPB0(1)
TC6_WO1G2	TC6WO1G2R	TC6WO1G2R[3:0]	1100 = RPB4
TC7_WO1G2	TC7WO1G2R	TC7WO1G2R[3:0]	1101 = RPB8
			1110 = RPB12(1)
			1111 = RPA0(1)

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-4. Input Pin Selection Group 3

Peripheral Pin (pin name)	[*pin name*]R SFR	[*pin name*]R bits	[*pin name*]R Value to RPn Pin Selection
EXTINT2	EXTINT2R	EXTINT2R[3:0]	0000 = OFF
SERCOM1_PAD0	SCOM1P3R	SCOM1P3R[3:0]	0001 = RPA5
QD3	QD3R	QD3R[3:0]	0010 = RPA9
CCLIN2	CCLIN2R	CCLIN2R[3:0]	0011 = RPA13(1)
CCLIN5	CCLIN5R	CCLIN5R[3:0]	0100 = RPB2*
TC0_WO1G3	TC0WO1G3R	TC0WO1G3R[3:0]	0101 = RPB6
TC1_WO0G3	TC1WO0G3R	TC1WO0G3R[3:0]	0110 = RPB10(1)
TC2_WO0G3	TC2WO0G3R	TC2WO0G3R[3:0]	0111 = RPA0(1)
TC3_WO0G3	TC3WO0G3R	TC3WO0G3R[3:0]	1000 = RPA4
TC4_WO0G3	TC4WO0G3R	TC4WO0G3R[3:0]	1001 = RPA8
TC5_WO0G3	TC5WO0G3R	TC5WO0G3R[3:0]	1010 = RPA12
TC6_WO0G3	TC6WO0G3R	TC6WO0G3R[3:0]	1011 = RPB1(1)
TC7_WO0G3	TC7WO0G3R	TC7WO0G3R[3:0]	1100 = RPB5
			1101 = RPB9
			1110 = RPB13(1)
			1111 = RPA1(1)

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-5. Input Pin Selection Group 4

Peripheral Pin (pin name)	[*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
QD0	QD0R	QD0R[3:0]	0011 = RPA14(1)
TC0_WO1G4	TC0WO1G4R	TC0WO1G4R[3:0]	0100 = RPB3(1)
TC2_WO1G4	TC2WO1G4R	TC2WO1G4R[3:0]	0101 = RPB7
TC3_WO1G4	TC3WO1G4R	TC3WO1G4R[3:0]	0110 = RPB11(1)
TC4_WO1G4	TC4WO1G4R	TC4WO1G4R[3:0]	0111 = RPA1(1)
TC5_WO1G4	TC5WO1G4R	TC5WO1G4R[3:0]	1000 = RPA5
TC6_WO1G4	TC6WO1G4R	TC6WO1G4R[3:0]	1001 = RPA9
TC7_WO1G4	TC7WO1G4R	TC7WO1G4R[3:0]	1010 = RPA13(1)
			1011 = RPB2(1)
			1100 = RPB6
			1101 = RPB10(1)
			1110 = RPA8
			1111 = RPA2(1)

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-6. Input Pin Selection Group 5

Peripheral Pin (pin name)	[*pin name*]R SFR	[*pin name*]R bits	[*pin name*]R Value to RPn Pin Selection
SERCOM0_PAD1	SCOM0P1R	SCOM0P1R[3:0]	0000 = OFF
SERCOM1_PAD1	SCOM1P1R	SCOM1P1R[3:0]	0001 = RPA3
REFI	REFIR	REFIR[3:0]	0010 = RPA5
			0011 = RPA6
			0100 = RPA7
			0101 = RPB5
			0110 = RPB6
			0111 = RPB8
			1000 = RPB9
			1001 = OFF
			1010 = OFF
			1011 = OFF
			1100 = OFF
			1101 = OFF
			1110 = OFF
			1111 = OFF

6.5.5.2 Remappable Output

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

Each remappable output can be programmed to an output function that is from its same output group number. For example, if RPA0 is part of Group 2, then the user can program it to have any Group 2 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, 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 – Group 1*, *Remappable Output Pin Configuration – Group 2*, *Remappable Output Pin Configuration – Group 3* and *Remappable Output Pin Configuration – Group 4* tables in the *Output Mapping in PIC32CX-BZ3 Family of Devices* from Related Links.

The user must follow the rules for which group belong to which pin such that multiple peripherals are not driving the same pin from different groups. For instance, pin RPA0 (PA0) as an output belongs to Group 2 and Group 3. If the peripheral driving the signal to RPA0 is coming from Group 2, the software must ensure that all Group 3 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'. By default, do this to ensure that remappable outputs remain disconnected from all output pins.

Related Links

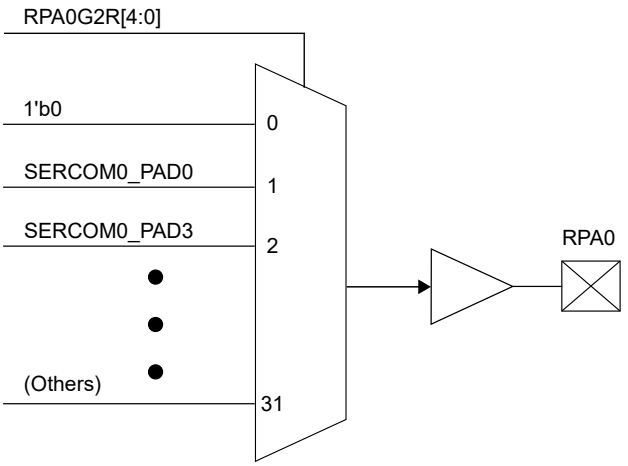
6.5.5.2.2. Output Mapping in PIC32CX-BZ3 Family of Devices

6.5.5.2.1 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. For a complete list of output function values and their associated register names, 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 *Output Mapping in PIC32CX-BZ3 Family of Devices* from Related Links.

When transitioning the pin usage from one peripheral to another, it is mandatory to make sure the pins are not configured for any other peripheral. Therefore, to avoid glitching outputs, the user is responsible for turning off the appropriate peripherals before remapping the pin functions associated with that peripheral. After Reset, map all inputs to a default value and disable all outputs. The user must perform the mapping after any device Reset.

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



Related Links

6.5.5.2.2. Output Mapping in PIC32CX-BZ3 Family of Devices
6.14.2. RPnR

6.5.5.2.2 Output Mapping in PIC32CX-BZ3 Family of Devices

The following tables provide output mapping in PIC32CX-BZ3 family of devices.

Table 6-7. PPS Output Groups

Group 1	Group 2	Group 3	Group 4
OFF	OFF	OFF	OFF
SERCOM0_PAD1	SERCOM0_PAD0	OFF	SERCOM0_PAD2
SERCOM1_PAD1	SERCOM0_PAD3	SERCOM0_PAD0	OFF
REFO1	SERCOM0_PAD2	SERCOM0_PAD3	SERCOM0_PAD0
REFO2	OFF	SERCOM1_PAD2	SERCOM1_PAD3
REFO3	SERCOM1_PAD3	OFF	OFF
REFO4	SERCOM1_PAD2	SERCOM1_PAD3	SERCOM1_PAD0
QSPI_SCK	TCC0_WO1	TCC0_WO2	TCC0_WO3
OFF	TCC0_WO5	TCC0_WO0	TCC0_WO1
OFF	TCC0_WO3	TCC0_WO4	TCC0_WO5
OFF	TCC1_WO1	TCC1_WO2	TCC1_WO3
OFF	TCC1_WO5	TCC1_WO0	TCC1_WO1
OFF	TCC1_WO3	TCC1_WO4	TCC1_WO5
OFF	TCC2_WO1	TCC2_WO0	TCC2_WO1
OFF	OFF	OFF	OFF
OFF	TC0_WO1	TC0_WO0	TC0_WO0
OFF	TC1_WO1	TC1_WO0	TC1_WO1
OFF	TC2_WO1	TC2_WO0	TC2_WO1
OFF	TC3_WO1	TC3_WO0	TC3_WO1

.....continued			
Group 1	Group 2	Group 3	Group 4
OFF	TC4_WO1	TC4_WO0	TC4_WO1
OFF	TC5_WO1	TC5_WO0	TC5_WO1
OFF	TC6_WO1	TC6_WO0	TC6_WO1
OFF	TC7_WO1	TC7_WO0	TC7_WO1
OFF	QSPI_CS	QSPI_CS	QSPI_CS
OFF	QSPI_DATA0	QSPI_DATA1	QSPI_DATA2
OFF	QSPI_DATA3	QSPI_DATA0	QSPI_DATA1
OFF	QSPI_DATA2	QSPI_DATA3	QSPI_DATA0
OFF	CCL_OUT1	CCL_OUT0	CCL_OUT1

Table 6-8. Remappable Output Pin Configuration – Group 1

RPn Port Pin	RPnG1R SFR	RPnG1R Bits	RPnG1R Value to Peripheral Pin Selection
RPA3	RPA3G1R	RPA3G1R[4:0]	00000 = OFF
RPA5	RPA5G1R	RPA5G1R[4:0]	00001 = SERCOM0_PAD1
RPA6	RPA6G1R	RPA6G1R[4:0]	00010 = SERCOM1_PAD1
RPA7	RPA7G1R	RPA7G1R[4:0]	00011 = REFO1
RPB5	RPB5G1R	RPB5G1R[4:0]	00100 = REFO2
RPB6	RPB6G1R	RPB6G1R[4:0]	00101 = REFO3
RPB7	RPB7G1R	RPB7G1R[4:0]	00110 = REFO4
RPB8	RPB8G1R	RPB8G1R[4:0]	00111 = QSPI_SCK
RPB9	RPB9G1R	RPB9G1R[4:0]	01000 = OFF
			01001 = OFF
			01000 = OFF
			01001 = OFF
			01010 = OFF
			01011 = OFF
			01100 = OFF
			01101 = OFF
			01110 = OFF
			01111 = OFF
			10000 = OFF
			10001 = OFF
			10010 = OFF
			10011 = OFF
			10100 = OFF
			10101 = OFF
			10110 = OFF
			10111 = OFF
			11000 = OFF
			11001 = OFF
			11010 = OFF
			11011 = OFF
			11100 = OFF
			11101 = OFF
			11110 = OFF
			11111 = OFF

.....continued			
RPn Port Pin	RPnG1R SFR	RPnG1R Bits	RPnG1R Value to Peripheral Pin Selection

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-9. Remappable Output Pin Configuration – Group 2

RPn Port Pin	RPnG2R SFR	RPnG2R Bits	RPnG2R Value to Peripheral Pin Selection
RPA0(1)	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	RPA3G2R	RPA7G2R[4.0]	00100 = OFF
RPA8	RPA8G2R	RPA8G2R[4.0]	00101 = SERCOM1_PAD3
RPB0(1)	RPB0G2R	RPB0G2R[4.0]	00110 = SERCOM1_PAD2
RPB1(1)	RPB1G2R	RPB1G2R[4.0]	00111 = TCC0_WO1
RPB4	RPB4G2R	RPB4G2R[4.0]	01000 = TCC0_WO5
RPB5	RPB5G2R	RPB5G2R[4.0]	01001 = TCC0_WO3
RPB8	RPB8G2R	RPB8G2R[4.0]	01010 = TCC1_WO1
RPB12(1)	RPB12G2R	RPB12G2R[4.0]	01011 = TCC1_WO5
RPB13(1)	RPB13G2R	RPB13G2R[4.0]	01100 = TCC1_WO3
			01101 = TCC2_WO1
			01110 = OFF
			01111 = TC0_WO1
			10000 = TC1_WO1
			10001 = TC2_WO1
			10010 = TC3_WO1
			10011 = TC4_WO1
			10100 = TC5_WO1
			10101 = TC6_WO1
			10110 = TC7_WO1
			10111 = QSPI_CS
			11000 = QSPI_DATA0
			11001 = QSPI_DATA3
			11010 = QSPI_DATA2
			11011 = CCL_OUT1
			11100 = Reserved
			11101 = Reserved
			11110 = Reserved
			11111 = Reserved

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-10. Remappable Output Pin Configuration - Group 3

RPn Port Pin	RPnG3R SFR	RPnG3R Bits	RPnG3R Value to Peripheral Pin Selection
RPA0(1)	RPA0G3R	RPA0G3R [4:0]	00000 = OFF
RPA1(1)	RPA1G3R	RPA1G3R[4.0]	00001 = OFF
RPA3	RPA3G3R	RPA3G3R[4.0]	00010 = SERCOM0_PAD0
RPA4	RPA4G3R	RPA4G3R[4.0]	00011 = SERCOM0_PAD3
RPA5	RPA5G3R	RPA5G3R[4.0]	00100 = SERCOM1_PAD2
RPA8	RPA8G3R	RPA8G3R[4.0]	00101 = OFF
RPA9	RPA9G3R	RPA9G3R[4.0]	00110 = SERCOM1_PAD3
RPA13(1)	RPA13G3R	RPA13G3R[4.0]	00111 = TCC0_WO2
RPB1(1)	RPB1G3R	RPB1G3R[4.0]	01000 = TCC0_WO0
RPB2(1)	RPB2G3R	RPB2G3R[4.0]	01001 = TCC0_WO4
RPB6	RPB6G3R	RPB6G3R[4.0]	01010 = TCC1_WO2
RPB9	RPB9G3R	RPB9G3R[4.0]	01011 = TCC1_WO0
RPB10(1)	RPB10G3R	RPB10G3R[4.0]	01100 = TCC1_WO4
RPB13(1)	RPB13G3R	RPB13G3R[4.0]	01101 = TCC2_WO0
			01110 = OFF
			01111 = TC0_WO0
			10000 = TC1_WO0
			10001 = TC2_WO0
			10010 = TC3_WO0
			10011 = TC4_WO0
			10100 = TC5_WO0
			10101 = TC6_WO0
			10110 = TC7_WO0
			10111 = QSPI_CS
			11000 = QSPI_DATA1
			11001 = QSPI_DATA0
			11010 = QSPI_DATA3
			11011 = CCL_OUT0
			11100 = Reserved
			11101 = Reserved
			11110 = Reserved
			11111 = Reserved

Note:

1. Denotes that these pins and their associated registers are not available in the 32-pin package.

Table 6-11. Remappable Output Pin Configuration – Group 4

RPn Port Pin	RPnG4R SFR	RPnG4R Bits	RPnG4R Value to Peripheral Pin Selection
RPA1(1)	RPA1G4R	RPA1G4R[4:0]	00000 = OFF
RPA2(1)	RPA2G4R	RPA2G4R[4.0]	00001 = SERCOM0_PAD2
RPA4	RPA4G4R	RPA4G4R[4.0]	00010 = OFF
RPA5	RPA5G4R	RPA5G4R[4.0]	00011 = SERCOM0_PAD0
RPA6	RPA6G4R	RPA6G4R[4.0]	00100 = SERCOM1_PAD3
RPA8	RPA8G4R	RPA8G4R[4.0]	00101 = OFF
RPA9	RPA9G4R	RPA9G4R[4.0]	00110 = SERCOM1_PAD0
RPA10	RPA10G4R	RPA10G4R[4.0]	00111 = TCC0_WO3
RPA13(1)	RPA13G4R	RPA13G4R[4.0]	01000 = TCC0_WO1
RPA14(1)	RPA14G4R	RPA14G4R[4.0]	01001 = TCC0_WO5
RPB2(1)	RPB2G4R	RPB2G4R[4.0]	01010 = TCC1_WO3
RPB3(1)	RPB3G4R	RPB3G4R[4.0]	01011 = TCC1_WO1
RPB7	RPB7G4R	RPB7G4R[4.0]	01100 = TCC1_WO5
RPB10(1)	RPB10G4R	RPB10G4R[4.0]	01101 = TCC2_WO1
RPB11(1)	RPB11G4R	RPB11G4R[4.0]	01110 = OFF
			01111 = TC0_WO0
			10000 = TC1_WO1
			10001 = TC2_WO1
			10010 = TC3_WO1
			10011 = TC4_WO1
			10100 = TC5_WO1
			10101 = TC6_WO1
			10110 = TC7_WO1
			10111 = QSPI_CS
			11000 = QSPI_DATA2
			11001 = QSPI_DATA1
			11010 = QSPI_DATA0
			11011 = CCL_OUT1
			11100 = Reserved
			11101 = Reserved
			11110 = Reserved
			11111 = Reserved

Note:
1. Denotes that these pins and their associated registers are not available in the 32-pin package.

6.6 Peripheral Multiplexing

Many pins also support one or more peripheral modules. When configured to operate with a peripheral, a pin may not be used for general input or output. In many cases, a pin must still be configured for input or output, although some peripherals override the TRISx configuration. The typical Multiplexed Port Structure Block Diagram illustrates how ports are shared with other peripherals and the associated I/O pin to which they are connected. See *Typical Multiplexed Port Structure Block Diagram* in *Block Diagram* from Related Links. Multiple peripheral functions may be multiplexed on each I/O pin. The priority of the peripheral function depends on the order of the pin descriptions. See *Function Priority for Device Pins* from Related Links.

Note: The TRISx register bit can control the output of a pin or, in some cases, by the peripheral itself.

Related Links

6.3. Block Diagram

6.7. Function Priority for Device Pins

6.6.1 Multiplexed Digital Input Peripheral

The following conditions are characteristics of a multiplexed digital input peripheral:

- Peripheral does not control the TRISx register. Some peripherals require the pin be configured as an input by setting the corresponding TRISx bit = 1.
- Peripheral input path is independent of I/O input path and uses an input buffer that is dependent on the peripheral.
- PORTx register data input path is not affected and can read the pin value.

6.6.2 Multiplexed Digital Output Peripheral

The following conditions are characteristics of a multiplexed digital output peripheral:

- The peripheral controls the output data. Some peripherals require the pin to be configured as an output by setting the corresponding TRISx bit = 0.
- If a peripheral pin has an automatic tri-state feature, the peripheral can tri-state the pin.
- The peripheral can affect the pin output driver type. For example, drive strength, slew rate and so on.
- PORTx register output data has no effect.

6.6.3 Multiplexing Digital Bidirectional Peripheral

The following conditions are characteristics of a multiplexed digital bidirectional peripheral:

- The peripheral automatically configures the pin as an output but not as an input. Some peripherals require the pin to be configured as an input by setting the corresponding TRISx bit = 1.
- Peripherals control output data.
- The pin output driver type can be affected by the peripheral (for example, drive strength, slew rate and so on).
- The PORTx register data input path is not affected and can read the pin value.
- PORTx register output data has no effect.

6.6.4 Multiplexing Analog Input Peripheral

The following condition is characteristic of a multiplexed analog input peripheral:

- All digital port input buffers are disabled.
- PORTx registers read '0' to prevent crowbar current.

6.6.5 Multiplexing Analog Output Peripheral

The following conditions are characteristics of a multiplexed analog output peripheral:

- All digital port input buffers are disabled.
- PORTx registers read '0' to prevent crowbar current.
- Analog output is driven onto the pin, independent of the associated TRISx setting.

6.7 Function Priority for Device Pins

The device pins have an associated priority order in which functionality is exhibited on each pin. This priority order impacts the availability of PPS functionality. For example, if enabling SERCOM0, choose the outputs to be High Speed mode in the DEVCFG1 fuses (bit 17), give priority to pins PB9, PA4, PA5 and PA6 and use them as SERCOM0 pins instead of GPIO/PPS pins. The following table provides

details for the priority in which functions are brought out on each device pin. Top entry is higher priority and bottom entry is lower priority for each specific pin.

Table 6-12. Priority for Device Pins PAn (n = 0-14)

Pin Name	Function In Priority Order	Reference Peripheral
pa0(1)	QSPI_DATA0	QSPI
	RTC_IN3	RTCC
	RPA0	PPS
	IOCA0	Change notification
	RA0	GPIO
pa1(1)	QSPI_SCK	QSPI
	RTC_IN2	RTCC
	RPA1	PPS
	IOCA1	Change notification
	RA1	GPIO
pa2(1)	QSPI_DATA3	QSPI
	RTC_IN1	RTCC
	RPA2	PPS
	IOCA2	Change notification
	RA2	GPIO
pa3	TRD2	Trace (Debug)
	SCLKI	Secondary oscillator
	DACOUT	DAC
	ANN0	ADC (Differential)
	RTC_IN0	RTCC
	RPA3	PPS
	IOCA3	Change notification
	RA3	GPIO
pa4	SERCOM0_PAD3	SERCOM0
	RTC_OUT	RTCC
	RPA4	PPS
	IOCA4	Change notification
	RA4	GPIO
pa5	SERCOM0_PAD0	SERCOM0
	AC_CMP0	Analog comparator
	RPA5	PPS
	IOCA5	Change notification
	RA5	GPIO

.....continued		
Pin Name	**Function In Priority Order**	**Reference Peripheral**
pa6	TRD3	Trace (Debug)
	SERCOM0_PAD1	SERCOM0
	AC_CMP1_ALT	Analog comparator
	RPA6	PPS
	IOCA6	Change notification
	RA6	GPIO
pa7	TRACECLK	Trace (Debug)
	SERCOM1_PAD0	SERCOM1
	RPA7	PPS
	IOCA7	Change notification
	RA7	GPIO
pa8	SERCOM1_PAD1	SERCOM1
	RPA8	PPS
	IOCA8	Change notification
	RA8	GPIO
pa9	SERCOM1_PAD2	SERCOM1
	RTC_IN0_ALT	RTCC
	RPA9	PPS
	IOCA9	Change notification
	RA9	GPIO
pa10	SERCOM1_PAD3	SERCOM1
	RTC_OUT_ALT	RTCC
	RPA10	PPS
	IOCA10	Change notification
	RA10	GPIO
pa11	SOSCI	Secondary oscillator
	RPA11	PPS (Re-mappable input only)
	RA11	GPIO (input only)
pa12	SOSCO	Secondary oscillator
	RPA12	PPS (Re-mappable input only)
	RA12	GPIO (input only)
pa13(1)	SERCOM2_PAD0	SERCOM2 (I2C only)
	AC_CMP1	Analog comparator
	RPA13	PPS
	IOCA13	Change notification
	RA13	GPIO

.....continued		
Pin Name	**Function In Priority Order**	**Reference Peripheral**
pa14(1)	SERCOM2_PAD1	SERCOM2 (I2C)
	RPA14	PPS
	IOCA14	Change notification
	RA14	GPIO

Note:
1. Indicates pins are not available on the 32-pin package and available only on the 48-pin package.

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

Pin Name	Function In Priority Order	Reference Peripheral
pb0(1)	AN4	ADC
	CVD4	CVD
	CVDR4	CVD
	CVDT4	CVD
	AC_AIN2	Analog comparator
	RPB0	PPS
	IOCB0	Change notification
	RB0	GPIO
pb1(1)	AN5	ADC
	CVD5	CVD
	CVDR5	CVD
	CVDT5	CVD
	AC_AIN3	Analog comparator
	RPB1	PPS
	IOCB1	Change notification
	RB1	GPIO
pb2(1)	AN6	ADC
	CVD6	CVD
	CVDR6	CVD
	CVDT6	CVD
	AC_AIN0	Analog comparator
	RPB2	PPS
	IOCB2	Change notification
	RB2	GPIO

.....continued		
Pin Name	**Function In Priority Order**	**Reference Peripheral**
pb3[1]	AN7	ADC
	CVD7	CVD
	CVDR7	CVD
	CVDT7	CVD
	AC_AIN1	Analog comparator
	RPB3	PPS
	IOCB3	Change notification
	RB3	GPIO
pb4	AN0	ADC
	CVD0	CVD
	CVDR0	CVD
	CVDT0	CVD
	RPB4	PPS
	IOCB4	Change notification
	RB4	GPIO
pb5	TRD0	Trace (Debug)
	AN1	ADC
	CVD1	CVD
	CVDR1	CVD
	CVDT1	CVD
	RPB5	PPS
	IOCB5	Change notification
	RB5	GPIO
pb6	TRD1	Trace (Debug)
	AN2	ADC
	CVD2	CVD
	CVDR2	CVD
	CVDT2	CVD
	RPB6	PPS
	IOCB6	Change notification
	RB6	GPIO

.....continued		
Pin Name	**Function In Priority Order**	**Reference Peripheral**
pb7	SWO	Debug
	AN3	ADC
	CVD3	CVD
	CVDR3	CVD
	CVDT3	CVD
	LVDIN	LVD Voltage reference
	RPB7	PPS
	IOCB7	Change notification
	RB7	GPIO
pb8	SWCLK	Debug
	RPB8	PPS
	IOCB8	Change notification
	RB8	GPIO
pb9	CM4_SWDIO	Debug
	SERCOM0_PAD2	SERCOM0
	RPB9	PPS
	INT0	Wake-up interrupt
	IOCB9	Change notification
	RB9	GPIO
pb10(1)	RPB10	PPS
	IOCB10	Change notification
	RB10	GPIO
pb11(1)	QSPI_DATA2	QSPI
	RPB11	PPS
	IOCB11	Change notification
	RB11	GPIO
pb12(1)	QSPI_DATA1	QSPI
	RPB12	PPS
	IOCB12	Change notification
	RB12	GPIO
pb13(1)	QSPI_CS	QSPI
	RTC_EVENT	RTCC
	RPB13	PPS
	IOCB13	Change notification
	RB13	GPIO

Note:

1. Indicates that pins are not available on the 32-pin package.

6.8 Operation in Power Saving Modes

6.8.1 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.

6.8.2 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.

6.9 Results of Various Resets

Table 6-14. 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.

6.10 Port Register Summary

See *PORT A* module in the *Product Memory Mapping Overview* from Related Links for base address.

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.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ANSELA	7:0					ANSA3			
		15:8								
		23:16								
		31:24								
0x04 ... 0x0F	Reserved									
0x10	TRISA	7:0	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx
		15:8		TRISAx	TRISAx			TRISAx	TRISAx	TRISAx
		23:16								
		31:24								
0x14 ... 0x1F	Reserved									
0x20	PORTA	7:0	RAx	RAx	RAx	RAx	RAx	RAx	RAx	RAx
		15:8		RAx	RAx	RAx	RAx	RAx	RAx	RAx
		23:16								
		31:24								
0x24 ... 0x2F	Reserved									
0x30	LATA	7:0	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx
		15:8		LATAx	LATAx			LATAx	LATAx	LATAx
		23:16								
		31:24								
0x34 ... 0x3F	Reserved									
0x40	ODCA	7:0	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx
		15:8		ODCAx	ODCAx			ODCAx	ODCAx	ODCAx
		23:16								
		31:24								
0x44 ... 0x4F	Reserved									
0x50	CNPUA	7:0	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx
		15:8		CNPUAx	CNPUAx			CNPUAx	CNPUAx	CNPUAx
		23:16								
		31:24								
0x54 ... 0x5F	Reserved									
0x60	CNPDA	7:0	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx
		15:8		CNPDAx	CNPDAx			CNPDAx	CNPDAx	CNPDAx
		23:16								
		31:24								
0x64 ... 0x6F	Reserved									
0x70	CNCONA	7:0								
		15:8	ON	FRZ	SIDL		EDGEDETECT			
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74 ... 0x7F	Reserved									
0x80	CNENA	7:0	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx
		15:8		CNENAx	CNENAx			CNENAx	CNENAx	CNENAx
		23:16								
		31:24								
0x84 ... 0x8F	Reserved									
0x90	CNSTATA	7:0	CNSTATAx	CNSTATAx	CNSTATAx	CNSTATAx	CNSTATAx	CNSTATAx	CNSTATAx	CNSTATAx
		15:8		CNSTATAx	CNSTATAx			CNSTATAx	CNSTATAx	CNSTATAx
		23:16								
		31:24								
0x94 ... 0x9F	Reserved									
0xA0	CNNEA	7:0	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx
		15:8		CNNEAx	CNNEAx			CNNEAx	CNNEAx	CNNEAx
		23:16								
		31:24								
0xA4 ... 0xAF	Reserved									
0xB0	CNFA	7:0	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx
		15:8		CNFAx	CNFAx			CNFAx	CNFAx	CNFAx
		23:16								
		31:24								
0xB4 ... 0xBF	Reserved									
0xC0	SRCON0A	7:0	SR0x	SR0x	SR0x	SR0x		SR0x	SR0x	SR0x
		15:8		SR0x	SR0x			SR0x	SR0x	SR0x
		23:16								
		31:24								
0xC4 ... 0xCF	Reserved									
0xD0	SRCON1A	7:0	SR1x	SR1x	SR1x	SR1x		SR1x	SR1x	SR1x
		15:8		SR1x	SR1x			SR1x	SR1x	SR1x
		23:16								
		31:24								
0xD4 ... 0xFF	Reserved									
0x0100	ANSELB	7:0	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx
		15:8								
		23:16								
		31:24								
0x0104 ... 0x010F	Reserved									
0x0110	TRISB	7:0	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx
		15:8			TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx
		23:16								
		31:24								
0x0114 ... 0x011F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0120	PORTB	7:0	RBx	RBx	RBx	RBx	RBx	RBx	RBx	RBx
		15:8			RBx	RBx	RBx	RBx	RBx	RBx
		23:16								
		31:24								
0x0124 ... 0x012F	Reserved									
0x0130	LATB	7:0	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx
		15:8			LATBx	LATBx	LATBx	LATBx	LATBx	LATBx
		23:16								
		31:24								
0x0134 ... 0x013F	Reserved									
0x0140	ODCB	7:0	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx
		15:8			ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx
		23:16								
		31:24								
0x0144 ... 0x014F	Reserved									
0x0150	CNPUB	7:0	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx
		15:8			CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx
		23:16								
		31:24								
0x0154 ... 0x015F	Reserved									
0x0160	CNPDB	7:0	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx
		15:8			CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx
		23:16								
		31:24								
0x0164 ... 0x016F	Reserved									
0x0170	CNCONB	7:0								
		15:8	ON	FRZ	SIDL		EDGEDETECT			
		23:16								
		31:24								
0x0174 ... 0x017F	Reserved									
0x0180	CNENB	7:0	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx
		15:8			CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx
		23:16								
		31:24								
0x0184 ... 0x018F	Reserved									
0x0190	CNSTATB	7:0	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx
		15:8			CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx
		23:16								
		31:24								
0x0194 ... 0x019F	Reserved									
0x01A0	CNNEB	7:0	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx
		15:8			CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01A4 ... 0x01AF	Reserved									
0x01B0	CNFB	7:0	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx
		15:8			CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx
		23:16								
		31:24								
0x01B4 ... 0x01BF	Reserved									
0x01C0	SRCON0B	7:0								
		15:8			SR0x	SR0x	SR0x	SR0x		
		23:16								
		31:24								
0x01C4 ... 0x01CF	Reserved									
0x01D0	SRCON1B	7:0								
		15:8			SR1x	SR1x	SR1x	SR1x		
		23:16								
		31:24								

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

6.11 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 require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual 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

6.11.1 Analog Select Register for PortA

Name: ANSELA
Offset: 0x00
Reset: 0x8
Property: -

The ANSELA register controls the operation of the analog portA pins.

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
					ANSA3			
Access					R/W			
Reset					1			

Bit 3 – ANSA3 Analog Select for PA3
Configures the PA3 as an analog input when this bit is set to '1'.

6.11.2 Tri-state Functions for PortA

Name: TRISA
Offset: 0x10
Reset: 0x0
Property: -

The TRISA register configures the data direction flow through port I/O pins.

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
		TRISAx	TRISAx			TRISAx	TRISAx	TRISAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx	TRISAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – TRISAx (x = 0 to 14; x = 0 for bit0 mapped to PA0, ... x = 14 for bit14 mapped to PA14) Tri-state pins for PortA

The tri-state data direction bit configures the selected I/O pin of Port A as an input or output.

Value	Description
1	Configures the I/O as input
0	Configures the I/O as output

6.11.3 Port Pin Data for PortA

Name: PORTA
Offset: 0x20
Reset: 0x0
Property: -

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

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
		RAx	RAx	RAx	RAx	RAx	RAx	RAx
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	6	5	4	3	2	1	0
	RAx	RAx	RAx	RAx	RAx	RAx	RAx	RAx
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 – RAx (x = 0 to 14; x = 0 for bit0 mapped to PA0, ... x = 14 for bit14 mapped to PA14) Port pin configuration for PortA

6.11.4 Latch Functions for PortA

Name: LATA
Offset: 0x30
Reset: 0x0
Property: -

The LATA register (PORTA data latch) holds data written to port I/O pins. A write to a LATA register latches data to corresponding port I/O pins. Those I/O port pins configured as outputs are updated. A read from a LATA register reads the data held in the PORTA data latch, not from the port I/O pins.

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
		LATAx	LATAx			LATAx	LATAx	LATAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx	LATAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – LATAx (x = 0 to 14; x = 0 for bit0 mapped to PA0, ... x = 14 for bit14 mapped to PA14) Latch configuration for PortA

6.11.5 Open-Drain Configuration for PortA

Name: ODCA
Offset: 0x40
Reset: 0x0
Property: -

This register configures each I/O pin individually for either normal digital output or open-drain output, associated with each I/O pin.

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
		ODCAx	ODCAx			ODCAx	ODCAx	ODCAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx	ODCAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – ODCAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Open-Drain Configuration for PortA

Note: After a Reset, the status of all the bits of the ODCA register is set to '0'.

Value	Description
1	Configures an I/O pin as an open-drain output.
0	Configures an I/O pin as an normal digital output.

6.11.6 Change Notice Pull-up for PortA

Name: CNPUA
Offset: 0x50
Reset: 0x0
Property: -

This register enables the weak internal pull-ups connected with an I/O pin when any of the control bits is set.
Note: If configuring the port pin as a digital output, the user must always disable this register.

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
		CNPUAx	CNPUAx			CNPUAx	CNPUAx	CNPUAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx	CNPUAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNPUAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Pull-up configuration for PortA

Value	Description
1	Enables the weak pull-up associated with an I/O pin.
0	Disables the weak pull-up associated with an I/O pin.

6.11.7 Change Notice Pull-down for PortA

Name: CNPDA
Offset: 0x60
Reset: 0x0
Property: -

This register enables the weak pull-down connected with an I/O pin when any of the control bits is set.
Note: If configuring the port pin as a digital output, the user must always disable this register.

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
		CNPDAx	CNPDAx			CNPDAx	CNPDAx	CNPDAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx	CNPDAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNPDAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Pull-down configuration for PortA

Value	Description
1	Enables the weak pull-down associated with an I/O pin.
0	Disables the weak pull-down associated with an I/O pin.

6.11.8 Change Notice Control for PortA

Name: CNCONA
Offset: 0x70
Reset: 0x0
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
	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 in the Debug mode
0 = Continues the module operation when 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. This is associated with CNENA (positive edge)/CNNEA (negative edge)/CNFA.
0 = Mismatch Style. Detects change from last PortA read. This is associated with CNENA/CNSTATA.

6.11.9 Change Notice Enable for PortA

Name: CNENA
Offset: 0x80
Reset: 0x0
Property: -

This register contains 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. When EDGEDETECT is set, CNENA controls the positive edge. CNENA enables a mismatch CN interrupt condition when EDGEDETECT is not set.

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
		CNENAx	CNENAx			CNENAx	CNENAx	CNENAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx	CNENAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNENAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Enable for PortA

Value	Description
1	Enables a mismatch/positive edge CN interrupt condition associated with an I/O pin.
0	Disables a mismatch/positive edge CN interrupt condition associated with an I/O pin.

6.11.10 Change Notice Status for PortA

Name: CNSTATA
Offset: 0x90
Reset: 0x0
Property: -

This register indicates whether a change occurred on the corresponding pin since the last read of the PortA bit.

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
		CNSTATAx	CNSTATAx			CNSTATAx	CNSTATAx	CNSTATAx
Access		R	R			R	R	R
Reset		0	0			0	0	0

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

Bits 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNSTATAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Status for PortA
'1' indicates change occurred in an I/O pin.

6.11.11 Change Notice Enable for PortA

Name: CNNEA
Offset: 0xA0
Reset: 0x0
Property: -

This register contains 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. When EDGEDETECT is set, CNNEA controls the negative edge.

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
		CNNEAx	CNNEAx			CNNEAx	CNNEAx	CNNEAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx	CNNEAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNNEAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Enable for PortA

Value	Description
1	Enables a negative edge CN interrupt condition associated with an I/O pin.
0	Disables a negative edge CN interrupt condition associated with an I/O pin.

6.11.12 Change Notice Flag for PortA

Name: CNFA
Offset: 0xB0
Reset: 0x0
Property: -

This register indicates edge-detect style change occurred on the corresponding pin since the last read of the PortA bit.

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
		CNFAx	CNFAx			CNFAx	CNFAx	CNFAx
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx	CNFAx
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 0,1,2,3,4,5,6,7,8,9,10,13,14 – CNFAx (x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14; x = 0 for bit0 mapped to PA0, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Change Notice Flag for PortA
When CNCONAx = 1, CNFAx stores the occurrence of the CN event until cleared by the software.
When CNCONAx = 0, CNFAx Reads '0'.

Value	Description
1	An Enabled Edge Event occurred on pin PORTAx
0	An Enabled Edge Event did not occur on pin PORTAx

6.11.13 Slew Rate Control 0 for PortA

Name: SRCON0A
Offset: 0xC0
Reset: 0x0
Property: -

This register configures the slew rate control bits associated with Port A.
Note: To configure the slew rate, the user must also configure the SRCON1A register associated with Port A. See *Slew Rate Control Bit Settings* table in the *Slew Rate Control* 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
		SR0x	SR0x			SR0x	SR0x	SR0x
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	SR0x	SR0x	SR0x	SR0x		SR0x	SR0x	SR0x
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 0,1,2,4,5,6,7,8,9,10,13,14 – SR0x (x = 0, 1, 2, 4, 5, 6, 7, 8, 9, 10; x = 0 for bit0 mapped to PA0, ... x = 2 for bit 2, x = 4 for bit4, ... x = 10 for bit 10, x = 13 for bit13, ... x = 14 for bit 14 mapped to PA14) Slew Rate Control 0 for PortA.

Related Links

6.4.1.8. Slew Rate Control

6.11.14 Slew Rate Control 1 for PortA

Name: SRCON1A
Offset: 0xD0
Reset: 0x0
Property: -

This register configures the slew rate control bits associated with Port A.
Note: To configure the slew rate, the user must also configure the SRCON0A register associated with Port A. See *Slew Rate Control Bit Settings* table in the *Slew Rate Control* 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
		SR1x	SR1x			SR1x	SR1x	SR1x
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	SR1x	SR1x	SR1x	SR1x		SR1x	SR1x	SR1x
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 0,1,2,4,5,6,7,8,9,10,13,14 – SR1x (x = 0, 1, 2, 4, 5, 6, 7, 8, 9, 10; x = 0 for bit0, ... x = 2 for bit 2, x = 4 for bit4, ... x = 10 for bit10, x = 13 for bit13, ... x = 14 for bit14) Slew Rate Control 1 for PortA

Related Links
6.4.1.8. Slew Rate Control

6.11.15 Analog Select Register for PortB

Name: ANSELB
Offset: 0x100
Reset: 0x8
Property: -

The ANSELB register controls the operation of the analog PortB pins.

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
	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx	ANSBx
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	1

Bits 0,1,2,3,4,5,6,7 – ANSBx (x = 0 to 7; x = 0 for bit0, ... x = 7 for bit7) Analog Select for PB
Configures the PB as an analog input when this bit is set to '1'.

6.11.16 Tri-state Functions for PortB

Name: TRISB
Offset: 0x110
Reset: 0x0
Property: -

The TRISB register configures the data direction flow through port I/O pins.

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
			TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx
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
	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx	TRISBx
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 – TRISBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Tri-state pins for PortB

The tri-state data direction bit configures the selected I/O pin of Port B as an input or output.

Value	Description
1	Configures the I/O as input
0	Configures the I/O as output

6.11.17 Port Pin Data for PortB

Name: PORTB
Offset: 0x120
Reset: 0x0
Property: -

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

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
			RBx	RBx	RBx	RBx	RBx	RBx
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
	RBx	RBx	RBx	RBx	RBx	RBx	RBx	RBx
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 – RBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) PortB configuration

6.11.18 Latch Functions for PortB

Name: LATB
Offset: 0x130
Reset: 0x0
Property: -

The LATB register (PORTB data latch) holds data written to port I/O pins. A write to a LATB register latches data to corresponding port I/O pins. Those I/O port pins configured as outputs are updated. A read from a LATB register reads the data held in the PORTB data latch, not from the port I/O pins.

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
			LATBx	LATBx	LATBx	LATBx	LATBx	LATBx
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
	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx	LATBx
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 – LATBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Latch configuration for PortB

6.11.19 Open-Drain Configuration for PortB

Name: ODCB
Offset: 0x140
Reset: 0x0
Property: -

This register configures each I/O pin individually for either normal digital output or open-drain output, associated with each I/O pin.

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
			ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx
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
	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx	ODCBx
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 – ODCBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Open-Drain Configuration for PortB

Note: After a Reset, the status of all the bits of the ODCB register is set to '0'.

Value	Description
1	Configures an I/O pin as an open-drain output.
0	Configures an I/O pin as an normal digital output.

6.11.20 Change Notice Pull-up for PortB

Name: CNPUB
Offset: 0x150
Reset: 0x0
Property: -

This register enables the weak internal pull-ups connected with an I/O pin when any of the control bits is set.
Note: This register must always be disabled when the port pin is configured as a digital output.

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
			CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx
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
	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx	CNPUBx
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 – CNPUBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Pull-up configuration for PortB

Value	Description
1	Enables the weak pull-up associated with an I/O pin.
0	Disables the weak pull-up associated with an I/O pin.

6.11.21 Change Notice Pull-down for PortB

Name: CNPDB
Offset: 0x160
Reset: 0x0
Property: -

This register enables the weak pull-down connected with an I/O pin when any of the control bits is set.
Note: This register must always be disabled when the port pin is configured as a digital output.

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
			CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx
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
	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx	CNPDBx
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 – CNPDBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Pull-down configuration for PortB

Value	Description
1	Enables the weak pull-down associated with an I/O pin.
0	Disables the weak pull-down associated with an I/O pin.

6.11.22 Change Notice Control for PortB

Name: CNCONB
Offset: 0x170
Reset: 0x0
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
	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 in the Idle mode
0 = Continues the module operation when in the Idle mode

Bit 11 – EDGEDETECT Change Notification Style bit
1 = Edge Style. Detects edge transitions. This is associated with CNENB (positive edge)/CNNEB (negative edge)/CNFA.
0 = Mismatch Style. Detects change from last PortA read. This is associated with CNENB/CNSTATB.

6.11.23 Change Notice Enable for PortB

Name: CNENB
Offset: 0x180
Reset: 0x0
Property: -

This register contains 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. When EDGEDETECT is set, CNENB controls the positive edge. CNENB enables a mismatch CN interrupt condition when EDGEDETECT is not set

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
			CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx
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
	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx	CNENBx
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 – CNENBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Enable for PortB

Value	Description
1	Enables a mismatch/positive edge CN interrupt condition associated with an I/O pin.
0	Disables a mismatch/positive edge CN interrupt condition associated with an I/O pin.

6.11.24 Change Notice Status for PortB

Name: CNSTATB
Offset: 0x190
Reset: 0x0
Property: -

This register indicates whether a change occurred on the corresponding pin since the last read of the PortB bit.

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
			CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx	CNSTATBx
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

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

Bits 0,1,2,3,4,5,6,7,8,9,10,11,12,13 – CNSTATBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Status for PortB
'1' indicates change occurred in an I/O pin.

6.11.25 Change Notice Enable for PortB

Name: CNNEB
Offset: 0x1A0
Reset: 0x0
Property: -

This register contains 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. When EDGEDETECT is set, CNNEB controls the negative edge.

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
			CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx
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
	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx	CNNEBx
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 – CNNEBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Enable for PortB

Value	Description
1	Enables a mismatch/negative edge CN interrupt condition associated with an I/O pin.
0	Disables a mismatch/negative edge CN interrupt condition associated with an I/O pin.

6.11.26 Change Notice Flag for PortB

Name: CNFB
Offset: 0x1B0
Reset: 0x0
Property: -

This register indicates the edge-detect style change occurred on the corresponding pin since the last read of the PortB bit.

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
			CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx
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
	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx	CNFBx
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 – CNFBx (x = 0 to 13; x = 0 for bit0 mapped to PB0, ... x = 13 for bit13 mapped to PB13) Change Notice Flag for PortB

When CNCONBx = 1, CNFBx stores the occurrence of the CN event until cleared by the software.
When CNCONBx = 0, CNFBx Reads '0'.

Value	Description
1	An Enabled Edge Event occurred on pin PORTBx
0	An Enabled Edge Event did not occur on pin PORTBx

6.11.27 Slew Rate Control 0 for PortB

Name: SRCON0B
Offset: 0x1C0
Reset: 0x0
Property: -

This register configures the slew rate control bits associated with PortB.
Note: To configure the slew rate, the user must also configure the SRCON1B register associated with PortB. See *Slew Rate Control Bit Settings* table in the *Slew Rate Control* 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
			SR0x	SR0x	SR0x	SR0x		
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								

Bits 10,11,12,13 – SR0x (x = 10, 11, 12, 13; x = 10 for bit10 mapped to PB10, ... x = 13 for bit13 mapped to PB13) Slew Rate Control 0 for PortB

Related Links

6.4.1.8. Slew Rate Control

6.11.28 Slew Rate Control 1 for PortB

Name: SRCON1B
Offset: 0x1D0
Reset: 0x0
Property: -

This register configures the slew rate control bits associated with Port B.
Note: To configure the slew rate, the user must also configure the SRCON0A register associated with Port B. See *Slew Rate Control Bit Settings* table in the *Slew Rate Control* 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
			SR1x	SR1x	SR1x	SR1x		
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								

Bits 10,11,12,13 – SR1x (x = 10, 11, 12, 13; x = 10 for bit10 mapped to PB10, ... x = 13 for bit13 mapped to PB13) Slew Rate Control 1 for PortB

Related Links

6.4.1.8. Slew Rate Control

6.12 Peripheral Pin Select (PPS) Input Mapping Register Summary

See the *PPS* module in the *Product Memory Mapping Overview* from Related Links for base address.

Table 6-15. Peripheral Pin Select Input Registers

Offset	Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x000	EXTINT0R	7:0	—	—	—	—	EXTINT0R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—				
0x004	EXTINT1R	7:0	—	—	—	—	EXTINT1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x008	EXTINT2R	7:0	—	—	—	—	EXTINT2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x00C	EXTINT3R	7:0	—	—	—	—	EXTINT3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—				
0x03C	NMIR	7:0	—	—	—	—	NMIR[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x040	SCOM0P0R	7:0	—	—	—	—	SCOM0P0R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x044	SCOM0P1R	7:0	—	—	—	—	SCOM0P1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x048	SCOM0P2R	7:0	—	—	—	—	SCOM0P2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x04C	SCOM0P3R	7:0	—	—	—	—	SCOM0P3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x050	SCOM1P0R	7:0	—	—	—	—	SCOM1P0R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x054	SCOM1P1R	7:0	—	—	—	—	SCOM1P1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x058	SCOM1P2R	7:0	—	—	—	—	SCOM1P2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x05C	SCOM1P3R	7:0	—	—	—	—	SCOM1P3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x084	QD0R	7:0	—	—	—	—	QD0R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x088	QD1R	7:0	—	—	—	—	QD1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x08C	QD2R	7:0	—	—	—	—	QD2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x090	QD3R	7:0	—	—	—	—	QD3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x094	REFIR	7:0	—	—	—	—	REFIR[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x098	CCLIN0R	7:0	—	—	—	—	CCLIN0R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x09C	CCLIN1R	7:0	—	—	—	—	CCLIN1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0A0	CCLIN2R	7:0	—	—	—	—	CCLIN2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x0A4	CCLIN3R	7:0	—	—	—	—	CCLIN3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0A8	CCLIN4R	7:0	—	—	—	—	CCLIN4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0AC	CCLIN5R	7:0	—	—	—	—	CCLIN5R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0B0	TC0WO0G1R	7:0	—	—	—	—	TC0WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0B4	TC0WO0G2R	7:0	—	—	—	—	TC0WO0G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0B8	TC0WO1G3R	7:0	—	—	—	—	TC0WO1G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0BC	TC0WO1G4R	7:0	—	—	—	—	TC0WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0C0	TC1WO0G1R	7:0	—	—	—	—	TC1WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0C4	TC1WO1G2R	7:0	—	—	—	—	TC1WO0G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0C8	TC2WO0G1R	7:0	—	—	—	—	TC2WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0CC	TC2WO0G3R	7:0	—	—	—	—	TC2WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x0D0	TC2WO1G2R	7:0	—	—	—	—	TC2WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0D4	TC2WO1G4R	7:0	—	—	—	—	TC2WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0D8	TC3WO0G1R	7:0	—	—	—	—	TC3WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0DC	TC3WO0G3R	7:0	—	—	—	—	TC3WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0E0	TC3WO1G2R	7:0	—	—	—	—	TC3WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0E4	TC3WO1G4R	7:0	—	—	—	—	TC3WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0E8	TC4WO0G1R	7:0	—	—	—	—	TC4WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0EC	TC4WO0G3R	7:0	—	—	—	—	TC4WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0F0	TC4WO1G2R	7:0	—	—	—	—	TC4WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0F4	TC4WO1G4R	7:0	—	—	—	—	TC4WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x0F8	TC5WO0G1R	7:0	—	—	—	—	TC5WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x0FC	TC5WO0G3R	7:0	—	—	—	—	TC5WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x100	TC5WO1G2R	7:0	—	—	—	—	TC5WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x104	TC5WO1G4R	7:0	—	—	—	—	TC5WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x108	TC6WO0G1R	7:0	—	—	—	—	TC6WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x10C	TC6WO0G3R	7:0	—	—	—	—	TC6WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x110	TC6WO1G2R	7:0	—	—	—	—	TC6WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x114	TC6WO1G4R	7:0	—	—	—	—	TC6WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x118	TC7WO0G1R	7:0	—	—	—	—	TC7WO0G1R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x11C	TC7WO0G3R	7:0	—	—	—	—	TC7WO0G3R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x120	TC7WO1G2R	7:0	—	—	—	—	TC7WO1G2R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x124	TC7WO1G4R	7:0	—	—	—	—	TC7WO1G4R[3:0]			
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

Related Links

8. Product Memory Mapping Overview

6.13 Peripheral Pin Select (PPS) Output Mapping Register Summary

See the *PPS* module in the *Product Memory Mapping Overview* from Related Links for base address.

Table 6-16. Peripheral Pin Select Output Registers

Offset	Register Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x200	RPA0G2R(1)	7:0	—	—	—	RPA0G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x204	RPA0G3R(1)	7:0	—	—	—	RPA0G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x208	RPA1G3R(1)	7:0	—	—	—	RPA1G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x20C	RPA1G4R(1)	7:0	—	—	—	RPA1G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x214	RPA2G4R(1)	7:0	—	—	—	RPA2G4R [4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x218	RPA3G1R	7:0	—	—	—	RPA3G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x21C	RPA3G2R	7:0	—	—	—	RPA3G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x220	RPA3G3R	7:0	—	—	—	RPA3G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x224	RPA4G2R	7:0	—	—	—	RPA4G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x228	RPA4G3R	7:0	—	—	—	RPA4G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Register Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x22C	RPA4G4R	7:0	—	—	—	RPA4G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x230	RPA5G1R	7:0	—	—	—	RPA5G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x234	RPA5G3R	7:0	—	—	—	RPA5G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x238	RPA5G4R	7:0	—	—	—	RPA5G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x23C	RPA6G1R	7:0	—	—	—	RPA6G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x240	RPA6G2R	7:0	—	—	—	RPA6G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x244	RPA6G4R	7:0	—	—	—	RPA6G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x248	RPA7G1R	7:0	—	—	—	RPA7G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x24C	RPA7G2R	7:0	—	—	—	RPA7G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x250	RPA8G2R	7:0	—	—	—	RPA8G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x254	RPA8G3R	7:0	—	—	—	RPA8G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Register Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x258	RPA8G4R	7:0	—	—	—	RPA8G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x260	RPA9G3R	7:0	—	—	—	RPA9G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x264	RPA9G4R	7:0	—	—	—	RPA9G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x26C	RPA10G4R	7:0	—	—	—	RPA10G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x278	RPA13G3R(1)	7:0	—	—	—	RPA13G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x27C	RPA13G4R(1)	7:0	—	—	—	RPA13G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x284	RPA14G4R(1)	7:0	—	—	—	RPA14G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x290	RPB0G2R(1)	7:0	—	—	—	RPB0G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x294	RPB1G2R(1)	7:0	—	—	—	RPB1G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x298	RPB1G3R(1)	7:0	—	—	—	RPB1G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x29C	RPB2G3R(1)	7:0	—	—	—	RPB2G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Register Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x2A0	RPB2G4R(1)	7:0	—	—	—	RPB2G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2A8	RPB3G4R(1)	7:0	—	—	—	RPB3G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2B0	RPB4G2R	7:0	—	—	—	RPB4G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2B4	RPB5G2R	7:0	—	—	—	RPB5G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2B8	RPB5G1R	7:0	—	—	—	RPB5G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2BC	RPB6G3R	7:0	—	—	—	RPB6G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2C0	RPB6G1R	7:0	—	—	—	RPB6G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2C4	RPB7G1R	7:0	—	—	—	RPB7G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2C8	RPB7G4R	7:0	—	—	—	RPB7G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2CC	RPB8G1R	7:0	—	—	—	RPB8G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2D0	RPB8G2R	7:0	—	—	—	RPB8G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

.....continued

Offset	Register Name	Bit Position	Bits							
			7	6	5	4	3	2	1	0
0x2D4	RPB9G1R	7:0	—	—	—	RPB9G1R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2D8	RPB9G3R	7:0	—	—	—	RPB9G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2DC	RPB10G3R(1)	7:0	—	—	—	RPB10G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2E0	RPB10G4R(1)	7:0	—	—	—	RPB10G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2E8	RPB11G4R(1)	7:0	—	—	—	RPB11G4R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2F0	RPB12G2R(1)	7:0	—	—	—	RPB12G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2F4	RPB13G2R(1)	7:0	—	—	—	RPB13G2R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—
0x2F8	RPB13G3R(1)	7:0	—	—	—	RPB13G3R[4:0]				
		15:8	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		31:24	—	—	—	—	—	—	—	—

Note:

1. Indicates pins are not available on the 32-pin package, and available only on the 48-pin package.

Related Links

8. Product Memory Mapping Overview

6.14 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 require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual 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

6.14.1 Peripheral Pin Select Input Register

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

Notes:

1. For offset address, see *Peripheral Pin Select Input Registers* table in the *Peripheral Pin Select (PPS) Input Mapping Register Summary* from Related Links.
2. The user can only change the register values 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
					[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*, *Input Pin Selection Group 4* and *Input Pin Selection Group 5* tables in the *Input Mapping in PIC32CX-BZ3 Family of Devices* for input pin selection values from Related Links.
Note: This field is only writable when CFGCON0.IOLOCK = 0.

Related Links
6.12. Peripheral Pin Select (PPS) Input Mapping Register Summary
6.5.5.1.4. Input Mapping in PIC32CX-BZ3 Family of Devices

6.14.2 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 *Peripheral Pin Select (PPS) Output Mapping Register Summary* from Related Links.
2. The user can only change the register values 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 – Group 1*, *Remappable Output Pin Configuration – Group 2*, *Remappable Output Pin Configuration – Group 3*, and *Remappable Output Pin Configuration – Group 4* tables in the *Output Mapping in PIC32CX-BZ3 Family of Devices* from Related Links.
Note: This field is only writable, when CFGCON0.IOLOCK = 0.

Related Links

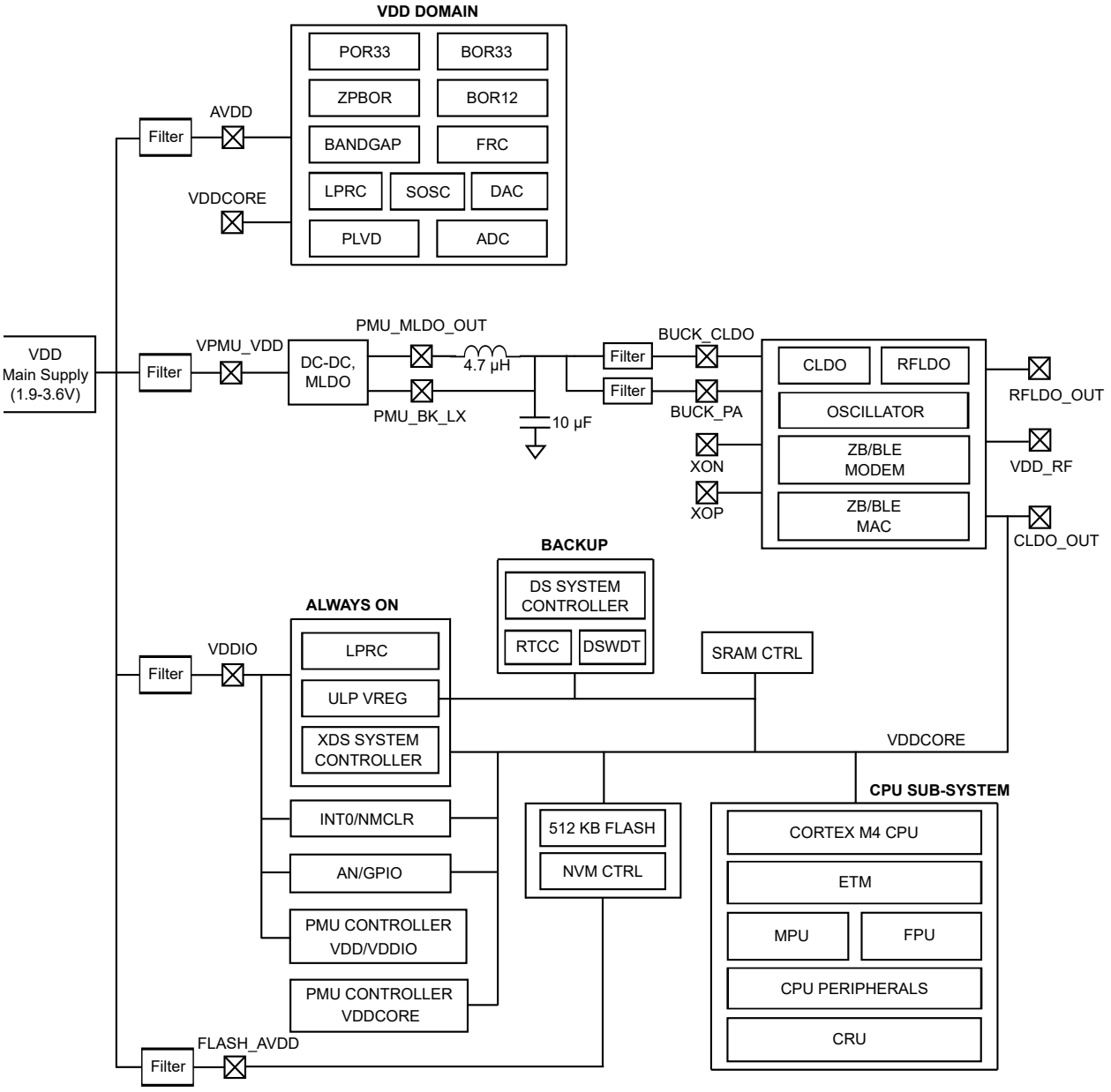
6.13. Peripheral Pin Select (PPS) Output Mapping Register Summary
6.5.5.2.2. Output Mapping in PIC32CX-BZ3 Family of Devices

7. Power Subsystem

7.1 Block Diagram

The following figure shows the detailed view of the power subsystem on the PIC32CX-BZ3 device.

Figure 7-1. Power Subsystem Block Diagram



The power domains of PIC32CX-BZ3 SoC are as follows:

- VDD – 1.9-3.6V, Main Supply powering VDDIO, FLASH_VDD, AVDD, PMU_VDDIO, PMU_VDDP. All other supplies are derived from VDD with or without filtering.
  - VDDIO
    - 1.9-3.6V, powering the Always ON (AON), PMU Controller, AN/GPIO, INT0/NMCLR, BKUP

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

Other power supply pins as follows:

- CLDO_OUT (1.2V ± 5%)
  - Output pin for the internal voltage regulator for decoupling, do not use this pin as an external power supply source
  - CLDO is powered with 1.35V ± 5% from the combination of DC-DC, MLDO and external board filtering
  - VDDCORE is derived from CLDO_OUT
  - Powers the core, memories and peripherals
- PMU_BK_LX
  - Pin for connecting the inductor for the internal switching regulator
- PMU_MLDO_OUT (1.35V ± 3.7%)
  - 1.35V PMU output pin. This is the shared output pin for both MLDO and the DC-DC converter
  - MLDO_OUT powers the internal LDO's in the Bluetooth/Zigbee subsystem

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

7.2 VDD Voltage Domain Overview

The PIC32CX-BZ3 Power System VDD Integration Block (SIB) consists of the following modules:

- Power-on Reset (POR) – Use this module to hold all the components in their inactive state until VDD reaches a stable operating voltage. Use this module to ensure that the supply voltage is sufficient for proper operation of all other analog modules (PD_AVDD) in the Power SIB module.
- Bandgap (BG) – This module provides a stable reference voltage for Brown-out Reset, ADC, Flash, Comparators and low-voltage detect. The ADC, Flash and Comparators are outside the Power SIB module.
- Single core voltage regulator (CLDO)-based architecture is used in the RF-Analog section.
- Feed the voltage regulator (CLDO) with 1.35V ± 5% from a combination of DC-DC and MLDO and external board filtering.
- Brown-out Reset (BOR) – Use the BOR module to monitor the VDD supply voltage. This module provides a more accurate trip point but is only enabled when the POR event is inactive and the bandgap reference voltage is enabled and ready. Use this module to ensure that the supply voltage is above the minimum operating voltage needed for program memory reads to be valid.
- Zero-power BOR (ZPBOR) – Use this low-power BOR during Deep Sleep operation. The user can only enable the ZPBOR when DSZPBOR configuration bit is '1'.
- Flash Low-Voltage Detect (LVD) uses PLVD.
- Master Clear Filter (MCLRF) – NMCLR generates a device Reset request based on the state of a device input pin. To minimize the effects of noise and to avoid unwanted Reset conditions, the MCLRF function filters the input pin to assure a specific pulse duration of the low input pulse.

Note: Ignore the nominal pulses below 400 ns.

- Programmable Low-Voltage Detect of VDD (PLVD) consists of the following sub-modules:
  - LVD comparator
  - Resistor ladder
  - Analog voltage switch
  - LVD control ($V_{DDCORE\ DOMAIN}$)

7.3 V_{DD-AON} Power Domain Overview

The PIC32CX-BZ3 V_{DD-AON} power domain block consists of the following modules:

- SOSC's analog component acts as a secondary oscillator and uses a low-power 32.768 kHz crystal oscillator for accurate time keeping.
- For context saving, use the Extreme Deep Sleep (XDS) system controller with the semaphore.
- The Deep Sleep Regulator (ULPVREG) is an ultra-low-power regulator that provides power during deep sleep modes and/or retention power to the rest of the system in various modes of operation. $V_{DDBKUPCORE}$ is the voltage output.
- Low Power RC oscillator (LPRC) operates at a nominal frequency of 32.768 KHz.

7.4 $V_{DDBKUPCORE}$ Power Domain

The PIC32CX-BZ3 $V_{DDBKUPCORE}$ domain consists of the following modules:

- Real Time Clock Calender (RTCC)
- Deep Sleep WDT (DSWDT)
- Deep Sleep System Controller (DSCTRL)
- 32.768 kHz Oscillator Controller (digital) (SOSC_DIG)

7.5 PMU Controller

The PMU controller acts as a control unit to monitor/program the BUCK/MLDO and provides unified control to various LDOs present on the PIC32CX-BZ3 device. The PMU controller does not contain LDOs or the BUCK/MLDO source. All the regulators are represented outside of the PMU controller unit and fed into the PMU controller.

7.6 Voltage Regulators

The following voltage regulators are available in power subsystem:

- VREG (DC-DC/MLDO):
  - MLDO mode – Linear voltage regulator generating 1.35V for powering CLDO and RF LDO; operates from 1.9-3.6V
  - DC-DC – Switching voltage regulator generating 1.35V; operates from 2.3-3.6V
- ULP_VREG – Ultra-low-power voltage regulator for operation in back-up mode
- RF LDO – Powering the different blocks of the RF subsystem
- CLDO – Powering the VDDCORE of PIC32CX-BZ3

7.7 Power Supply Modes

The PIC32CX-BZ3 device supports a single power supply from 1.9-3.6V. 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:

- RUN mode – The PIC32CX-BZ3 device automatically gets into RUN mode upon power-up. This is the default state of the device.

  - MLDO mode – Provide a soft start-up by using the MLDO, limiting the charging current. The MLDO also helps to extend the supply voltage range down to 1.9V below Buck BOR. This mode does not require external inductor.
  - BUCK mode – The most efficient mode when the CPU and peripherals are running. In this mode, the DC-DC converter powers the SoC. This mode requires external LC-filtering and appropriate decoupling on-board before supplying power to other blocks.
    Note: BUCK mode is not supported in this silicon version due to an accuracy issue.

    BUCK mode has the following two operating modes:
    - PWM (Pulse Width Modulation) mode – Buck can deliver the highest output current with good efficiency. The internal switching clock in this mode is 1-2 MHz. In PWM mode, it is expected to reach an efficiency of 85%.
    - PSM (Pulse Skipping Mode) – This mode is recommended when the load current demand is low. The PSM mode is a type of frequency modulation scheme with an efficiency of up to 80%.

- Low-power modes – The PIC32CX-BZ3 supports various low-power modes; Sleep, Deep-Sleep (DS) and Extreme Deep Sleep (XDS). See *Power Management Unit (PMU)* from Related Links for more details on how to transition from RUN mode to low-power modes.
- IDLE mode – See *Power Management Unit (PMU)* from Related Links.

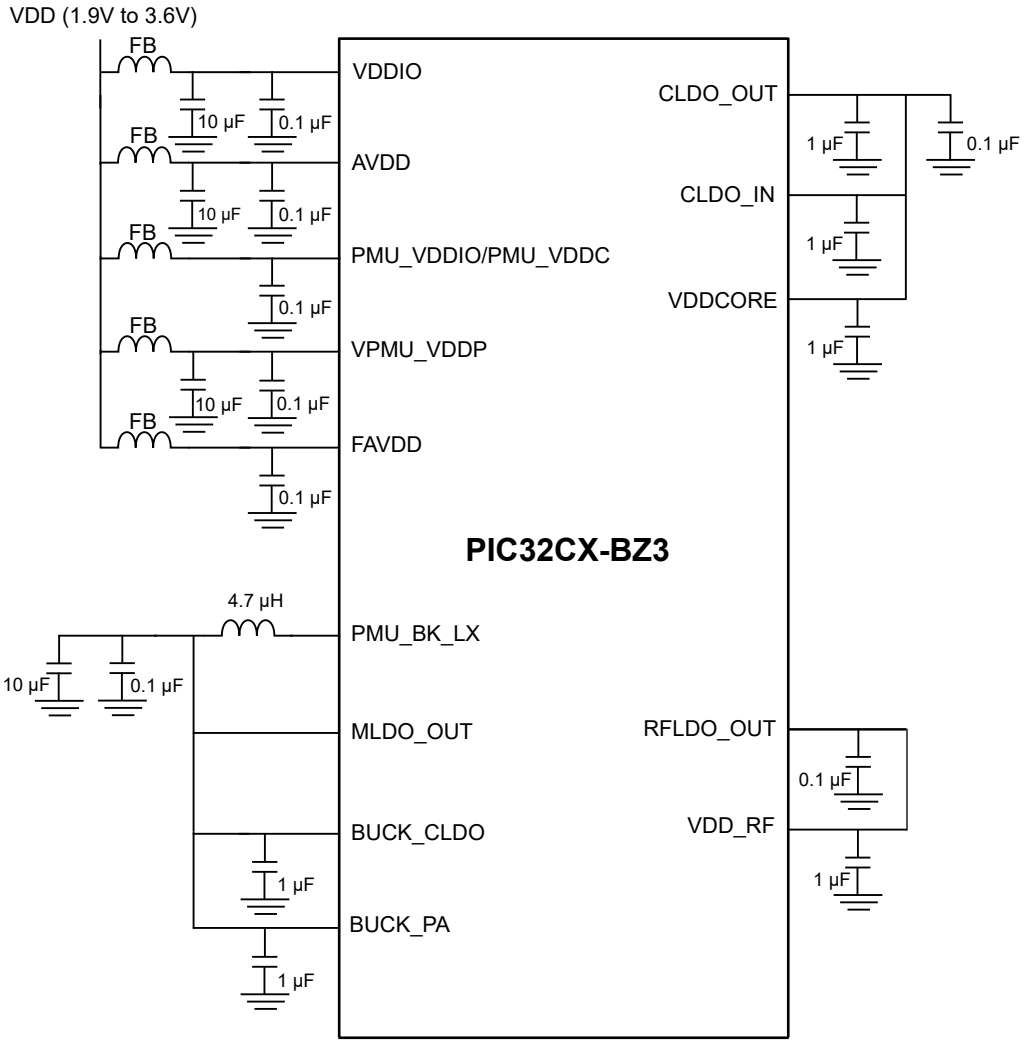
The on-the-fly software can do the selection between Switching (BUCK) mode and Linear (MLDO) mode. Be sure to design the power supply according to the type of mode in use.

Related Links

18. Power Management Unit (PMU)

7.8 Typical Power Supply Connection for SoC

Figure 7-2. Typical Power Supply Connection for SoC

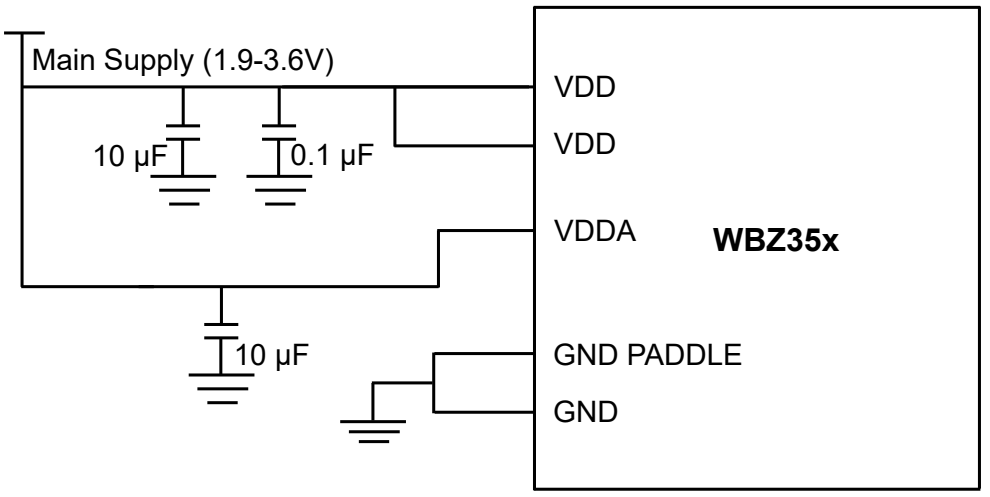


7.9 Typical Power Supply Connection for WBZ35x Module

The WBZ35x modules require only a single power supply on the VDD pins of the module.

- VDD ranges from 1.9-3.6V for non-ECC WBZ35x modules.

Figure 7-3. WBZ35x Module Schematics with VDD and Optional Bulk Capacitors



7.10 Power-Up Sequence

Characteristics of power-up sequence are as follows:

- The VDD/AVDD domains must rise at the same time.
- At Power-on Reset, the PIC32CX-BZ3 operates in the MLDO mode.
- The LDOs start with their default settings and VDDCORE is powered-up.
- The RF block is maintained in the Sleep mode during the power-up time.
- The PMU controller switches the MLDO mode to DC-DC mode based on device settings in the Flash BCFG/TCFG area.

7.10.1 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.

7.10.2 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.

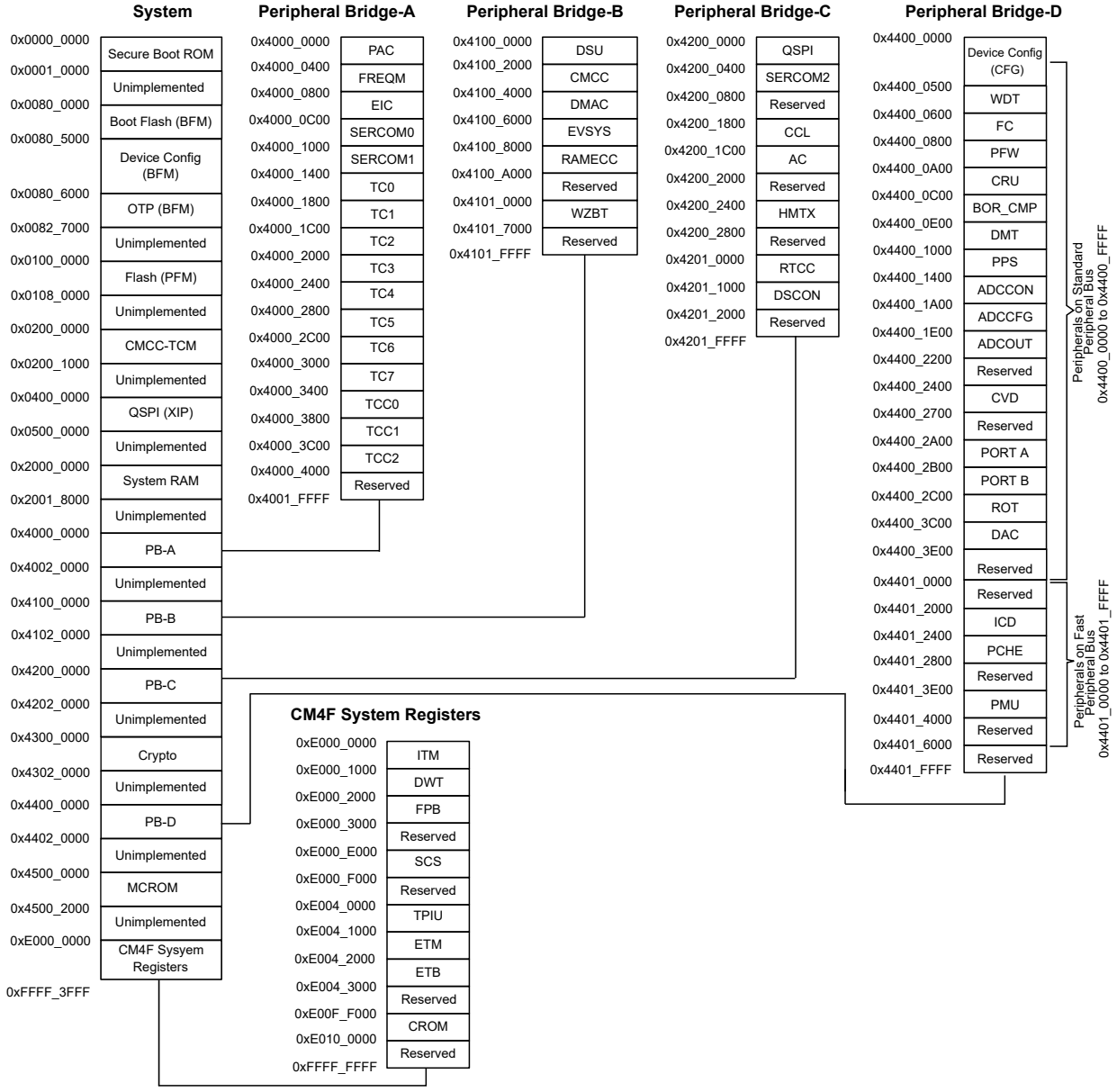
7.10.3 BOR and POR

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-BZ3 and the WBZ35x 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



Notes:

- Access attempts to any unimplemented memory location generate a bus error.
- CFGCON1.QSCHE_EN controls the QSPI (XIP) space “cacheable and bufferable” attribute.
- The MCROM is the microcode crypto ROM associated with the crypto engine. Only the crypto engine has read permissions to MCROM.
- The DAP derives the base address of the components from CoreSight ROM (CROM) entry values.
- Component base address = CROM base address + CROM entry value
- For more details on each component register space, refer to the *Cortex-M4 Technical Reference Manual r0p0* (CM4F) documentation.

8.1 Embedded Memories

- Internal ROM for secure boot
- Internal high-speed Flash
- Internal high-speed RAM with retention capability in the low power modes
- eFuse One-Time-Programmable memory secure boot key storage

8.2 Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed, and they are never remapped in any way, even during boot.

Table 8-1. Physical Memory Map

Memory	Start Address	Size
		PIC32CX510x/WBZ35x
Boot ROM	0x00000000	64 KB
Boot Flash	0x00800000	32 KB
Embedded Program Flash	0x01000000	512 KB
Embedded SRAM	0x20000000	96 KB
Peripheral Bridge A	0x40000000	—
Peripheral Bridge B	0x41000000	
Peripheral Bridge C	0x42000000	
Peripheral Bridge D	0x44000000	
eFuse	—	3072 bits

8.3 Boot ROM

A 64-KB ROM is dedicated for the secure boot firmware as a part of Root of Trust (RoT) macro. On a POR, secure boot firmware, which actually authenticates the rest of the program image in the Flash, is always run. The RoT macro stores the keys and credentials required for code authentication that are a part of the eFuses.

8.4 Flash Memory Parameters

A single page contains 4K bytes, which is applicable to all the device part numbers listed in the configuration summary chapter (see *Configuration Summary* from Related Links).

The number of pages available in a device part number will vary depending on the available maximum Flash memory size.

Equation 8-1. Calculating Flash Memory

$$Number of Pages = \frac{Flash Size(Bytes)}{Page Size(Bytes)}$$

Related Links
2. Configuration Summary

8.5 eFuse Memory

An eFuse is OTP memory that exists as a part of the RoT macro to facilitate key and other required credential storage needed by secure boot.

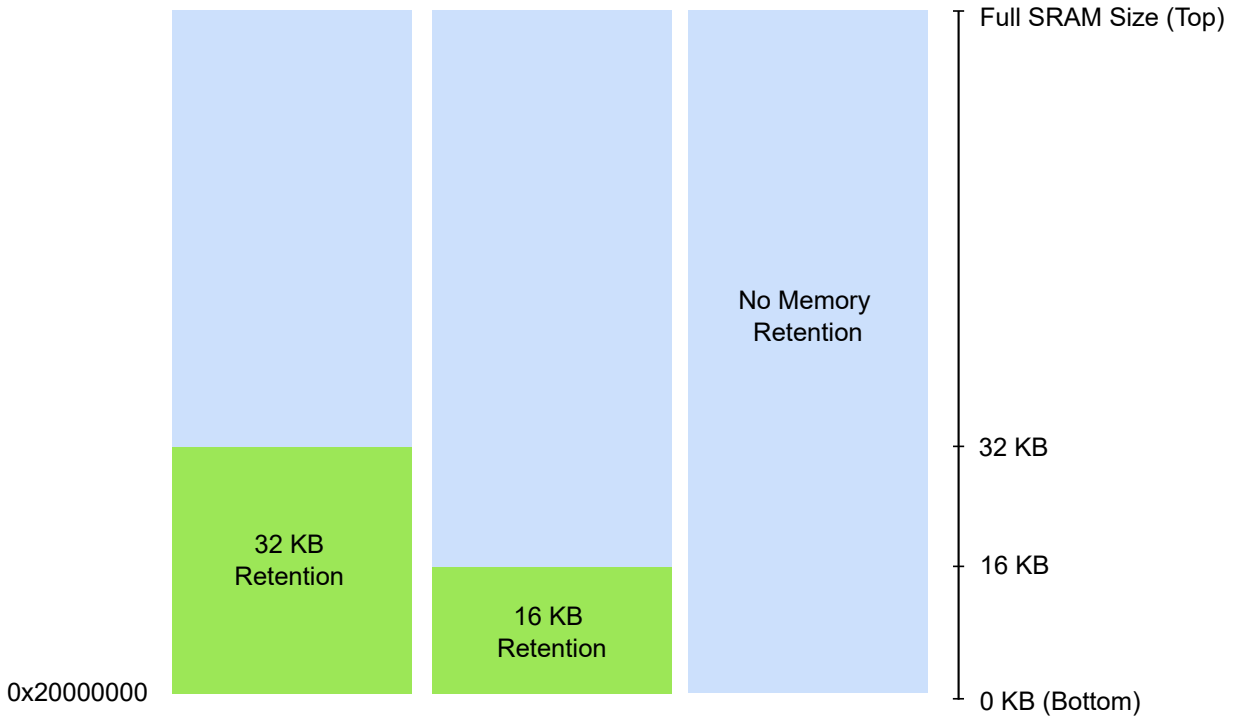
8.6 SRAM Memory Configuration

Retention

Depending on the application and power budget needs, part of the system memory can be retained in the Deep Sleep mode. The amount of the SRAM retained in this mode is software selectable, by writing the WCMSIZ register in the PMU module, up to 32 KB of SRAM.

By default, no retention is selected.

Figure 8-2. Retention Options



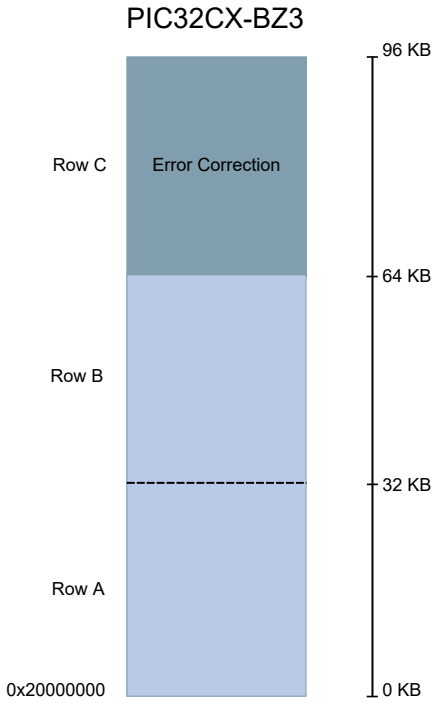
RAM Error Correction

For safety applications, the PIC32CX-BZ3 family embeds Error Correction Codes (ECC) to detect and correct single bit errors or to enable dual error detection for the system memory. The ECC is software selectable through the DEVCFG0.FRECCDIS bit in the boot Flash device configuration. By default, ECC is disabled.

ECC can be applied only for 32 KB of SRAM. When enabled, the top 32 KB of memory is reserved to store the ECC, and it is not be available for the application.

Therefore, when ECC is enabled, the usable system RAM is 64 KB (96-32 KB) for the 96 KB data RAM variant. ECC support for row A or row B can be selected using the CFGCON1.ECC_SEL_MEM bit. If CFGCON1.ECC_SEL_MEM is '0', ECC supports the contents in Row A. If CFGCON1.ECC_SEL_MEM is '1', ECC supports the contents in row B.

Figure 8-3. Memory with RAM Error Correction

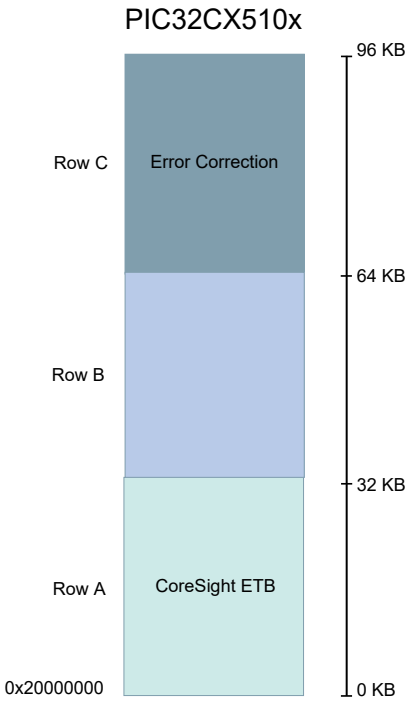


Note: ECC is not possible if the user enables SRAM retention.

CoreSight ETB Connection

When enabled, the bottom 32 KB system memory space is reserved for CoreSight (Embedded Trace Buffer) ETB debug usage. Therefore, when CoreSight ETB is enabled, the usable system RAM is 64 KB (96-32 KB) for the 96 KB data RAM variant. The following figure illustrates an example where both ECC and CoreSight ETB are enabled.

Figure 8-4. Memory with ECC and CoreSight ETB



8.7 Boot Flash Device Configuration Word

The PIC32CX-BZ3 device provides several user-writable configuration registers related to the configuration and operation of the system. The device configuration words are programmed in boot Flash memory (NVR pages) and get loaded on equivalent registers after the device Reset. The following table shows the device configuration words in Boot Flash.

Table 8-2. Boot Flash and Device Configuration Word

Physical Address	Register Name	Bit Range	31:0							
0x00805E88	ALTFUSERID	31:0	See *USER_ID* in the *CFG Register Summary* from Related Links							
0x00805E8C	ALTDEVCFG4	31:0	See *CFGCON4(L)* in the *CFG Register Summary* from Related Links							
0x00805E90	ALTDEVCFG2	31:0	See *CFGCON2(L)* in the *CFG Register Summary* from Related Links							
0x00805E94	ALTDEVCFG1	31:0	See *CFGCON1(L)* in the *CFG Register Summary* from Related Links							
0x00805E98	ALTDEVCFG0	31:0	See *CFGCON0(L)* in the *CFG Register Summary* from Related Links							
0x00805E9C	ALTFBCFG0	31:0	See *BCFG0* in the *CFG Register Summary* from Related Links							
0x00805F88	FUSERID	31:0	See *USER_ID* in the *CFG Register Summary* from Related Links							
0x00805F8C	DEVCFG4	31:0	See *CFGCON4(L)* in the *CFG Register Summary* from Related Links							
0x00805F90	DEVCFG2	31:0	See *CFGCON2(L)* in the *CFG Register Summary* from Related Links							
0x00805F94	DEVCFG1	31:0	See *CFGCON1(L)* in the *CFG Register Summary* from Related Links							
0x00805F98	DEVCFG0	31:0	See *CFGCON0(L)* in the *CFG Register Summary* from Related Links							
0x00805F9C	FBCFG0	31:0	See *BCFG0* in the *CFG Register Summary* from Related Links							
0x00805FBC	FCPN0	7:0	—							
		15:8	—							
		23:16	—							
		31:24	—	—	—	CP	—	—	—	—

Related Links

22.8. Register Summary

8.8 Boot Flash Code Protection Register

Name: FCPN0
Offset: 0x00805FBC
Reset: 0x00000000
Property: —

Note: Offset is an absolute address of this register.

Bit	31	30	29	28	27	26	25	24
				CP				
Access				R				
Reset				p				

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
Access								
Reset								

Bit 28 – CP Flash (BFM, PFM) Code Protect

Note: The value of this bit is the inverse polarity of the value read from the BCFG0 register.

Value	Description
0	Protection is disabled
1	Protection is enabled

9. Processor and Architecture

9.1 Cortex M4F Processor

The PIC32CX-BZ3 implements the ARM Cortex-M4F processor, which is a high performance 32-bit processor with secure boot feature. 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-M4F 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 the 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.

9.1.1 System Level Interface

The Cortex-M4F 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-M4F processor has an MPU that provides:

- Fine grain memory control
- Enabling applications to utilize multiple privilege levels
- Separating and protecting code

- Data
- Stack on a task-by-task basis

In the automotive sector, these requirements are becoming critical in many embedded applications.

9.1.2 Integrated Configurable Debug

The Cortex-M4F 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) provide up to eight hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to eight 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 there is a requirement for a patch or not. If there is a requirement for a patch, 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.

9.1.3 Cortex-M4F Processor Features and Configuration

- The 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.

Table 9-1. Cortex-M4F Processor Features and Configuration

Features	PIC32CX-BZ3 Configuration
Interrupts	43
Number of priority bits	3 = eight levels of priority
Data endianness	Little-endian

.....continued	
Features	**PIC32CX-BZ3 Configuration**
SysTick Timer calibration value	0x80000000
MPU	Present
Debug support level	3 = Full debug plus DWT data matching
Trace support level	2 = Full trace. Standard trace plus ETM
JTAG	Not present
Bit Banding	Not present
FPU	Present

9.1.4 Cortex-M4F Core Peripherals

Table 9-2. Cortex-M4F Core Peripherals

Cortex-M4F Core Peripherals	Description
Nested Vectored Interrupt Controller	The Nested Vector Interrupt Controller is an embedded interrupt controller that supports low latency interrupt processing.
System Control Block	The System Control Block (SCB) is the programmer's model interface to the processor. It provides system implementation information and system control, including configuration, control and reporting of system exceptions. For more details, refer to the *Cortex-M4 Technical Reference Manual*.
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. For more details, refer to the *Cortex-M4 Technical Reference Manual*.
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. For more details, refer to the *Cortex-M4 Technical Reference Manual*.
Floating-Point Unit	The Floating Point Unit (FPU) provides IEEE 754-compliant operations on single-precision, 32-bit, floating-point values. For more details, refer to the *Cortex-M4 Technical Reference Manual*.

9.1.5 Cortex-M4F Address Map

Table 9-3. Cortex-M4F 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

9.2 Nested Vector Interrupt Controller (NVIC)

9.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the PIC32CX-BZ3 family devices supports 43 interrupts with eight different priority levels. For more details, refer to the *Cortex-M4 Technical Reference Manual* (www.arm.com).

9.2.2 Interrupt Line Mapping

The following table provides details about each of the interrupt lines that is connected to one peripheral instance. 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 configuring it in the peripheral's Interrupt Enable 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 NVIC interrupt lines (see the following table).

An interrupt request sets 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.

Table 9-4. NVIC Interrupt Mapping

Module	Source	NVIC Line
EIC NMI – External Interrupt Control	NMI	NMI
RTCC – Real-Time Counter and Calendar	CMP A 0..3	0
	OVF A	
	PER A 0..7	
	TAMPER A	
EIC – External Interrupt Controller	EXTINT 0..3	1
FREQM – Frequency Meter	DONE	2
FC – Flash Controller and PCHE	Flash Controller Program/erase complete	3
	PFW CRC Done	
	PCACHE	
PORT-A	PortA Input Change Interrupt	4
PORT-B	PortB Input Change Interrupt	5
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	

.....continued

Module	Source	NVIC Line
PAC – Peripheral Access Controller	ERR	10
RAMECC - RAM Error Correction Code	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	
TCC0 – Timer Counter Control 0	CNT A	14
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0..5	
TCC1 – Timer Counter Control 1	CNT A	15
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0..5	

.....continued

Module	Source	NVIC Line
TCC2 – Timer Counter Control 2	CNT A	16
	DFS A	
	ERR A	
	FAULTA A	
	FAULTB A	
	FAULT0 A	
	FAULT1 A	
	OVF	
	TRG	
	UFS A	
	MC 0..1	
TC0 – Basic Timer Counter 0	ERR A	17
	MC 0	
	MC 1	
	OVF	
TC1 – Basic Timer Counter 1	ERR A	18
	MC 0	
	MC 1	
	OVF	
TC2 – Basic Timer Counter 2	ERR A	19
	MC 0	
	MC 1	
	OVF	
TC3 – Basic Timer Counter 3	ERR A	20
	MC 0	
	MC 1	
	OVF	
TC4 – Basic Timer Counter 4	ERR A	21
	MC 0	
	MC 1	
	OVF	
TC5 – Basic Timer Counter 5	ERR A	22
	MC 0	
	MC 1	
	OVF	
TC6 – Basic Timer Counter 6	ERR A	23
	MC 0	
	MC 1	
	OVF	
TC7 – Basic Timer Counter 7	ERR A	24
	MC 0	
	MC 1	
	OVF	

.....continued		
Module	**Source**	**NVIC Line**
ADC - Analog-to-Digital Converter	ADC_GIRQ	25
	ADC_DIRQ0, ADC_DIRQ1	
	ADC_AIRQ0, ADC_AIRQ1	
	ADC_FLT	34
	ADC_FCC	35
	ADC_BGVR_RDY	36
AC – Analog Comparators	COMP 0	26
	COMP 1	
	WIN 0	
CRYPTO	INT0	27
	INT1	28
	INT2	41
QSPI – Quad SPI interface	QSPI	29
Wireless Subsystem (WZBT)	ZB_INT0	30
	BT_INT0	31
	BT_INT1	32
	ARBITER	33
	CLKI_WAKEUP_NMI	37
	BT_LC	42
CVD - Capacitive Voltage Divider Controller	CVD	38
SERCOM2 Serial Communication Interface 2(1) I2CM, I2CS	0 1 2 3 4 5 6 7	40

Note:

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

9.3 High-Speed Bus System

The high-speed bus system matrix connects a multitude of initiator logic cores/IPs to a multitude of target logic cores/IPs, supporting AHB2/APB2 buses.

9.3.1 Features

High-Speed Bus Matrix has the following features:

- AMBA Advanced High-performance Bus (AHB Lite)-compliant interfaces
- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different Initiator to different target
- 32-bit data bus
- APB compliant user interface

9.3.2 Configuration

Figure 9-1. High-Speed Bus Matrix Inter-connectivity

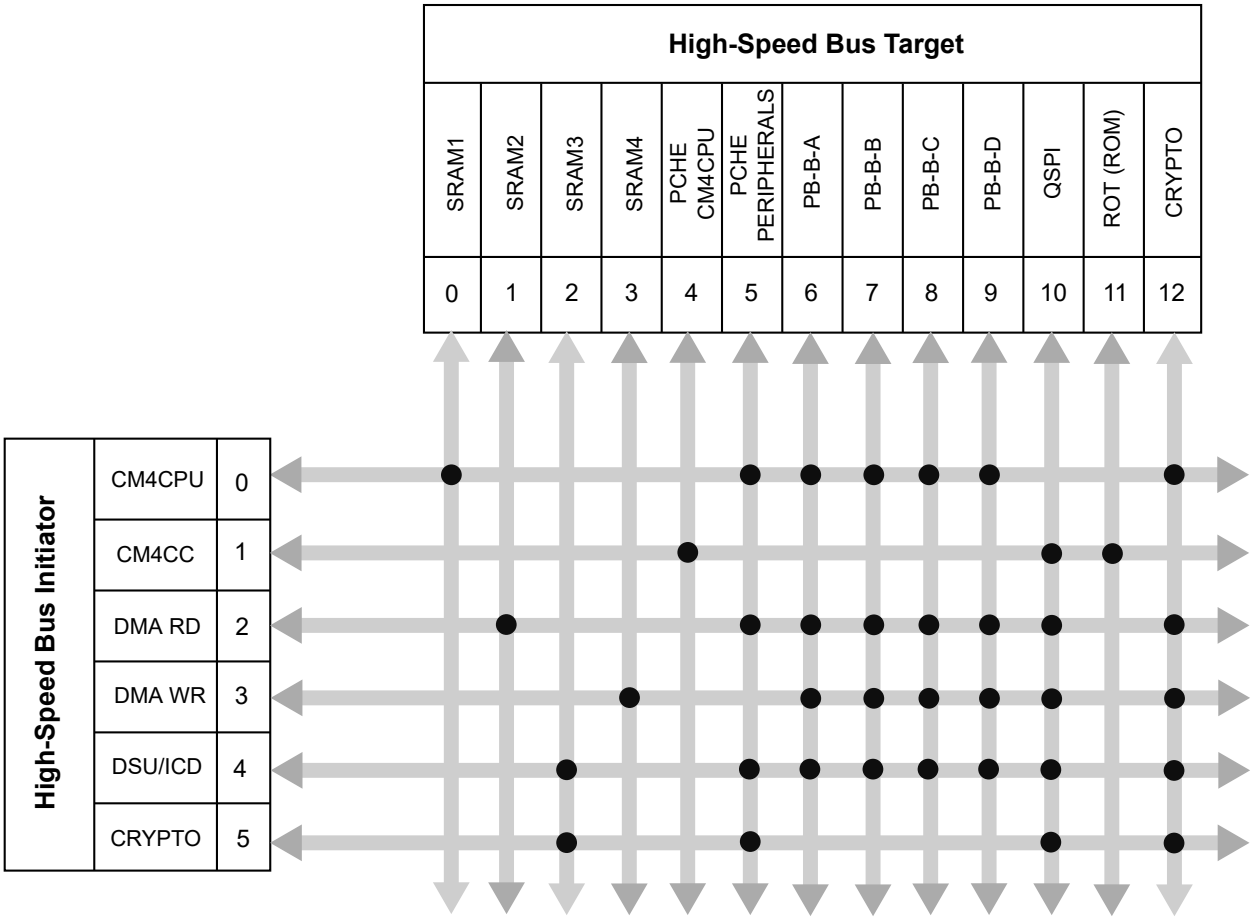


Table 9-5. High Speed Bus Matrix Initiator

High-Speed Bus Matrix Initiator	Initiator ID
CM4CPU – Cortex-M4F Processor	0
CM4CC – Cortex M Cache Controller	1
DMA RD – DMA-Read	2
DMA-WR – DMA-Write	3
DSU/ICD (Test mode only) – Device Service Unit/In-Chip Debugger	4
CRYPTO	5

Table 9-6. High-Speed Bus Matrix Target

High-Speed Bus Matrix Target	Target ID
SRAM1 – SRAM Port 1	0
SRAM2 – SRAM Port 2	1
SRAM3 – SRAM Port 3	2
SRAM4 – SRAM Port 4	3
PCHE – Prefetch Cache of CM4CC	4
PCHE – Prefetch Cache of Peripherals	5
PB-B-A – Peripheral Bridge A	6

.....continued	
High-Speed Bus Matrix Target	**Target ID**
PB-B-B – Peripheral Bridge B	7
PB-B-C – Peripheral Bridge C	8
PB-B-D – Peripheral Bridge D	9
QSPI – Quad SPI Interface	10
ROT – Root of Trust	11
CRYPTO	12

10. Prefetch Cache (PCHE)

10.1 Overview

The prefetch cache is a performance-enhancing module included in the PIC32CX-BZ3 devices, along with the L1 cache (Cortex M Cache Controller) to the Cortex-M4F CPU.

10.2 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

10.3 Overview

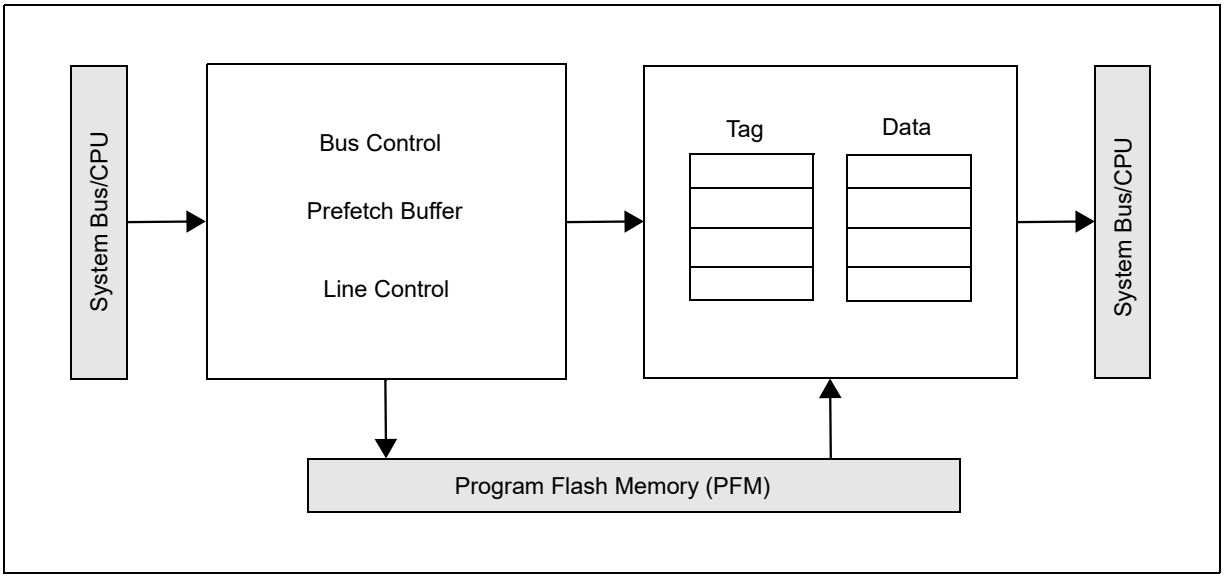
When running the prefetch module at high-clock rates, insert the Wait states into Program Flash Memory (PFM) read transactions to meet the access time of the PFM. The user can hide the Wait states to the core by prefetching and storing the instructions in a temporary holding area that the CPU can access 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 10-1. Prefetch Cache Block Diagram



10.3.1 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 10-2. Tag Line

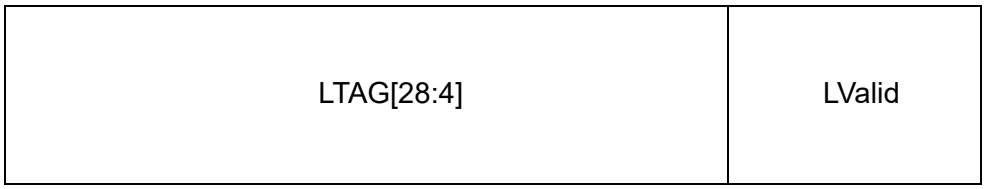
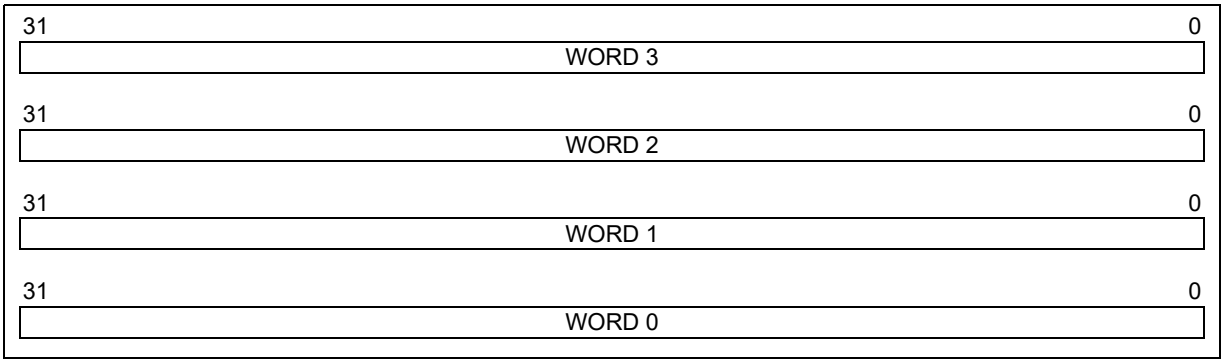


Figure 10-3. Data Line



10.4 Product Dependencies

Not applicable.

10.4.1 I/O Lines

Not applicable.

10.4.2 Power Management

10.4.2.1 Standby Sleep Mode

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

10.4.2.2 Idle Mode

When the device enters 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.

10.4.3 Clocks

The PCHE interfaces with the CPU through the AHB (SYS_CLK).

10.4.4 DMA

Not applicable.

10.4.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the PCACHE interrupt(s) requires the NVIC interrupt controller to be configured first.

10.4.6 Events

Not applicable.

10.4.7 Debug Operation

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

10.4.8 Register Access Protection

Not applicable.

10.4.9 Analog Connections

Not applicable.

10.5 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'.

10.6 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.

10.7 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.

10.8 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.

10.9 Effects of Reset

10.9.1 On Reset

Upon a device Reset, the following occurs:

- All lines are invalidated
- All tag bits are cleared

10.9.2 After Reset

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

Related Links

10.12.1. CHECON

10.10 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.

10.10.1 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 PFMDED 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.

10.10.2 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 generates an interrupt. 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 '0'. 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 generates an ECC SEC interrupt. The ECC SEC interrupt persists as long as the PFMSECEN and PFMSEC bits remain set.

10.11 Register Summary

See the *PCHE* module in the *Product Memory Mapping Overview* from Related Links for base address.

Note: CHECON and CHESTAT 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	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					PFMDED	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]							

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

10.12 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

10.12.1 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/W
Reset				0				1

Bit	7	6	5	4	3	2	1	0
	PFMSECEN		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 enabled
0	Caching disabled (and all lines invalidated)

Bit 25 – DCHEEN Data Cache Enable bit

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

Bit 24 – ICHEEN Instruction Data Cache Enable bit

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

Bit 22 – ACHEINV Manual Invalidate Control for Peripheral Data Cache

Note: 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: 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:

- Predictive Prefetch Buffer (PFB) is included with iCache invalidate.
- 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	Enable performance counters
0	Disable performance counters

Bit 8 – ADRWS Address Wait State Enable

Total Flash wait states are ADRWS + PFMWS.

Note: CPU hang is observed when CHECON.ADRWS configuration switches from '1' to '0' and a Flash read access. To Avoid CPU hangs, execute the CHECON configuration from SRAM or Boot ROM during system initialization until the configuration change is done, then resume execution from Flash after configuration is set.

Value	Description
1	Add 1 address Wait state - Allowing for higher clock frequencies
0	Add 0 address Wait states - Allowing for higher performance at lower clock frequencies

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:

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

Related Links

43. Electrical Characteristics

10.12.2 CHESTAT - Prefetch Module Status Register

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

Bit	31	30	29	28	27	26	25	24
					PFMDED	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 – PFMDED 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
9.2. Nested Vector Interrupt Controller (NVIC)

10.12.3 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.

10.12.4 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.

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

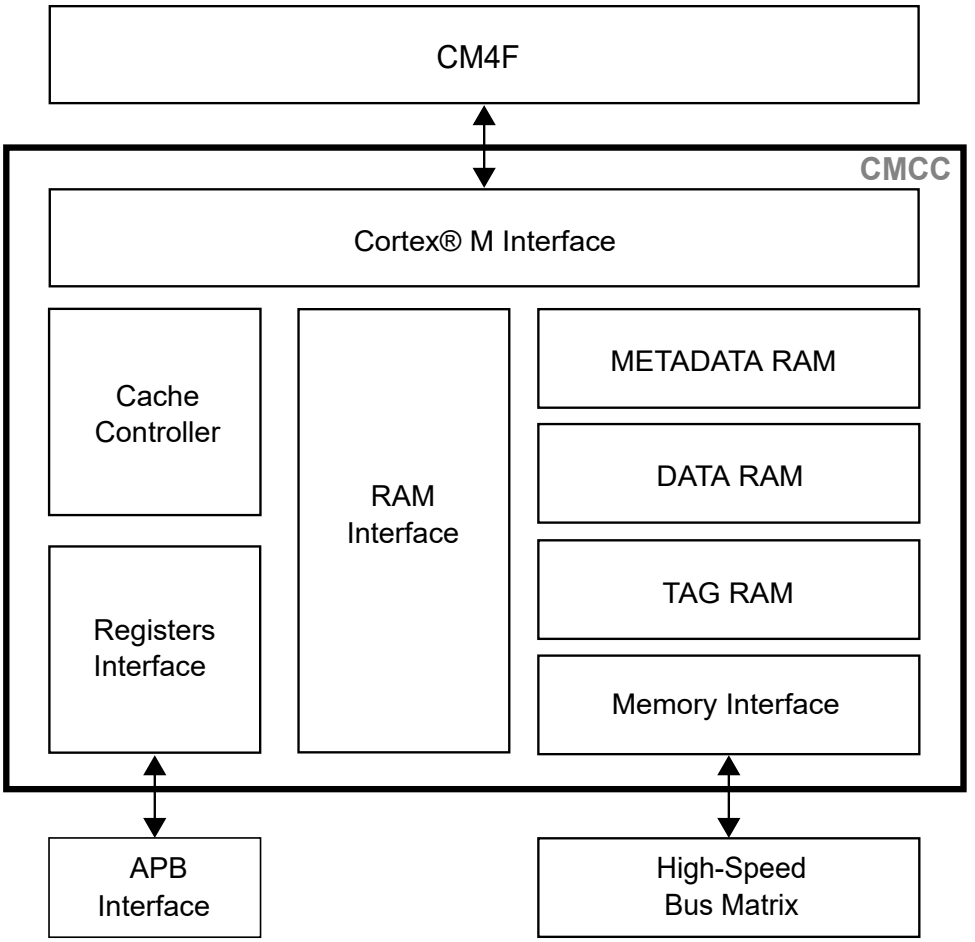
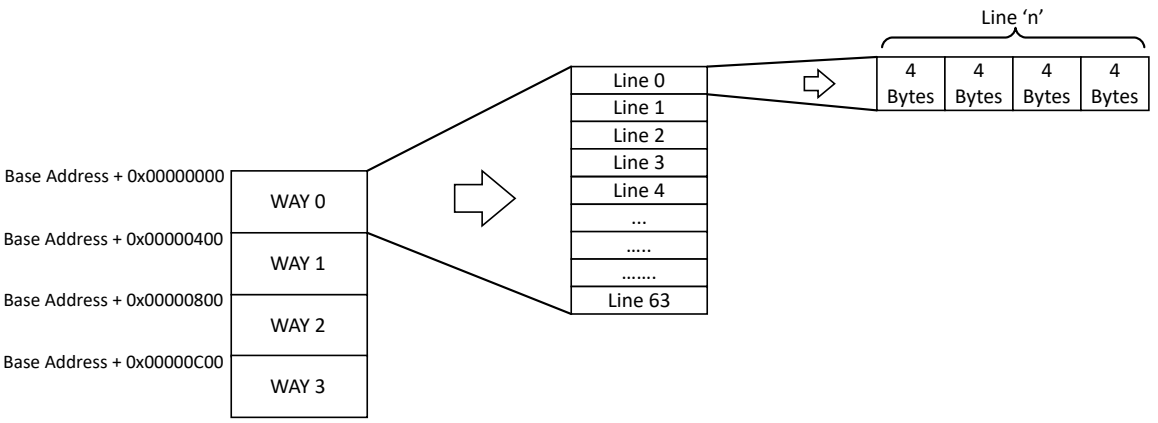


Figure 11-2. CMCC Organization



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

The CMCC interfaces with the CPU through the AHB (SYS_CLK) and is connected to the APB bus (PB2_CLK) interface for its configuration.

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 user can activate the cache controller by using 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 that 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 CMCC does not update the locked way 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 resets the round robin algorithm of the invalidated line, which becomes 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 forces the cache to refill the previous invalidated line in the right way. Validate only the first byte. The cache automatically refills 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 user can use the remaining WAYS as I-cache or D-cache as per the requirement.

11.6.7 Tightly Coupled Memory

Users can use a part of the cache as Tightly Coupled Memory (TCM). The Cache Size Configuration determines the cache size 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 user can obtain the TCM start address 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 *Product Memory Mapping Overview* from Related Links.

Table 11-1. TCM Sizes

Maximum Cache	Configured Cache	TCM Size
4 KB	4 KB	0 KB
4 KB	2 KB	2 KB
4 KB	1 KB	3 KB
4 KB	0 KB	4 KB

The TCM is also accessible in its maximum size in case of disabling the CMCC. The TCM does not need to be locked 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

If issuing an invalidate by line command, 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 '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 '0' 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 '1' to the Cache Controller Invalidate All bit in the Cache Maintenance 0 register (MAINT0.INVALL).
- Enable the CMCC by writing '1' to CTRL.CEN.

11.6.9 Cache Performance Monitoring

The Cortex M cache controller includes a programmable monitor/32-bit counter. The user can configure the monitor to count the number of clock cycles, the number of data hit or the number of instructions 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 (for example, 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 RAM Properties

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

Table 11-2. Access to RAM

Access Condition	Data RAM	Tag RAM	Metadata RAM
CPU access when CMCC DISABLED	Read/Write	no Read/Write - hardfault	no Read/Write - hardfault
CPU access when CMCC ENABLED	CACHE section configured: Read/Write(1) TCM section configured: Read/Write	no Read/Write - hardfault	no Read/Write - hardfault
Debugger access when CMCC DISABLED	Read/Write	Read/Write	Read/Write
Debugger access when CMCC ENABLED	CACHE section configured: Read/Write(1) TCM section configured: R/W	no Read/Write	no Read/Write
CPU Test access when CMCC DISABLED	Read/Write	Read/Write	Read/Write
CPU Test access when CMCC ENABLED	CACHE section configured: Read/Write(1) TCM section configured: Read/Write	no Read/Write	no Read/Write

Note:

1. A write operation in this zone can corrupt the coherency of the cache. There is a need for an invalidate operation.

11.8 Register Summary

See *CMCC* module in the *Product Memory Mapping Overview* from Related Links for base address.

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.

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	GCLKDIS
		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]							
		15:8					INDEX[7:4]			
		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]							

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

11.9 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 in case of disabling the peripheral. The enable-protected property denotes the enable protection in each individual 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

11.9.1 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. 0x02 is the value read for PIC32CX-BZ3 devices, as cache line size is 16 bytes.

Value	Name	Description
0x0	CLSIZE_4B	Cache line size is 4 bytes
0x1	CLSIZE_8B	Cache line size is 8 bytes
0x2	CLSIZE_16B	Cache line size is 16 bytes
0x3	CLSIZE_32B	Cache line size is 32 bytes
0x4	CLSIZE_64B	Cache line size is 64 bytes
0x5	CLSIZE_128B	Cache line size is 128 bytes
0x6-0x7	—	Reserved

Bits 10:8 – CSIZE[2:0] Cache Size

This bit field configures the cache size. 0x02 is the value read for PIC32CX-BZ3 devices, as cache size is 4 KB.

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	CSIZE_8KB	Cache size is 8 KB
0x4	CSIZE_16KB	Cache size is 16 KB
0x5	CSIZE_32KB	Cache size is 32 KB
0x6	CSIZE_64KB	Cache size is 64 KB
0x7	—	Reserved

Bit 7 – LCKDOWN Lock Down Supported

Value	Description
0	Lockdown is not supported
1	Lockdown is supported

Bits 6:5 – WAYNUM[1:0] Number of Way

This bit field configures the mapping of the cache. 0x02 is the value read for PIC32CX-BZ3 devices, as the device supports 4-Way set associative.

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

Value	Description
0	Round robin policy is not supported
1	Round robin policy is supported

Bit 3 – LRUP Least Recently Used Policy Supported

Value	Description
0	Least recently used policy is not supported
1	Least recently used policy is supported

Bit 2 – RANDP Random Selection Policy Supported

Value	Description
0	Random victim selection is not supported
1	Random victim selection is supported

Bit 1 – GCLK Dynamic Clock Gating

Value	Description
0	Cache controller does not support clock gating
1	Cache controller uses dynamic clock gating

Bit 0 – AP Access Port Access Allowed

Value	Description
0	Access port access is disabled
1	Access port access is enabled

11.9.2 Cache Configuration

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

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
		CSIZESW[2:0]				DCDIS	ICDIS	GCLKDIS
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	1	0		0	0	0

Bits 6:4 – CSIZESW[2:0] Cache Size Configured by Software
This field configures the cache size.

Value	Name	Description
0x0	CONF_CSIZE_1KB	The Cache Size is configured to 1 KB
0x1	CONF_CSIZE_2KB	The Cache Size is configured to 2 KB
0x2	CONF_CSIZE_4KB	The Cache Size is configured to 4 KB
0x3	—	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.

Bit 0 – GCLKDIS 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.

11.9.3 Cache Control

Name: CTRL
Offset: 0x08
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
								CEN
Access								W
Reset								0

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

11.9.4 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 to this bit has no effect.
Reading '0' shows CMCC is disabled.
Reading '1' shows CMCC is enabled.

11.9.5 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.

11.9.6 Cache Maintenance 0

Name: MAINT0
Offset: 0x20
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
								INVALL
Access								W
Reset								0

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.

11.9.7 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

11.9.8 Cache Monitor Configuration

Name: MCFG
Offset: 0x28
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
							MODE[1:0]	
Access							R/W	R/W
Reset							0	0

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

11.9.9 Cache Monitor Enable

Name: MEN
Offset: 0x2C
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
								MENABLE
Access								R/W
Reset								0

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.

11.9.10 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.

11.9.11 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. Secure Boot ROM

12.1 Overview

The boot ROM ensures the integrity of the device at boot.

An immutable boot sequence (Secure Boot Code) is implemented into a ROM called the secure boot ROM. The secure boot ROM manages system initialization, firmware, data authentication and necessary system configurations.

For secure boot support on the PIC32CX-BZ3, a 64K ROM is dedicated to the secure boot firmware. Keys, unique device ID, storage key and secure boot key required for code authentication are stored as a part of eFuses. On every reset, secure boot firmware authenticates the rest of the program image in the Flash.

Note: The default clock FRC is 8 MHz and jumps to 64 MHz. In addition, the boot ROM has extra security features, such as device integrity checks, memory/peripherals security attributions, and secure boot, which can be executed before jumping to the Flash in the Secure state.

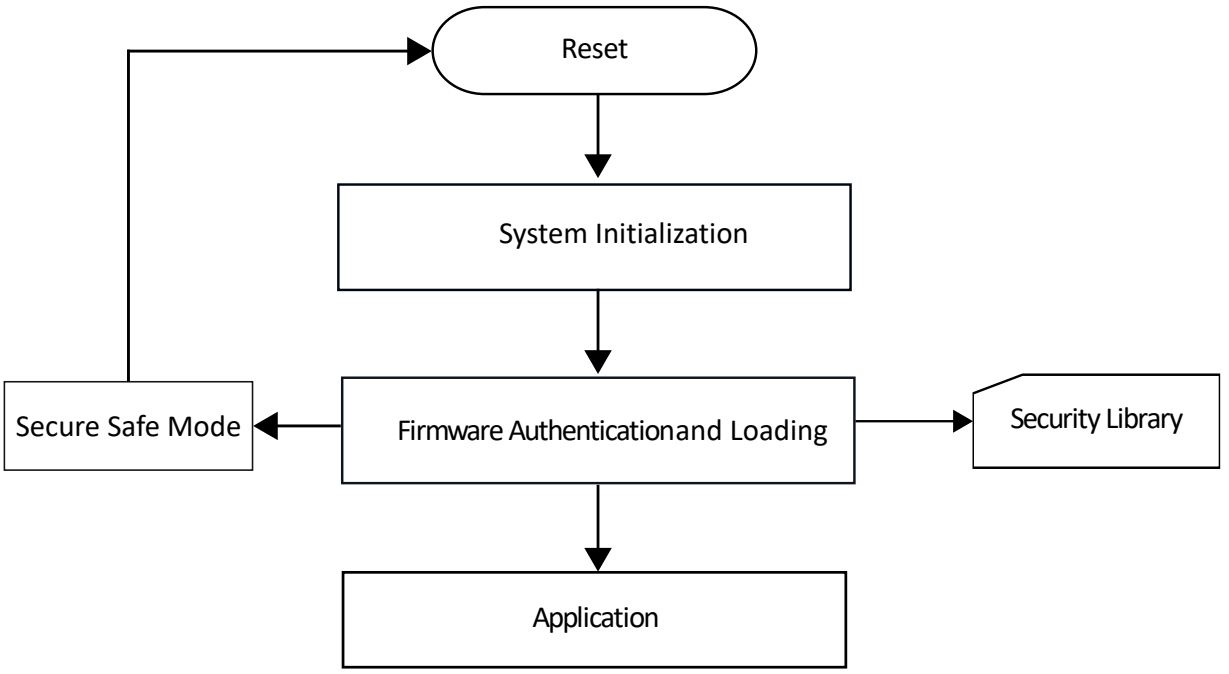
12.2 Features

- Immutable Boot Support with ROM
- Firmware Code Authentication (Only at Boot Time)
- Support for Immutable Keys
- Microchip or Customer Programmable Secure Boot Key
- Support for Secure Execution Environment
- Supports Anti-Rollback
- Support for Firmware Readable Life Cycle Counter
- Defined System Boot State

12.3 Functional Description

12.3.1 Boot ROM Flow

Figure 12-1. Boot ROM Flow



When the device comes out of reset, Boot ROM firmware starts with system initialization. After the completion of system initialization, the bootstrap code checks for a valid firmware image and executes. If the device does not find any valid image, it jumps to Secure Safe mode. For validating the firmware image, the bootstrap code uses the security library depending on the firmware images authentication scheme specified for that firmware.

Firmware programming is handled by Device Firmware Update (DFU). DSU supports the Debug mode. Also, it implements a CoreSight™ Debug ROM that provides device identification, as well as identification of other debug components within the system. See *Device Service Unit (DSU)* from Related Links.

The following options are valid images for a secured and unsecured PIC32CX-BZ3 device:

- An image with valid structure (defined in 12.3.2.1. Firmware Image Format) and without using any authentication scheme
- An image with a valid structure (defined in 12.3.2.1. Firmware Image Format) with an authentication scheme supported by the device. For supporting the authentication scheme using the public key cryptography, EFUSE must have a valid SECURE_BOOT_KEY.
- If the device is unsecured (and SECURE_BOOT_KEY), is invalid and the device's secured state is not set, the bootstrap code will look for valid images (valid sequence number, header and so on). Root of trust may not be possible in this scenario.
- If the device is secured (with SECURE_BOOT_KEY! = 0), the valid image must always use the `ECDSA p384 + SHA 384 or ECDSA p256 + SHA 256` for authentication and execute only trusted code.

Related Links

16. Device Service Unit (DSU)

12.3.1.1 DSU Mode Entry

Immediately after resetting, the hardware performs a read operation of the register bits of code protect and Device Secured State bits to determine the Device Security state. Depending on the Security state, the ROM firmware needs to perform specific actions.

12.3.1.2 System Initialization

All the necessary hardware modules that the Boot ROM firmware uses will be initialized – System clock, Interrupts and Jumping to application firmware from Boot ROM code.

12.3.2 Firmware Authentication and Loading

The firmware image validation block is the primary block that ensures the integrity/authenticity of the images that are run on a secured device.

12.3.2.1 Firmware Image Format

The firmware image comes with a metadata header, metadata payload and metadata footer that gives the ROM firmware information about location of the firmware image, authentication information, sequence number and more. The user can only execute the application (firmware image) from embedded Flash.

The following table provides details about the contents of each firmware image.

Table 12-1. Firmware Image Description

Offset	Name	Description
Metadata Header		
0x00:0x03	SEQ_NUM	Metadata sequence number of the image Little Endian (LE). Monotonically decreasing image index. Values of 0 or 0xFFFFFFFF indicate that the image is invalid. Boot ROM Image selection algorithm will prefer a valid image with the lowest sequence number.
0x04	MD_REV	Support metadata header version: 3 **Note:** This field must be set to 0x01 for this version of metadata header.
0x05	CONT_IDX	0x01 – Firmware image
0x06:0x09	IDENTIFIER	MCHP ASCII string identifier
0x0A	MD_AUTH_MTHD	Metadata authentication method 0x00 – None 0x02 – ECDSA p256 + SHA-256 0x03 – ECDSA p256 + SHA-384
0x0B	MD_AUTH_KEY	Key index for authenticating metadata 0x00 – Secure boot Key
0x0C	Reserved	0x00
0x0D	Reserved	0x00
0x0E:0x0F	PL_LEN	Metadata payload length. The payload length for this version must be 0x74.
Metadata Payload = Firmware Image Header		
0x10:0x13	FW_IMG_REV	Firmware image revision (LE)
0x14:0x17	FW_IMG_SRC_ADDR	Firmware image source address The source address of the firmware image in persistent storage
0x18:0x1B	FW_IMG_DST_ADDR	Firmware image destination address The destination address will be used as the jump address during the application transition.

.....continued

Offset	Name	Description
0x1C:0x1F	FW_IMG_LEN	Firmware image length The firmware image length must be a multiple of 4096 (count) bytes.
0x20	FW_IMG_AUTH_MTHD	Firmware image authentication method 0x00 – None 0x02 – ECDSA p256 + SHA-256 0x03 – ECDSA p384 + SHA-384
0x21	FW_IMG_AUTH_KEY	Firmware image authentication key index 0x00 – Secure boot key
0x22	Reserved	0x00
0x23	Reserved	0x00
0x24:0x83	FW_IMG_SIG	Firmware image signature: The concatenated R and S term of the ECDSA signature (P-256) of the SHA-256 hash of the firmware image specified by FW_IMG_SRC_ADDR and FW_IMG_LEN. (or) The concatenated R and S term of the ECDSA signature (P-384) of the SHA-384 hash of the firmware image specified by FW_IMG_SRC_ADDR and FW_IMG_LEN.
Metadata Footer		
0x84:0xE3	MD_SIG	Metadata payload signature: The concatenated R and S term of the ECDSA signature (P-256) of the SHA-256 hash of the firmware image specified by FW_IMG_SRC_ADDR and FW_IMG_LEN. (or) The concatenated R and S term of the ECDSA signature (P-384) of the SHA-384 hash of the firmware image specified by FW_IMG_SRC_ADDR and FW_IMG_LEN.

Table 12-2. Firmware Images Lookup Table

Embedded Flash Variant		
	Address	Size (bytes)
Secondary bootloader metadata	0x0080_0000	512
Secondary bootloader	0x0080_0200	Max: 15872
Image 0 metadata	0x0100_0000	512
Image 0	0x0100_0200	Max: 523776
Image 1 metadata	0x0104_0000	512
Image 1	0x0104_0200	Max: 261632

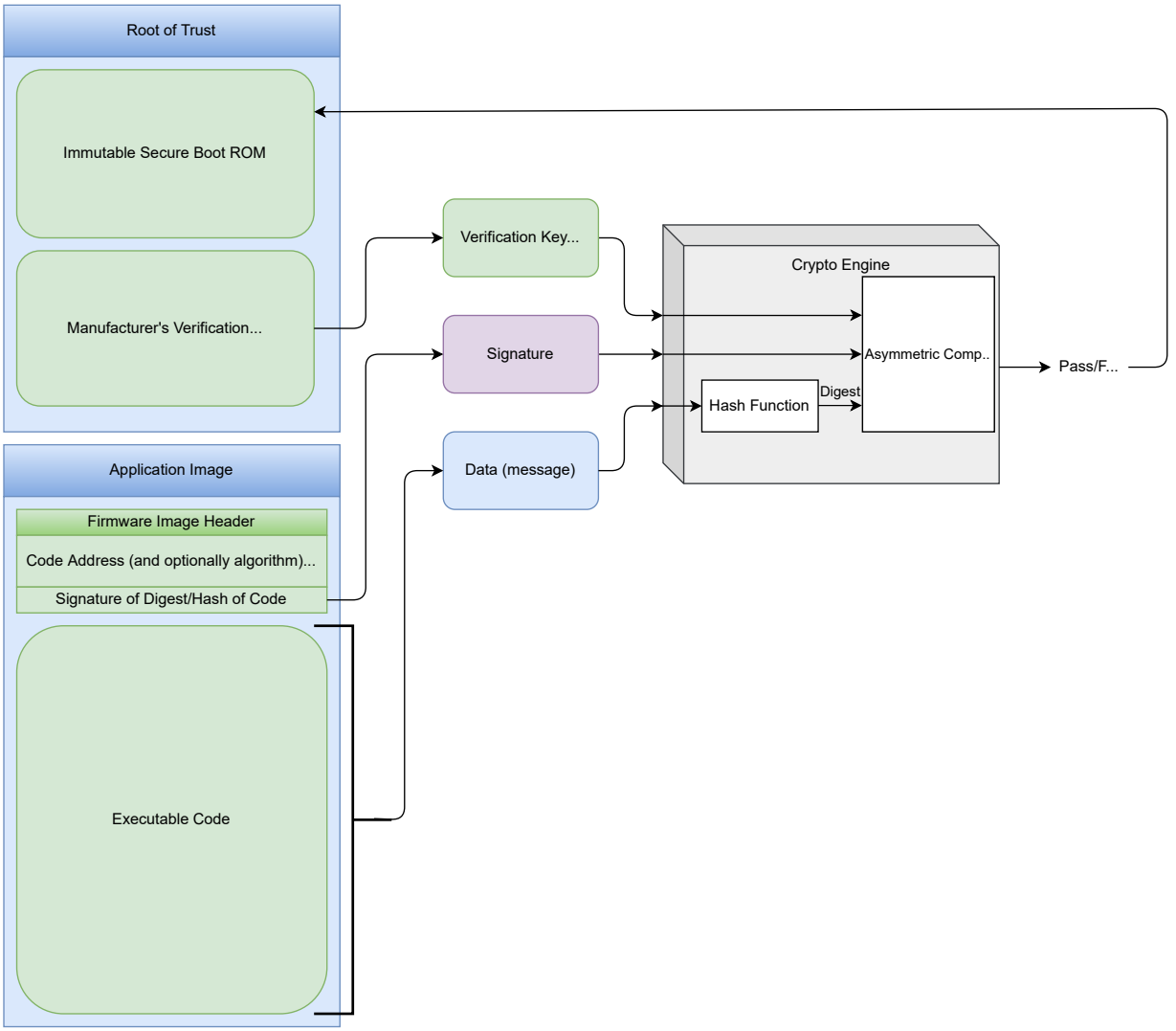
For an embedded Flash-only device, the ROM firmware checks for a valid image in three memory locations, see Table 12-2.

12.3.2.2 Firmware Image Validation – Secured Mode

The following are the main blocks in the process of image validation:

1. READ_CFG – Read device configuration (Device id, secure boot key, anti-rollback counter, life cycle counter). Secure elements and device id information can be accessed from eFuse memory.
   a. Device id – Device identifier for each device
   b. Secure boot key – Bootstrap code evaluates only images that use the secure boot key.
   c. Anti-rollback counter – This counter is used to keep track of the firmware version. The values are written by firmware in the eFuse region.
   d. Life cycle counter – This counter is used by firmware to track the changes in the life cycle of the device.
2. LOOKUP_METADATA – Search for valid firmware image metadata with the lowest sequence number.
3. AUTHENTICATE_IMAGE – Authenticate the image based on the security configuration of the device (eFuse) and as indicated by its metadata. The firmware image header dictates the authentication scheme. Secure boot is done on successful authentication of the firmware.
4. RUN_APPLICATION – If the valid image is in embedded flash and is available in its destination address space, jump to the DST_ADDR as indicated by its metadata. If the device does not find any valid image it jumps to Secure safe mode.
5. SECURE_SAFE_MODE – Secure Safe mode is a `while (1) loop` in the firmware, where the device is waiting for a valid application image to execute using the events like Power-on Reset or a programming tool to put the device into the image Lookup state to validate it.

Figure 12-2. Secure Boot Firmware Validation



12.3.2.3 Firmware Image Validation – Unsecured Mode

In the Unsecured mode, the device is not checked for the authentication scheme that needs SECURE_BOOT_KEY.

If the device is unsecured (and SECURE_BOOT_KEY is invalid and the Device Secured state is not set), the bootstrap code looks for valid images (valid sequence number, header and more). Root of trust cannot be possible in this scenario.

12.3.2.4 Firmware Loading

Firmware metadata can be stored in embedded Flash, RAMs or external storage. The firmware loading is done by the secondary boot loader. The firmware loading depends on the firmware metadata.

The secure boot code loads the primary firmware image into the SRAMs and authenticates it on the SRAM directly. After loading the firmware from the source, it relinquishes control of it.

12.3.3 Device Protection

Code Protect and Device secured state bits to determine the device security state and accessibility. The ROM firmware accessibility is defined based on the device state.

Table 12-3. Device Protection

	No Code Protect				Code Protect			
	Unsecured Device		Secured Device		Unsecured Device		Secured Device	
	Boot Key = 0	Boot Key != 0	Boot Key = 0	Boot Key != 0	Boot Key = 0	Boot Key ! = 0	Boot Key = 0	Boot Key != 0
Usage	Device with no security/ code protect	Debug of secured device	Device with no security, lock program interfaces	Secured device	Device with no security	No intended usage	Device with no security, lock program interfaces	Secured device/code
Firmware Authentication	No	Optional	No	Mandatory	No	Optional	No	Protected device
Debug Support	Open	Open	Secure debug entry required	Secure debug entry required	Secure debug entry required	Secure debug entry required	Secure debug entry required	Secure debug entry required
Serial Programming	All resources	All resources	None	Flash controller RoT	Chip erase only	Chip erase only	Chip erase only	Chip erase only
DFU	Present	Present	Locked	Present	Present	Present	Locked	Present
OTA	Present	Present	Present	Present	Present	Present	Present	Present

12.4 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0BFF	Reserved									
0x0C00	SEC_BOOT	7:0								
		15:8	BOOT_STATUS(2)[7:0]							
		23:16							SEC_BOOT_DONE(1)[1:0]	
		31:24								

12.5 Register Description

12.5.1 SEC_BOOT – Secure Boot Register

Name: SEC_BOOT
Offset: 0xC00
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SEC_BOOT_DONE(1)[1:0]	
Access							R/S	R/S
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	BOOT_STATUS(2)[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 17:16 – SEC_BOOT_DONE(1)[1:0] Bits to Indicate that Secure Boot is Done
FW can only set these bits. FW can never clear these bits.
These bits drive sec_boot_done output.

Value	Description
00	Secure boot is done
01	Secure boot is not done
10	Secure boot is not done
11	Secure boot is done

Bits 15:8 – BOOT_STATUS(2)[7:0] Firmware Managed Bits to Indicate Secure Boot Status
The 8-bit code is written to this field to indicate the secure boot status - like authentication success or failure or any other kind of indication. The code and its corresponding message is to be determined by the firmware. Refer to 12.6. Application Transition for the Boot Status firmware definition.
This field can be reset only on POR reset.
This field drives boot_status[7:0] output from the macro.

Notes:

1. These register bits are reset on **any** device reset.
2. These register bits are reset **only on** POR reset.

12.6 Application Transition

After loading and authenticating a valid image, the ROM firmware must perform several additional setup steps before transitioning to the application.

1. Remap interrupt and exception vectors to the application space addresses.
2. Set the SEC_BOOT_DONE configuration register bits.

3. Jump to application code at FW_IMG_DST_ADDR.

The ROM firmware programs the SECBOOTDONE bits to 0b11 when the firmware image is validated and before switching to the application. The BOOTSTATUS bits will be programmed with the firmware at each step and will indicate firmware status while trying to load the firmware.

Table 12-4. BOOT_STATUS[4:2] – Primary Status Codes

Code	Value	Description
DSU Mode		
TMODE_PE	0x02	System is in Parallel Execution Mode
TMOD_LOCK_DOWN	0x03	System is in Test Lock Down Mode; check for secondary status code.
Secure Boot		
SYS_FW_AUTH	0x04	System is currently authenticating the firmware; do nothing.
SYS_FW_LOAD	0x05	System is loading or programming the firmware image; do nothing.
SYS_APP_MODE	0x06	System is in application mode; a firmware image is loaded and is running.
SYS_LOCK_DOWN	0x07	System is Locked down; check secondary status.

Table 12-5. BOOT_STATUS[1:0] – Secondary Status Codes

Code	Value	Description
TMOD_LOCK_DOWN		
TMOD_8_10	0x00	System is in TMOD 8 or TMOD 10. Proceed with Factory reset.
ERR_BAD_TMOD	0x01	System does not support this test mode; reset the device.
SYS_LOCK_DOWN		
SYS_LD_ERR_FW_AUTH_FAIL	0x00	Reset the device and program a valid firmware image.
SYS_LD_ERR_NO_FW_IMG	0x01	Reset the device and program a valid firmware image.

13. eFuse Controller

13.1 Overview

The PIC32CX-BZ3 devices contain a programmable eFuse memory controller, which is used to store various parameters like a user programmable boot key, anti-rollback counter, life cycle counter and secure configuration details.

The eFuse controller is required to perform two modes of operation:

- Program Mode - Programming of eFuse panel
- Read Mode - Reading of eFuse panel

13.2 Hardware Mode

The Hardware (HW) mode is essentially using a state machine (FSM) within the eFuse Controller to perform the program and read operations. All control of the eFuse panel is done via the eFuse controller.

The user can program using the following registers/register bits:

- EFUSE_RWDATA.ADDR
- EFUSE_RWDATA.DATA
- EFUSE_CON.PGM_1BIT
- EFUSE_CON.EN_PGM
- EFUSE_CON.PGM_MODE

For fuse reading, the user can initiate fuse loading by the following registers/register bits:

- EFUSE_RWDATA.ADDR
- EFUSE_CON.EN_LD
- EFUSE_CON.EN_LD_ALL

By default, the eFuse controller starts in hardware mode so that it is automatically initialized. Read of the fuses can be performed after a device reset.

Note: To allow the use of the eFuse contents in determining the device configuration, the eFuse controller will automatically perform an initial read of the fuses after a POR (Power-On Reset) reset.

The EFUSE_RWDATA register is used to program data into the fuse panels, while the eFuse read operation leads to writing into System Registers.

13.3 eFuse Programming

In the eFuse programming operations, the eFuse controller can perform bit programming or byte programming determined by EFUS_CON.PGM_1BIT. In PIC32CX-BZ3, the programming voltage gets supplied by the eFuse LDO.

13.3.1 eFuse Programming Sequence

For eFuse programming operations, the eFuse controller will program 1-bit or 8 bits together at a time.

1. Write EFUSE_CON.PGM_MODE = 0.
2. Write EFUSE_CON.EN_LD = 0, and EFUSE_CON.EN_LD_ALL = 0.
3. Write EFUSE_CON.EN_OTP_LDO = 1. This enables the PMU OTP LDO output to 1.5V.
4. Write the EFUSE_RWDATA register with the offset address of eFuse and the data to be written.
   a. For single bit write operation: write eFuse offset address on EFUSE_RWDATA.ADDR[11:0] and data on EFUSE_RWDATA.DATA[0].

b. For 8-bit write operation: write eFuse offset address on EFUSE_RWDATA.ADDR[11:3] and data on EFUSE_RWDATA.DATA[7:0].
The above options are controlled by EFUSE_CON. PGM_1BIT register bit as in Step 5.2.

5. All the below steps to be done in a single step:
   a. Set EFUSE_CON.PGM_MODE register bit.
   b. Set EFUSE_CON. PGM_1BIT register bit for 1-bit write operation. Clear the bit for 8-bit write operation.
   c. Set EFUSE_CON.EN_PGM register bit.
6. The eFuse controller clears the EFUSE_CON.EN_PGM register bit when the eFuse programming is complete.
7. Repeat the steps 4 to 5 until all the values are programmed.
8. Write EFUSE_CON.EN_OTP_LDO = 0 to switch off PMU OTP LDO.

Note: Lifecycle counter and anti-rollback programming values to be programed in 1- bit operation mode.

13.3.2 eFuse Reading Sequence

The read from the eFuse panel happens at boot time and the holding registers are loaded with the eFuse values. Therefore, the memory map of the eFuse is in 1:1 correspondence with the memory map of the holding registers. If required to verify the values in eFuse, reset the device after the eFuse programming sequence is done. The modified eFuse values are read from the eFuse and loaded on holding registers at boot-up. Read 13.4.1. Holding Registers (similar to the process for the peripheral registers), then verify these registers.

13.4 eFuse Auto-Loading

In the time of POR, the eFuse controller loads the eFuse values from the eFuse panel into the holding registers.

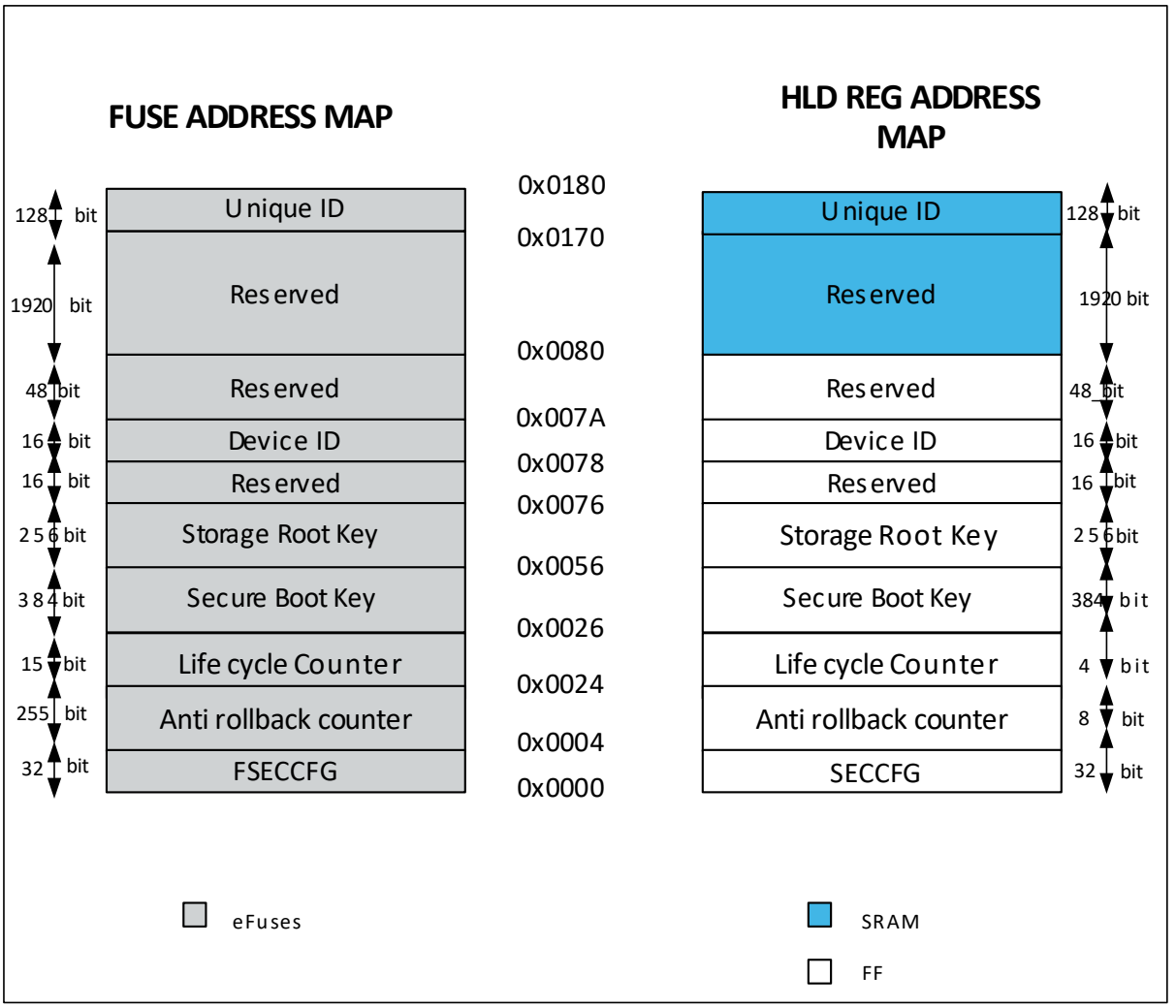
13.4.1 Holding Registers

The RoT Holding Registers store eFuse values that auto-loaded on a POR or when the EN_LD_ALL/ EN_LD register bits are set to perform a manual load operation.

The eFuse blocks are used to implement the following holding registers:

- Unique ID
- 16-bit device ID
- 256-bit Storage Root Key (SRK)
- 384-bit public boot key
- 15-bit lifecycle counter
- 256-bit anti-rollback counter
- Security/RoT configuration

Figure 13-1. Memory Map of the Fuse Panels



The holding registers are governed by their read and write access permissions. The holding registers are loaded from their corresponding fuses if any of the following conditions are met:

- After a POR during the auto-load state
- Explicit FW setting of EN_LD_ALL or EN_LD bits
- After fuse programming if EN_LD_ALL or EN_LD bits are set in the same cycle as EN_PGM being set

The above scheme minimizes the power and boot time associated with an eFuse load operation.

Holding registers can also be written by firmware dictated by the write locks as specified in the following table.

Table 13-1. Device Elements Provisioning

Secure Element	Size in Bits	Provisioner	To Program Lock
Unique ID	128	MCHP	Yes
Device ID	16	MCHP	Yes
SECCFG	32	Code signer (MCHP, 3rd Parties, Customers)	No
Anti Rollback Counter	255	Code signer (MCHP, 3rd Parties, Customers)	No

.....continued			
Secure Element	Size in Bits	Provisioner	To Program Lock
Life Cycle Counter	15	Code signer (MCHP, 3rd Parties, Customers)	No
Secure Boot Key	384	Code signer (MCHP, 3rd Parties, Customers)	Yes
Storage Root Key	256	MCHP	Yes

All the holding registers are reset by POR. All the regular SFRs in the macro are reset by a system reset signal unless specifically mentioned otherwise.

13.5 Security Keys

13.5.1 Secure Boot Key

Secure boot key is the public key part of the private/public key pair that is used for code authentication. It can be 256 bits or 384 bits depends on the authentication algorithm.

13.5.2 Storage Root Key

The storage root key is also called a local storage key. This is a 256-bit (symmetric crypto) secret encryption key used to store important data secretly. It is a programmable key that is randomly generated for each device in the factory.

13.6 Counters

There are two thermometer counters in this PIC32CX-BZ3 device as follows:

- Lifecycle counter
- Anti-rollback counter

13.6.1 Life Cycle Counter

This is a 15-bit thermometer counter that is decoded into a 4-bit value. This counter is used by firmware to track the changes in the lifecycle of the device. These fuses are programmable by firmware. The holding registers for this counter hold the decoded value from the fuses.

The possible states are as follows:

- BLANK
- FACTORY
- CUSTOMER

An example of decoding the thermometer counter can be seen in Figure 13-2.

13.6.2 Anti-Rollback Counter

This is a 255-bit thermometer counter that is decoded into an 8-bit value. This counter is used to keep track of the firmware version. These fuses are programmable by firmware. The holding registers for this counter hold the decoded value from the fuses. The holding register is not writable and is only readable by firmware.

The firmware image major revision number is greater than or equal to the value of the anti-rollback counter. Also, the counter is used to indicate the minimum acceptable firmware revision.

The following figure provides an example for the decode of the thermometer counter.

Figure 13-2. 4-bit Thermometer Counter

Counter	Decode
0000	0
0001	1
0011	2
0111	3
1111	4
0010	2
0100	3
0101	3
0110	3
1000	4
1001	4
1010	4
1110	4
1011	4
1100	4
1101	4
1110	4

Notes:

- Rows in green indicate the correct flow of the increment steps of the counters.
- However, if the user programs an incorrect value to the counter, the decode results are shown in rows of orange.

13.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0C03	Reserved									
0x0C04	EFUSE_RWDATA	7:0	DATA[7:0]							
		15:8								
		23:16	ADDR[7:0]							
		31:24					ADDR[11:8]			
0x0C08	EFUSE_CON	7:0	EN_PGM	EN_LD_ALL	EN_LD				PGM_MODE	PGM_1BIT
		15:8								
		23:16								EN_OTP_LDO
		31:24								

13.8 Register Description

13.8.1 EFUSE_CON – eFuse Configuration Register

Name: EFUSE_CON
Offset: 0xC08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								EN_OTP_LDO
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
	EN_PGM	EN_LD_ALL	EN_LD				PGM_MODE	PGM_1BIT
Access	R/W/HC	R/W/HC	R/W/HC				R/W	R/W
Reset	0	0	0				0	0

Bit 16 – EN_OTP_LDO Enable OTP LDO

The separate OTP LDO gets automatically enabled for programming operations. If longer settling times are desired or many values need to be programmed consecutively, the OTP LDO can be enabled manually.

Note: While the OTP LDO is enabled manually, it is not possible to start any loading operation. To load newly programmed values to the holding registers, EN_OTP_LDO needs to be set to 0 first, before starting the load using EN_LD_ALL/EN_LD.

Value	Description
1	Enable OTP LDO
0	Disable OTP LDO

Bit 7 – EN_PGM eFuse Programming Start Bit

Notes:

- This bit has no effect when PGM_MODE = 0.
- This bit will be automatically cleared by hardware when the programming operation is complete.
- When the EN_PGM bit is set, the EFUSE_CON & EFUSE_RWDATA registers may not be changed until the EN_PGM is cleared.
- EN_PGM, PGM_MODE and PGM_1BIT can all be written in the same cycle.

Value	Description
1	Start eFuse Programming operation
0	eFuse Programming operation has completed

Bit 6 – EN_LD_ALL eFuse Panel Read Start Bit for Loading into the Holding Registers

Notes:

- This bit will be automatically cleared by hardware when the read operation is complete.
- This bit resets to '1', so that an initial auto-load of the fuse values is performed after a device reset. This allows the value of the fuse to be used in determining functionality like device pinout, memory configurations, etc. When this auto-load is completed, the hardware clears this register bit.
- The user can set this bit after programming the fuse for the fuses to be loaded into the holding register. After this bit is cleared, the user can read the fuses (holding registers) to verify the programmed value.
- When the EN_LD_ALL bit is set, the EFUSE_CON & EFUSE_RWDATA registers cannot be changed until EN_LD_ALL is cleared.

Value	Description
1	Start eFuse Read operation for entire eFuse panel
0	eFuse Read operation has completed

Bit 5 – EN_LD eFuse Word Read Start Bit for Loading the Fuse Byte Pointed by ADDR Field into the Holding Register

Notes:

- This bit will be automatically cleared by hardware when the read operation is complete.
- When

  the EN_LD bit is set, the EFUSE_CON & EFUSE_RWDATA registers cannot be changed until EN_LD is cleared.

Value	Description
1	Start eFuse Read operation for specified eFuse word as addressed in EFUSE_RWDATA register
0	eFuse Read operation has completed

Bit 1 – PGM_MODE eFuse Programming Mode Enable Bit

Note: EN_PGM, PGM_MODE and PGM_1BIT can all be written in the same cycle.

Value	Description
1	eFuse Programming enabled
0	eFuse Programming disabled

Bit 0 – PGM_1BIT eFuse CTRL to Program 1 Bit at a Time. Valid only when EN_PGM is Set

Note: EN_PGM, PGM_MODE and PGM_1BIT can all be written in the same cycle.

Value	Description
1	eFuse controller will program EFUSE_RWDATA.DATA[0] to the address in EFUSE_RWDATA.ADDR[11:0]
0	eFuse controller will program EFUSE_RWDATA.DATA[7:0] to the address in EFUSE_RWDATA.ADDR[11:3]

Note:

1. EN_PGM, PGM_MODE, PGM_1BIT, EN_LD_ALL, EN_LD - All the bits can be written in the same cycle (as per requirement). Combinational setting of configuration bits for programming a read operation is not allowed when OTP LDO is enabled manually with EN_OTP_LDO.

13.8.2 EFUSE_RWDATA – eFuse Read/Write Data Register

Name: EFUSE_RWDATA
Offset: 0xC04
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
					ADDR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[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	15	14	13	12	11	10	9	8
Access								
Reset								

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 27:16 – ADDR[11:0] eFuse Program/Read Address
When EN_PGM = 1 & PGM_1bit = 0, **ADDR[11:3]** is used as eFuse Program Address.
When EN_PGM = 1 & PGM_1bit = 1, **ADDR[11:0]** is used as eFuse Program Address.
When EN_LD_ALL = 0 & EN_LD = 1, **ADDR[11:3]** is used as eFuse Sense to Read Address.
When EN_LD_ALL = 1, **ADDR[11:0]** is not used.

Bits 7:0 – DATA[7:0] eFuse Program (Write) Data
When EN_PGM = 1 & PGM_1bit = 0, **DATA[7:0]** is used as eFuse Program Data.
When EN_PGM = 1 & PGM_1bit = 1, **DATA[0]** is used as eFuse Program Data.
When EN_PGM = 0, **DATA[7:0]** is not used.

14. Security Features

14.1 Overview

This device includes a set of components that can support a high level of system security to the device.

14.2 Features

The key features of the module are:

- High-Performance Cryptographic Accelerators
- Authentication, Using Public Key Algorithms
- Integrity, Using Secure Hash Algorithms
- Privacy, Using Symmetric Encryption (AES)
- Entropy, Using a True Random Number Generator
- Public Key, RSA Encryption, and Decryption with Key Sizes of 1024 Bits, 2048 Bits, 3072 Bits and 4096 Bits
- ECC, Crypto Element with Protected Hardware-Based Key Storage
- Passing the Storage Root Key from the Root of Trust Macro into the Security Module
- Integrated Scatter-Gather DMA
- AHB Host and Client Interfaces

14.3 Reference Documentation

- American National Standards Institute, "*Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*", X9.63-2011, December 2011
- American National Standards Institute, "*Public Key Cryptography for the Financial Servic3es Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*", X9.62-2005, November 2005
- International Standards Organization, "*Information Technology - Security techniques - Cryptographic techniques based on elliptic curves -- Part 2: Digital Signatures*", ISO/IEC 15946-2, December 2002
- National Institute of Standards and Technology, "*Secure Hash Standard (SHS)*", FIPS Pub 180-4, March 2012
- National Institute of Standards and Technology, "*Digital Signature Standard (DSS)*", FIPS Pub 186-3, June 2009
- National Institute of Standards and Technology, "*Advanced Encryption Standard (AES)*", FIPS Pub 197, November 2001
- National Institute of Standards and Technology, "*Recommendation for Block Cipher Modes of Operation*", FIPS SP 800-38A, 2001
- RSA Laboratories, "*PKCS#1 v2.2: RSA Cryptography Standard*", October 2012

14.4 Interface

This block is designed to be accessed internally via a registered host interface.

Figure 14-1. I/O Block Diagram

14.5 Description

The security hardware incorporates the following functions:

14.5.1 Symmetric Encryption/Decryption

Standard AES encryption and decryption with key sizes of 128 bits, 192 bits and 256 bits are supported with a hardware accelerator. AES modes that can be configured include Electronic Code Block (ECB).

14.5.2 Cryptographic Hashing

Standard SHA hash algorithms, including SHA-256 and SHA-384, are supported by hardware.

14.5.3 Public Key Cryptographic Engine

A large variety of public key algorithms are supported directly in hardware. These include:

- RSA encryption and decryption with key sizes of 1024 bits and 2048 bits
- Elliptic Curve point multiply with all standard NIST curves using either binary fields or prime fields
- Elliptic Curve point multiply with Curve25519
- The Elliptic Curve Digital Signature Algorithm (ECDSA) using all supported NIST curves
- The Elliptic Curve Korean Certificate-based Digital Signature Algorithm (EC-KCDSA) using all supported NIST curves
- The Edwards-curve Digital Signature Algorithm (EdDSA) using Curve25519

14.5.4 True Random Number Generator

A true Random Number Generator includes a module provided for the pre-calculation of random bits. This block has a Health Check function included with it.

14.5.5 Cryptographic API

The Boot ROM includes an API for direct software access to cryptographic functions. API functions for Hashing and AES include a DMA interface so the operations can function on large blocks of SRAM with a single call.

14.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CRYPTOCON	7:0		RUNSTDBY					ENABLE	SWRST
		15:8								
		23:16								
		31:24								

14.7 Register Description

There are no registers directly accessible to the application in this block. Users must use the APIs to use this block. For the list of APIs, see *Secure Boot ROM* from Related Links. Apart from the crypto module, enable and disable the register.

Related Links

12. Secure Boot ROM

14.7.1 CRYPTOCON – Crypto Control Register

Name: CRYPTOCON
Offset: 0x00
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
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby bit

Value	Description
1	If enabled, the clock to security module remains enabled in Idle mode
0	Security module clock disabled in Idle mode

Bit 1 – ENABLE Module Enable Bit

Value	Description
1	Crypto module is enabled. Security module clock is enabled
0	Crypto module is disabled. Security module clock is disabled

Bit 0 – SWRST Software Reset

Write '1' to reset registers and internal state. Writing '0' has no effect. SWRST stays high until the reset completes.

Value	Description
1	Reset registers and internal state
0	No effect

15. Flash Controller (FC)

15.1 Overview

The PIC32CX-BZ3 devices contain a single bank of Flash memory with their Program Flash Memory (PFM) partition and Boot Flash Memory (BFM) partition for storing user code or non-volatile data. The user can use the Flash controller to access the Flash memory. For commands and configuration of the Flash controller, use the peripheral bus interface.

15.2 Features

Flash Controller

- PB-Bridge-D Interface that Provides Access to the Flash Controller Registers
- AHB Initiator for Bus Hosted Reads the Row Programming Data from SRAM
- Write Protect for Program Flash (PFM)
  - Single page protection resolution
  - Protect < Address
  - Protect ≥ Address
- Individual Page Write Protection for Boot Flash (BFM)
- Error Correction Code Support
- Supports Chip and Page Erase
- Supports Single Word, Quad Word and Row Program Options
- Supports Fash Erase/Retry to Increase Retention and Endurance

Flash Memory

- 128-Bit Wide Flash Memory Access
- 4 Kbytes Page Size
- Row Size is 1 KB (256 IW)
- Flash-Based One-Time-Programmable (OTP) Page

The Flash controller allows the Flash memory to be accessed through the following methods:

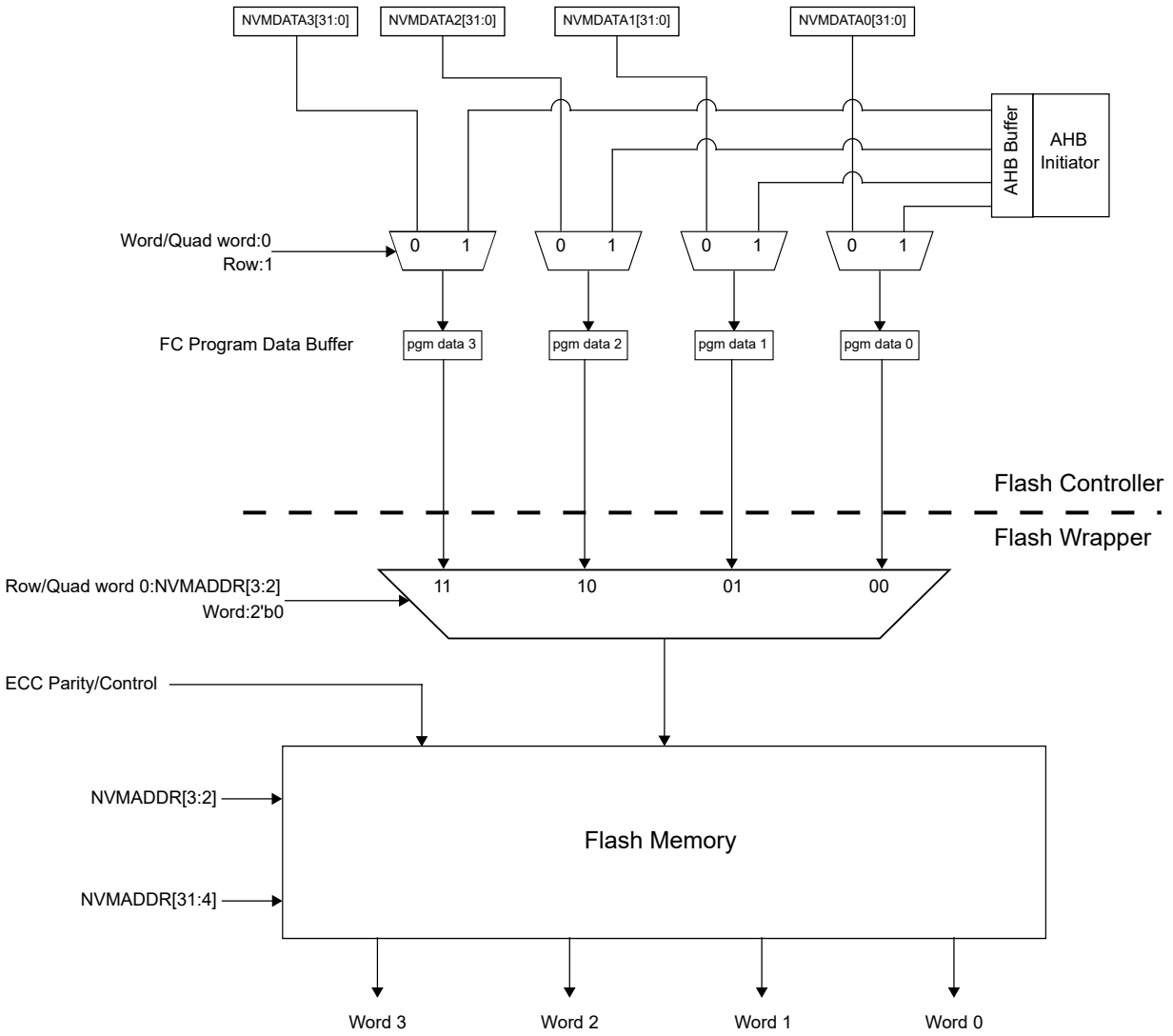
1. Run-Time Self-Programming (RTSP)
2. Serial Wire Debug (SWD) programming using DSU (See *Device Service Unit (DSU)* from Related Links and *PIC32CX-BZ3 Programming Specification*.)

Related Links

16. Device Service Unit (DSU)

15.3 Functional Block Diagram

Figure 15-1. Flash Memory Block Diagram



15.4 Product Dependencies

Not applicable.

15.4.1 I/O Lines

Not applicable.

15.4.2 Power Management

The Flash Controller does not operate in Power-saving/Low-power/Sleep modes. If a WAIT instruction is encountered when programming, the CPU stops execution (stall), waits for the programming operation to complete, then enters the Power-saving/Low-power mode.

15.4.3 Clocks

AHB (SYS_CLK) initiator for bus manager reads of row programming data from system FlexRAM. Use the PB1_CLK bus clock for control register access.

15.4.4 DMA

Not applicable.

15.4.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the Flash controller interrupt(s) requires the NVIC interrupt controller to be configured first.

15.4.6 Events

Not applicable.

15.4.7 Debug Operation

Programming operations continues to completion if the processor execution is halted in Debug mode.

15.4.8 Register Access Protection

Not applicable.

15.4.9 Analog Connections

Not applicable.

15.5 Flash Memory Addressing

Flash memory addressing uses physical addresses only. For more information on addressing, see *Product Memory Mapping Overview* from Related Links.

Related Links

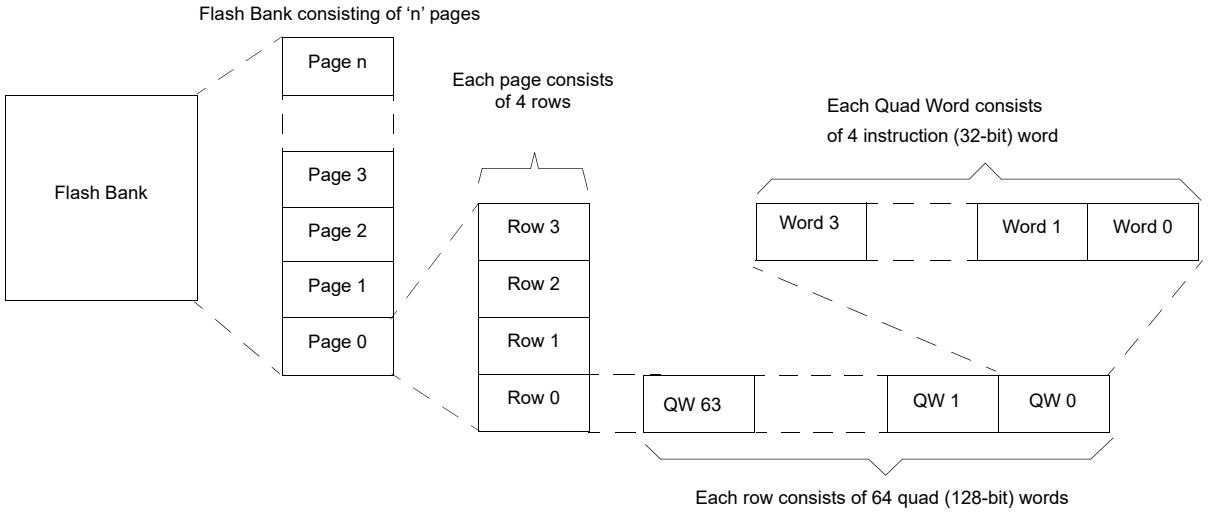
8. Product Memory Mapping Overview

15.6 Memory Configuration

15.6.1 Flash Memory Construction

Flash memory is divided into pages. A page is the smallest unit of memory that can be erased at one time. Each page of memory is segmented into four rows. A row is the largest unit of memory that can be programmed at one time. A row consists of 64 Quad (128-bit) Word. Each Quad Word consists of a four instruction (32-bit) Word. Flash memory can be programmed in rows, Quad Word (128-bit) or Single Word (32-bit) units.

Figure 15-2. Flash Construction



15.6.2 Flash Memory Organization

The device Flash memory is divided into two logical Flash partitions:

1. Main Program Flash Memory (PFM)
2. Boot/Configuration Flash Memory (BFM)
   a. Boot Flash
   b. Device/Boot Configuration – Device and boot configuration data
   c. OTP – User system calibration data

The following table provides details about each Flash section having different protection status.

Table 15-1. Protection Status

Flash Partition	Memory Region	Write Protection	Erase Protection	Chip Erase through DSU
BFM	Boot Flash	Yes. Page-wise Configurable	Yes. Page-wise Configurable	Erased
	Device/Boot Configuration	Yes. Configurable	Yes. Configurable	Erased
	One Time Programmable (OTP)	Yes. Configurable	Always Erase protected. Can not be erased	Not Erased
PFM	Program Flash	Yes. Configurable	Yes. Configurable	Erased

15.7 Boot Flash Memory (BFM) Partitions

15.7.1 BFM Write Protection

Pages in the BFM regions can be protected individually using bits in the NVMLBWP register. At Reset, all pages are in a write-protected state and must be disabled prior to performing any programming operations on the BFM regions. There is also an unlock bit, ULOCK(NVMLBWP[31]), that is set at Reset and can be cleared by the user software. When cleared, changes to write protection for that region can no longer be made. When cleared, the ULOCK bit can only be set by a Reset.

The user can only change the NVMLBWP write-protect register by following the unlock sequence. See *NVMKEY Register Unlocking Sequence* from Related Links.

Related Links

15.12. NVMKEY Register Unlocking Sequence

15.8 Program Flash Memory (PFM) Partitions

15.8.1 PFM Write Protection

Write protection for the PFM region is implemented by pages and defined by the NVMPWPLT and NVMPWPGTE registers. The NVMPWP* registers define an area within the program space (PFM) that is write-protected. This write-protected address resolves to Flash page boundaries; therefore, the user can ignore the 12 LSBs for a 4 KB page Flash of any address written to the NVMPWP* registers. The size of the Flash determines the width of each NVMPWP* address register. Use the NVMPWPLT register to set the Program Flash pages lower than the provided address as write-protected. Use the NVMPWPLT register to set the Program Flash pages greater than or equal to the provided address as write-protected. Therefore, a value of all `0`s in the NVMPWPLT register and all `1`s in the NVMPWPGTE register results in no region of Flash being write-protected (default state at Reset).

There is also an unlock bit, ULOCK (NVMPWPLT [31] and NVMPWPGTE[31]), that is set at Reset and can be cleared by the user software. After clearing, the user can no longer make changes to the write-protection of the PFM, including the ULOCK bit. The NVMPWPLT and NVMPWPGTE

write-protected register can only be changed by following the unlock sequence. See *NVMKEY Register Unlocking Sequence* from Related Links.

Related Links

15.12. NVMKEY Register Unlocking Sequence

15.9 Error Correcting Code (ECC) and Flash Programming

The PIC32CX-BZ3 devices incorporate Error Correcting Code (ECC) features that detect and correct errors resulting in extended Flash memory life. For more details on this feature, see *Prefetch Cache* from Related Links.

ECC is implemented in 128-bit Quad Flash Words or 32-bit single Word. As a result, when programming Flash memory on a device with ECC, the programming operation must be, at minimum, four instruction Words or in groups of four instruction Words. This is the reason that the Quad Word programming command exists and why row programming always programs multiples of four Words.

For a given software application, ECC can be enabled at all times, disabled at all times or dynamically enabled using the ECCCTL Configuration bits in the CFGCON0 register. When ECC is enabled at all times, the single Word NVMOP programming command does not function and the Quad Word is the smallest unit of memory that can be programmed. If disabling or enabling the ECC dynamically, both the single Word and Quad Word programming NVMOP commands are functional and the programming method used determines how ECC is handled.

If dynamic ECC and if the memory was programmed with the single Word command, ECC is turned-off for that Word, and, when it is read, no error correction is performed. If the memory was programmed with the Quad Word or row programming commands, ECC data is written and tested for errors (and corrected if needed) when read. The following table describes the different ECC scenarios.

Table 15-2. ECC Programming Summary

ECCCTL Setting	Programming Operation			Data Read
	Single Word Write	Quad Word Write	Row Write	
Disabled	Allowed	Allowed	Allowed	ECC is never applied on a Flash read
Enabled	Not allowed	Allowed	Allowed	ECC is applied on every Flash Word read
Dynamic	Allowed but when used, the programmed Word is flagged to NOT USE ECC	Writes ECC data and flags programmed Words to USE ECC	Writes ECC data and flags programmed Words to USE ECC	ECC is only applied on Words that are flagged to USE ECC
Note: When using dynamic ECC, the user must program all the non-ECC locations with the 32-bit Word programming command and program all the ECC-enabled locations with a 128-bit Quad Word or row programming command. Divisions between ECC and non-ECC memory must be on even Quad Word boundaries (address bits 0 through 3 are equal to '0').				

Related Links

10. Prefetch Cache (PCHE)

15.10 Interrupts

An interrupt is generated when the Flash controller clears the WR bit upon completion of a Flash program or erase operation. The interrupt event causes a CPU interrupt if it was configured and enabled in the NVIC. For the vector mapping table, see *Nested Vector Interrupt Controller (NVIC)* from Related Links. The interrupt occurs regardless of the outcome of the program or erase operation, successful or unsuccessful. The only exception is the No Operation (NOP) programming operation (NVMOP = 0), which is used to manually clear the error flags and does not create an interrupt event on completion but does clear the WR bit.

The Flash Controller interrupts are not persistent, and, therefore, there is no need for an additional step to clear the cause or source of the interrupt.

After configuring the interrupt controller, the Flash event causes the CPU to jump to the vector assigned to the Flash event. The CPU starts executing the code at the vector address. The user software at this vector address must perform the required operations and, then, exit.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

15.10.1 Interrupts and CPU Stalling

The CPU cannot fetch the code from the same Flash bank, either BFM or PFM, that is the target of the programming operation. When attempting this operation, the CPU ceases to execute code (stall) while the programming operation is in progress. CPU code execution does not resume until the programming operation is complete, and, when this occurs, any pending interrupts, including those from the Flash Controller, are processed in order of priority.

Note: Code that is already loaded into the processor cache continues to execute up to the point where an attempt is made to fetch code or data from the same Flash bank as the active programming operation. At this point the CPU stops.

The user can avoid the stalling of the CPU by placing any needed executable code in SRAM during Flash programming.

15.11 Error Detection

The NVMCON register includes two bits for detecting error conditions during a program or erase operation. They are Low-Voltage Detect Error, LVDERR bit (NVMCON[12]), and Write Error, WRERR bit (NVMCON[13]).

The WRERR is set each time the WR bit (NVMCON[15]) is set, initiating a programming operation. When the Flash operation is complete, indicated by hardware clearing the value of the WR bit (WR bit is set to '0'), hardware updates the value in the WRERR bit to indicate if an error occurred. Firmware must check the value of the WR bit to see if the Flash operation completed before checking the value of the WRERR bit. When the WRERR bit is set, any future attempt to initiate programming or erase operation is ignored. WRERR must be cleared before commencing Flash program or erase operations.

The LVDERR bit is set when a Brown-out Reset (BOR) occurs during a programming operation. The only Reset that clears the LVDERR bit is a Power-on Reset (POR). Other Reset types do not affect the LVDERR bit. When the LVDERR bit is set, any attempt to initiate programming or erase operation is ignored. Clear the bit before commencing Flash program or erase operations.

In the software, manually clear both the WRERR and LVDERR bits by initiating a Flash operation (setting WR) referred to as NOP (0x00) (see the NVMOP bit fields).

Note: Executing the NVMOP NOP command clears WRERR, LVDERR and WR bits but does not generate an interrupt event on completion.

Table 15-3. Programming Error Cause and Effects

Cause of Error	Effect on Programming Erase Operation	Indication
A low-voltage event occurred during a programming sequence.	The last programming or erase operation might not have completed.	LVDERR = 1, WRERR = 1
A non-POR Reset occurred during programming.	Programming or erase operation is aborted.	WRERR = 1
Attempt to program or erase a page out of the Flash memory range.	Erase or programming operation is not initiated.	WRERR = 1

.....continued		
Cause of Error	**Effect on Programming Erase Operation**	**Indication**
Attempt to erase or program a write-protected PFM page.	Erase or programming operation is not initiated.	WRERR = 1
Attempt to erase or program a write-protected BFM page.	Operation occurs, but the page is not programmed or erased.	WRERR = 0
Bus host error or row programming data underrun error during programming.	Programming or erase operation is aborted.	WRERR = 1

15.12 NVMKEY Register Unlocking Sequence

Important register settings that can compromise the Flash memory if inadvertently changed are protected by a register-unlocking sequence. The user can implement this feature using the NVMKEY register. The NVMKEY register is a write-only register that is used to implement an unlock sequence to help prevent accidental writes or erasures of the Flash memory.

In some instances, the operation is also dependent on the setting of the WREN bit (NVMCON[14]) (see the following table).

Table 15-4. NVMKEY Register Unlocking and WREN

Operation	WREN Setting	Unlock Sequence Required
Changing value of NVMOP[3:0] (NVMCON[3:0])	0	No
Setting WR (NVMCON[15]) to start a write or erase operation	1	Yes
Changing any fields in the NVMPWP* register	—	Yes
Changing any fields in the NVMLBWP register	—	Yes

The user must follow the following steps in the exact order as shown to enable writes to registers that require this unlock sequence:

1. Write 0x00000000 to NVMKEY.
2. Write 0xAA996655 to NVMKEY.
3. Write 0x556699AA to NVMKEY.
4. Write the value to the register NVMCON, NVMCON2, NVMPWP* or NVMLBWP requiring the unlock sequence.

When using the unlock sequence to set or clear bits in the NVMCON register (see step 4). The user must execute steps 2 through 4 without any other activity on the peripheral bus that is in use by the Flash Controller. Disable the interrupts and DMA transfers that access the same peripheral bus as the Flash Controller. In addition, the operation in step 4 must be atomic. Use the Set, Clear and Invert registers, where applicable, for the target register in step 4.

The following code shows code written in the C language to initiate an NVM Operation (NVMOP) command. In this particular example, the WR bit is being set in the NVMCON register and, therefore, must include the unlock sequence.

Initiate NVM Operation (System Unlock Sequence Example):

```
void NVMInitiateOperation(void)
{
  // Disable Interrupts
  asm volatile("di%0" : "=r"(int_status));
  uint32_t globalInterruptState= __get_PRIMASK();
// Disable Interrupts
 __disable_irq();
```

```
  NVMKEY = 0x0;
  NVMKEY = 0xAA996655;
  NVMKEY = 0x556699AA;
  NVMCONSET = 1 << 15;// must be an atomic instruction

  // Restore Interrupts
  __set_PRIMASK(globalInterruptState);
}
```

Note: After writing the unlock codes to the NVMKEY register, the next activity on the same peripheral bus as the Flash Controller resets the lock. As a result, the user can only use the atomic operations. Use of the NVMCONSET register sets the WR bit in a single instruction without changing other bits in the register. Using NVMCONbits.WR = 1 fails as this line of code compiles to a read-modify-write sequence.

15.13 Word Programming

The smallest block of data that can be programmed in a single operation is one Flash write Word (32-bit). The data to be programmed must be written to the NVMDATA0 register, and the address of the Word must be loaded into the NVMADDR register before initiating the programming sequence. The instruction Word at the physical location pointed to by the NVMADDR register is, then, programmed. Programming occurs on 32-bit Word boundaries; therefore, the user can ignore the bits '0' and '1' of the NVMADDR register.

If programming a Word, erase it before programming again, even if changing a bit from an erased '1' state to a '0' state.

Word programming only succeeds if the target address is in a page that is not write-protected. Programming to a write-protected PFM page fails and results in the WRERR bit being set in the NVMCON register. Programming a write-protected BFM page fails but does not set the WRERR bit.

A programming sequence consists of the following steps:

1. Write 32-bit data to be programmed to the NVMDATA0 register.
2. Load the NVMADDR register with the address to be programmed.
3. Set the WREN bit = 1 and NVMOP bits = 1 in the NVMCON register. This defines and enables the programming operation.
4. Initiate the programming operation. (See *NVMKEY Register Unlocking Sequence* from Related Links.)
5. Monitor the WR bit of the NVMCON register to flag completion of the operation.
6. Clear the WREN bit in the NVMCON register.
7. Check for errors and process accordingly.

The following code shows code for Word programming, where a value of 0x12345678 is programmed into location 0x1008000.

Word Programming Code Example:

```
…
// Set up Address and Data Registers
  NVMADDR= 0x1008000;           // physical address
  NVMDATA0 = 0x12345678;         // value

// set the operation, assumes WREN = 0
  NVMCONbits.NVMOP = 0x1;          // NVMOP for Word programming

// Enable Flash for write operation and set the NVMOP
  NVMCONbits.WREN = 1;

// Start programming
  NVMInitiateOperation();                // see Initiate NVM Operation (Unlock Sequence
Example)

// Wait for WR bit to clear
```

```
  while (NVMCONbits.WR);

// Disable future Flash Write/Erase operations
  NVMCONbits.WREN = 0;

// Check Error Status
  if(NVMCON & 0x3000)            // mask for WRERR and LVDERR
  {
        // process errors
    }
…
```

Related Links

15.12. NVMKEY Register Unlocking Sequence

15.14 Quad Word Programming

The process for Quad Word programming is identical to Word programming except that all four of the NVMDATAx registers are used. The value of the NVMDATA0 register is programmed at address NVMADDR, NVMDATA1 at NVMADDR + 0x4, NVMDATA2 at NVMADDR + 0x8, and NVMDATA3 at address NVMDATA + 0xC.

Always perform the Quad Word programming on a Quad Word boundary so the user can ignore the NVMADDR address bits 3 through 0.

Quad Word programming only succeeds, if the target address is in a page that is not write-protected. If programming a Quad Word, erase it before programming any word in it again, even if changing a bit from an erased '1' state to a '0' state.

A value of 0x11111111 is programmed into location 0x1008000, 0x22222222 into 0x1008004, 0x33333333 into 0x1008008, and 0x44444444 into location 0x100800C.

The following example code provides details about the Quad Word programming.

Quad Word Programming Code Example:

```
…

// Set up Address and Data Registers
    NVMADDR  = 0x1008000;      // physical address
    NVMDATA0 = 0x11111111;     // value written to 0x1008000
    NVMDATA1 = 0x22222222;     // value written to 0x1008004
    NVMDATA2 = 0x33333333;     // value written to 0x1008008
    NVMDATA3 = 0x44444444;     // value written to 0x100800C

// set the operation, assumes WREN = 0
    NVMCONbits.NVMOP = 0x2;    // NVMOP for Quad Word programming

// Enable Flash for write operation and set the NVMOP
    NVMCONbits.WREN = 1;

// Start programming
    NVMInitiateOperation();    // see Initiate NVM Operation (Unlock Sequence Example)

// Wait for WR bit to clear
    while(NVMCON & NVMCON_WR);

// Disable future Flash Write/Erase operations
    NVMCONbits.WREN = 0;

// Check Error Status
    if(NVMCON & 0x3000)    // mask for WRERR and LVDERR bits
```

15.15 Row Programming

The largest block of data that can be programmed is a row.

Unlike Word and Quad Word Programming where the SFR memory stores the data source, in Row programming, the SRAM stores the source data. The NVMSRCADDR register is a pointer to the physical location of the source data for Row programming.

Like other Non-Volatile Memory (NVM) programming commands, the NVMADDR register points to the target address of the operation. Row programming always occurs on row boundaries with the row size of 1024. The user can ignore the bits 0 through 9 of the NVMADDR register.

Row Word programming only succeeds if the target address is in a page that is not write-protected. If row programming, erase it before programming any word in it again, even if changing a bit from an erased '1' state to a '0' state.

Array `rowbuff` is populated with data and programmed into a row located at physical address `0x1008000`.

Notes:

- When assigning the value to the NVMSRCADDR register, convert it to a physical address.
- Concurrent access from the Wireless Subsystem and Flash controller using row programming (NVM_RowWrite API) can cause corrupt data. While using row programming, the Wireless Subsystem must be Inactive. Otherwise, use the quad-word programming (NVM_QuadWordWrite API) method.

Row Programming Code Example:

```
…

unsigned long rowbuff[256]; // example is for a 256 Word row size.
int x;                          // loop counter

// put some data in the source buffer
      for (x = 0; x < (sizeof(rowbuff) * sizeof (int)); x++)
            ((char *)rowbuff)[x] = x;

// set destination row address
     NVMADDR = 0x1008000;             // row physical address

// set source address. Must be converted to a physical address.
     NVMSRCADDR = (unsigned int)((int)rowbuff & 0x1FFFFF);

// define Flash operation
      NVMCONbits.NVMOP = 0x3;            // NVMOP for Row programming

// Enable Flash Write
      NVMCONbits.WREN = 1;

// commence programming
      NVMInitiateOperation();            // see Initiate NVM Operation (Unlock Sequence
Example)

// Wait for WR bit to clear
      while(NVMCONbits.WR);

// Disable future Flash Write/Erase operations
       NVMCONbits.WREN = 0;

// Check Error Status
       if(NVMCON & 0x3000)            // mask for WRERR and LVDERR bits
      {
         // process errors
       }
…
```

15.16 Page Erase

A Page Erase performs an erase of a single page of either PFM or BFM.

The page to be erased is selected using the NVMADDR register. Pages are always erased on page boundaries; therefore, for a device with an instruction Word page size of 4096, the user can ignore the bits 0 through 11 of the NVMADDR register.

A Page Erase only succeeds, if the target address is a page that is not write-protected. Erasing a write-protected page fails and result in the WRERR bit being set in the NVMCON register.

The following code shows the code for a single Page Erase operation at address 0x1008000.

Page Erase Code Example:

```
…

// set destination page address
    NVMADDR = 0x1008000;     // page physical address

// define Flash operation
    NVMCONbits.NVMOP = 0x4;    // NVMOP for Page Erase

// Enable Flash Write
    NVMCONbits.WREN = 1;

// commence programming
    NVMInitiateOperation();    // see Initiate NVM Operation (Unlock Sequence Example)

// Wait for WR bit to clear
    while(NVMCONbits.WR);

// Disable future Flash Write/Erase operations
    NVMCONbits.WREN = 0;

// Check Error Status
    if(NVMCON & 0x3000)    // mask for WRERR and LVDERR bits
    {
        // process errors
    }

…
```

15.16.1 Page Erase Retry

Page Erase Retry is a method to improve the life of a Flash by attempting to erase again if the Page Erase was not successful. The user can only use the Erase Retry for a Page Erase.

Page Erase Retry works by increasing the voltage used on the Flash when erasing. Initially, the minimum voltage necessary is applied by setting the RETRY[1:0] bits (NVMCON2[9:8]) = 00. If the page erase is not successful, the voltage may be increased by incrementing the setting of the RETRY[1:0] bits.

Note: Each Flash page, as it ages and wears, can have different voltage requirements; therefore, a higher setting on one Flash page does not indicate that the same setting must be used on all pages.

The maximum voltage for Page Erase is used when the RETRY[1:0] bits = 11. If Page Erase is not successful after 7 trials, the Flash for that page, or the Words that did not erase, must be considered non-functional.

Together with the normal Page Erase controls, Page Erase Retry also uses the WS[4:0], CREAD1, VREAD1 and RETRY[1:0] bits in the NVMCON2 register. The ERS[3:0] bits (NVMCON2[31:28]) are for the benefit of software performing the programming sequence in the event that a drop in power causes a BOR event but not a POR event.

Perform the following steps to set up a Page Erase Retry:

1. Set the NVMADDR register with the address of the page to be erased.
2. Execute the write unlock sequence.
3. Save the value of the NVMCON2 register.
4. Do the following in the NVMCON2 register:
   a. Set the ERS[3:0] bits as desired.
   b. Set the WS[4:0] bits per the description.
   c. Set the VREAD1 bit to '1'.
   d. Set the CREAD1 bit to '1'.
   e. Set the RETRY[1:0] bits to '00'.
5. Run the unlock sequence using the Page Erase command to start the sequence.

6. Wait for the WR bit (NVMCON[15]) to be cleared by hardware.
7. Clear the WREN bit (NVMCON[14]).
8. Verify the erase using the CPU. To shorten the verify time, use CREAD1 = 1 to perform a hardware compare to logic '1' of each bit in the Flash Word including ECC. A successful compare yields a read of 0x00000001 in the lowest addressed Word in a Flash Word (128 bits). This is the Compare Word. All other Words are 0x00010000. If any bit is logic '0', all Words in the Flash Word read 0x00000000. Remember to increment the address by the number of bytes in a Flash Word between reads.
9. If all Compare Words verify correctly, the Page Erase Retry process is complete. Go to step 11.
10. If a Compare Word yields a read of 0x00000000, perform steps 4 through 9 up to six more times with the following change to step 4:
    a. Increment the RETRY[1:0] bits by one if the bit has not already reached the '11' setting.
    b. Maintain all other fields.
11. Restore the value of the NVMCON2 register, which was saved in step 3.

Notes:

1. When CREAD1 is set to '1' to perform hardware compare, the compare happens on the entire Flash panel (PFM, BFM). Hence, execute the code that performs this page erase retry operation from SRAM.
2. When the VREAD1 = 1, the Flash uses the WS[3:0] bits for the Flash access wait state generation to the panel selected by NVMADDR. Software is responsible for writing the VREAD1 bit back to '0' when both erase and verify is complete.
3. The device configuration boot page (the page containing the DEVCFGx values) does not support Page Erase Retry.

The following code provides code for a single page erase operation at address 0x1008000, in case of Page Erase Retry.

Page Erase Retry Code Example:

```
uint32_tsaveNVMCON2;
uint32_t*cmpPtr;
uint8_terased;
uint8_ttryCount;

// set destination page address
    NVMADDR = 0x1008000;    // Page physical address

// define flash operation
    NVMCONbits.NVMOP = 0x4;    // NVMOP for Page Erase

// Unlock sequence
    NVMKEY = 0x0;
    NVMKEY = 0xAA996655;
    NVMKEY = 0x556699AA;

// save NVMCON2
    saveNVMCON2 = NVMCON2;

// set up Page Erase Retry
    NVMCON2bits.ERS = 0;    // Stage 0 - SW use only
    NVMCON2bits.VREAD1 = 1;
    NVMCON2bits.CREAD1 = 1;
    NVMCON2bits.RETRY = 0b00;

    tryCount = 0;        // Up to 4 attempts

    do {
        tryCount++;

        // commence programming
        NVMInitiateOperation();
```

```
        // Wait for WR bit to clear
        while(NVMCONbits.WR);

        // Turn off WREN
        NVMCONbits.WREN = 0;

        // Check that the page was erased
        erased = 1;
        cmpPtr = (uint32_t *)NVMADDR;
        erased &= (*cmpPtr == 0x00000001);
        cmpPtr++;
        erased &= (*cmpPtr == 0x00010000);
        cmpPtr++;
        erased &= (*cmpPtr == 0x00010000);
        cmpPtr++;
        erased &= (*cmpPtr == 0x00010000);

        if (!erased) {
            // Erase failed. Try with different settings.
            NVMCON2bits.RETRY++;

            NVMCONbits.NVMOP = 0x4;
            NVMCONbits.WREN = 1;
        }
    } while (!erased && (tryCount < 4));

// Restore settings
    NVMCON2 = saveNVMCON2;
```

15.17 Program Flash Memory (PFM) Erase

Program Flash memory can be erased entirely. All three discrete NVMOP values, 0111, 0110, 0101, do the same operation of erase of entire Flash. When erasing the entire PFM area, in case of RTSP (Run Time Self Programming), the code must be executing from BFM. When erasing the entire PFM area, PFM write-protection must be completely disabled.

The following code shows code for erasing the entire Flash bank.

Program Flash Erase Code Example:

```
…

// define Flash operation
    NVMCONbits.NVMOP = 0x7;        // NVMOP for entire PFM erase

// Enable Flash Write
    NVMCONbits.WREN = 1;

// commence programming
    NVMInitiateOperation();        // see Initiate NVM Operation (Unlock Sequence Example)

// Wait for WR bit to clear
    while(NVMCONbits.WR);

// Disable future Flash Write/Erase operations
    NVMCONbits.WREN = 0;

// Check Error Status
    if(NVMCON & 0x3000)        // mask for WRERR and LVDERR bits
    {
         // process errors
    }

…
```

15.18 Device Code Protection Bit (CP)

The PIC32CX-BZ3 family of devices features code protection, which, when the user enables it, prevents reading of the Flash memory by an external programming device (SWD through DSU).

If enabling code protection, the user can only disable it by erasing the device with the `Chip Erase` command through an external programmer. See *Device Service Unit (DSU)* from Related Links.

When programming a device that opts to utilize code protection, the external programming device must perform verification prior to enabling code protection. Enabling code protection must be the last step of the programming process. For the location of the code protection enable bits, see *PIC32CX-BZ3 Programming Specification* and *System Configuration Registers (CFG)* from Related Links.

Related Links

16. Device Service Unit (DSU)
22. System Configuration and Register Locking (CFG)

15.19 Effects of Various Resets

Device Resets, other than a Power-on Reset (POR), reset the entire contents of the NVMPWP and NVMLBWP registers. All other register content persists through a non-POR reset.

All Flash Controller registers are forced to their reset states upon a POR.

15.20 Register Summary

See *FC* module in the *Product Memory Mapping Overview* from Related Links for the base address.

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	NVMCON	7:0					NVMOP[3:0]			
		15:8	WR	WREN	WRERR	LVDERR				
		23:16								
		31:24								
0x04 ... 0x0F	Reserved									
0x10	NVMCON2	7:0								NVMPREPG
		15:8		TEMP	CREAD1	VREAD1			RETRY[1:0]	
		23:16				WS[4:0]				
		31:24	ERS[3:0]							SLEEP
0x14 ... 0x1F	Reserved									
0x20	NVMKEY	7:0	NVMKEY[7:0]							
		15:8	NVMKEY[15:8]							
		23:16	NVMKEY[23:16]							
		31:24	NVMKEY[31:24]							
0x24 ... 0x2F	Reserved									
0x30	NVMADDR	7:0	NVMADDR[7:0]							
		15:8	NVMADDR[15:8]							
		23:16	NVMADDR[23:16]							
		31:24	NVMADDR[31:24]							
0x34 ... 0x3F	Reserved									
0x40	NVMDATA0	7:0	NVMDATA[7:0]							
		15:8	NVMDATA[15:8]							
		23:16	NVMDATA[23:16]							
		31:24	NVMDATA[31:24]							
0x44 ... 0x4F	Reserved									
0x50	NVMDATA1	7:0	NVMDATA[7:0]							
		15:8	NVMDATA[15:8]							
		23:16	NVMDATA[23:16]							
		31:24	NVMDATA[31:24]							
0x54 ... 0x5F	Reserved									
0x60	NVMDATA2	7:0	NVMDATA[7:0]							
		15:8	NVMDATA[15:8]							
		23:16	NVMDATA[23:16]							
		31:24	NVMDATA[31:24]							
0x64 ... 0x6F	Reserved									
0x70	NVMDATA3	7:0	NVMDATA[7:0]							
		15:8	NVMDATA[15:8]							
		23:16	NVMDATA[23:16]							
		31:24	NVMDATA[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74 ... 0xBF	Reserved									
0xC0	NVMSRCADDR	7:0	NVMSRCADDR[7:0]							
		15:8	NVMSRCADDR[15:8]							
		23:16	NVMSRCADDR[23:16]							
		31:24	NVMSRCADDR[31:24]							
0xC4 ... 0xCF	Reserved									
0xD0	NVMPWPLT	7:0	PWPLT[7:0]							
		15:8	PWPLT[15:8]							
		23:16	PWPLT[23:16]							
		31:24	ULOCK							
0xD4 ... 0xDF	Reserved									
0xE0	NVMPWPGTE	7:0	PWPGTE[7:0]							
		15:8	PWPGTE[15:8]							
		23:16	PWPGTE[23:16]							
		31:24	ULOCK							
0xE4 ... 0xEF	Reserved									
0xF0	NVMLBWP	7:0	LBWP[7:0]							
		15:8	LBWP[15:8]							
		23:16	LBWP[23:16]							
		31:24	ULOCK							

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

15.21 Register Description

The following NVM control registers control the Flash program, erase and write protection operations:

- NVMCON: Programming Control Register
  - This register is the control register for Flash program/erase operations. The following are the uses of this register:
    - Selects the operation to be performed
    - Initiates the operation
    - Provides status of the result after completing the operation
- NVMCON2: Programming Control2 Register
  - This register is the control and status register for Flash program/erase operations.
- NVMKEY: Programming Unlock Register
  - This is a write-only register that helps to or that helps the user to implement an unlock sequence to help prevent accidental writes/erasures of Flash memory and write permission settings.
- NVMADDR: Flash Address Register
  - This register stores the physical target address for row, Quad Double Word and Single Double Word programming as well as page erasing.
- NVMDATAx: Flash Program Data Register (x = 0-3)

  - These registers hold the data to be programmed during Flash Word program operations.
- NVMSRCADDR: Source Data Address Register
  - This register points to the physical address of the data to be programmed when executing a row program operation.
- NVMPWPLT: Flash Program Write Protect Lower Register
  - This register sets the program flash pages lower than provided address as a write protected.
- NVMPWPGTE: Flash Program Write Protect Greater Register
  - This register sets the program flash pages greater than provided address as a write protected.
- NVMLBWP: Flash Boot Write Protect Register
  - This register sets the boot flash partition pages as a write protected.

The following is 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
- HS = Hardware Set
- HC = Hardware Cleared

15.21.1 NVMCON - Programming Control Register

Name: NVMCON
Offset: 0x00
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
	WR	WREN	WRERR	LVDERR				
Access	R/HS/HC	R/W	R/HS/HC	R/HS/HC				
Reset	0	0	0	0				

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

Bit 15 – WR Write Control Bit(1)

Note: This field can only be modified when WREN = 1, TEMP = 1 and the NVMKEY unlock sequence is satisfied.

Value	Description
1	Initiate a Flash operation. Hardware clears this bit when the operation completes
0	Flash operation complete or inactive

Bit 14 – WREN Write Enable Bit(1)

Value	Description
1	Enables writes to WR
0	Disables writes to WR

Bit 13 – WRERR Write Error Bit(1)

Note: Cleared by setting NVMOP == 0000b and initiating a Flash operation (WR)

Value	Description
1	Program or erase sequence did not complete successfully
0	Program or erase sequence completed normally

Bit 12 – LVDERR Low Voltage Detect Error Bit(1)

The error is only captured for programming/erase operations (when WR = 1).

Note: Cleared by setting NVMOP == 0000b and initiating a Flash operation (WR)

Value	Description
1	Low voltage is detected (possible data corruption if WRERR is set)
0	Normal voltage is detected

Bits 3:0 – NVMOP[3:0] NVM Operation bits

These bits are only writable when WREN = 0.

Value	Description
1111	Reserved
1110	Chip Erase Operation: Erases PFM, BFM (except configuration page) when accessed through SWD interface only
...	
...	
1000	Reserved
0111	Program erase operation: erase all of program Flash memory (PFM) (all pages must be unprotected)
0110	Upper program Flash memory erase operation: erases only the upper mapped region of program Flash (all pages in that region must be unprotected). It is a single bank flash in PIC32CX-BZ3, so this NVMOP performs the same as NVMOP = 0111.
0101	Lower program Flash memory erase operation: erases only the lower mapped region of program Flash (all pages in that region must be unprotected). It is a single bank flash in PIC32CX-BZ3, so this NVMOP performs the same as NVMOP = 0111.
0100	Page erase operation: erases page selected by NVMADDR if it is not write-protected
0011	Row program operation: programs row selected by NVMADDR if it is not write-protected
0010	Quad Word (128-bit) program operation: programs the 128-bit Flash Word selected by NVMADDR if it is not write-protected
0001	Word program operation: programs Word selected by NVMADDR if it is not write-protected(2)
0000	No operation

Notes:

1. These bits are only reset by a POR and are not affected by other Reset sources.
2. This operation results in a No Operation (NOP) when the Dynamic Flash ECC Configuration bits = 00 (ECCCTL[1:0](CFGCON0[29:28])), which enables ECC at all times. This command executes for all other ECCCTL[1:0] bit settings but does not write the ECC bits for the Word. It can cause Double-bit Error Detected (DED) errors if dynamic Flash ECC is enabled (ECCCTL[1:0] = 01).

15.21.2 NVMCON2 - Programming Control 2 Register

Name: NVMCON2
Offset: 0x10
Reset: 0x011F4000
Property: -

Bit	31	30	29	28	27	26	25	24
	ERS[3:0]							SLEEP
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				1

Bit	23	22	21	20	19	18	17	16
				WS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
		TEMP	CREAD1	VREAD1			RETRY[1:0]	
Access		R	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
								NVMPREPG
Access								R/W
Reset								0

Bits 31:28 – ERS[3:0] Erase Retry State
The software uses these bits to track the software state of the erase retry procedure in the event of a system Reset ($\overline{MCLR}$) or Brown-out Reset (BOR) event.

Bit 24 – SLEEP Power Down in Sleep Mode
Note: This field can only be modified when the NVMKEY unlock sequence is satisfied.

Value	Description
1	Configures Flash for power down when the system is in Sleep mode
0	Configures Flash for standby when the system is in Sleep mode

Bits 20:16 – WS[4:0] Flash Access Wait State Control for VREAD1 = 1
Notes:
1. When VREAD1 = 1, WS[4:0] only affects the memory containing NVMADDR[31:0].
2. The NVMKEY unlock sequence must be satisfied to modify this field.

Value	Description
11111	31 wait states (32 total system clocks)
11110	30 wait states (31 total system clocks)
. . .	
00010	2 wait states (3 total system clocks)
00001	1 wait state (2 total system clocks)
00000	0 wait state (1 total system clock)

Bit 14 – TEMP Operating Temperature Control bit
This bit is always read as '1' for the PIC32CX-BZ3 device.

Bit 13 – CREAD1 Compare Read of Logic 1 bit

Compare read 1 causes all bits in a Flash Word (including ECC if it exists) to be evaluated during the read. If all bits are 1, the lowest Word in the Flash Word evaluates to 0x0000_0001, all other Words are 0x0001_0000. If any bit is 0, the read evaluates to 0x0000_0000 for all Words in the Flash Word.

Notes:

1. When using erase retry in an ECC Flash system, use CREAD1 = 1.
2. To modify this field, satisfy the NVMKEY unlock sequence.

Value	Description
1	Compare read enabled, only if VREAD1 = 1
0	Compare read disabled

Bit 12 – VREAD1 Verify Read of logic 1 Control bit

Notes:

1. When VREAD1 = 1, Flash Wait state control is from WS[] for the memory containing NVMADDR[].
2. Using erase retry and verify read procedure increases the life of the Flash memory.
3. This field becomes modifiable only when NVMCON.WR == 0 and the NVMKEY unlock sequence is satisfied.

Value	Description
1	Selects erase retry procedure with verify read
0	Selects single erase without verify read

Bits 9:8 – RETRY[1:0] Erase Retry Control bit, only used when VREAD1 = 1

Note: This field becomes modifiable only when NVMCON.WR == 0.

Value	Description
11	Erase strength for fourth up to seventh retry cycle
10	Erase strength for third retry cycle
01	Erase strength for second retry cycle
00	Erase strength for first retry cycle

Bit 0 – NVMPREPG NVM Pre-Program Control Bit

Note: This field can only be modified when NVMCON.NVMWR == 0.

Value	Description
1	Program Operations include Pre-Program step
0	Program Operations exclude Pre-Program step

15.21.3 NVMKEY – Programming Unlock Register

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

Bit	31	30	29	28	27	26	25	24
	NVMKEY[31:24]							
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
	NVMKEY[23:16]							
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
	NVMKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NVMKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NVMKEY[31:0] Unlock Register bits
These bits are write-only and read '0' on any read.
Note: This register is used as part of the unlock sequence to prevent inadvertent writes to the program Flash.

15.21.4 NVMADDR – Flash Address Register

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

Bit	31	30	29	28	27	26	25	24
	NVMADDR[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
	NVMADDR[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
	NVMADDR[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
	NVMADDR[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 – NVMADDR[31:0] Flash (Word) Address bits

Table 15-5. Flash (Word) Address Bits

NVMOP	Flash Address Bits
Page Erase	• Address identifies the page to erase • Any address within a 4 Kbytes page boundary will cause the page to be erased
Row program	• Address identifies the row to program • The value of the address must be aligned to a row boundary
Word program	• Address identifies the 32-bit Word to program • NVMADDR[1:0] bits are ignored • Must be aligned to a Word boundary
Quad Word program	• Address identifies the 128-bit Quad Word to program • NVMADDR[3:0] bits are ignored • Must be aligned to a Quad Word boundary

Notes:

1. Hardware prevents writes to this register when NVMCON.WR = `1`.
2. For all other NVMOP[3:0] bit settings, the Flash address is ignored. For additional information on these bits, see the *NVMCON* register from Related Links.
3. The bits in this register are reset by a POR only and are not affected by other Reset sources.

Related Links
15.21.1. NVMCON

15.21.5 NVMDATA0 – Flash Program Data Register 0

Name: NVMDATA0
Offset: 0x40
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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 – NVMDATA[31:0] Flash Programming Data bits
The value in this register is written to Flash when a program operation is commanded.

- Single Word program (32-bit)
  - Writes NVMDATA0 to the target Flash address defined in NVMADDR[31:2].
- Quad Word program (128-bit)
  - Writes NVMDATA3:NVMDATA2:NVMDATA1:NVMDATA0 to the target Flash address defined in NVMADDR[31:4]. NVMDATA0 contains the Least Significant Instruction Word.

Notes:

1. Hardware prevents writes to this register when NVMCON.WR = 1.
2. The bits in this register are reset on a POR only and are unaffected by other Reset sources.

15.21.6 NVMDATA1 – Flash Program Data Register 1

Name: NVMDATA1
Offset: 0x50
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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 – NVMDATA[31:0] Flash Programming Data bits
The value in this register is written to Flash when a program operation is commanded.

- Single Word program (32-bit)
  - Writes NVMDATA0 to the target Flash address defined in NVMADDR[31:2].
- Quad Word program (128-bit)
  - Writes NVMDATA3:NVMDATA2:NVMDATA1:NVMDATA0 to the target Flash address defined in NVMADDR[31:4]. NVMDATA0 contains the Least Significant Instruction Word.

Notes:

1. Hardware prevents writes to this register when NVMCON.WR = 1.
2. The bits in this register are reset on a POR only and are unaffected by other Reset sources.

15.21.7 NVMDATA2 – Flash Program Data Register 2

Name: NVMDATA2
Offset: 0x60
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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 – NVMDATA[31:0] Flash Programming Data bits
The value in this register is written to Flash when a program operation is commanded.

- Single Word program (32-bit)
  - Writes NVMDATA0 to the target Flash address defined in NVMADDR[31:2].
- Quad Word program (128-bit)
  - Writes NVMDATA3:NVMDATA2:NVMDATA1:NVMDATA0 to the target Flash address defined in NVMADDR[31:4]. NVMDATA0 contains the Least Significant Instruction Word.

Notes:

1. Hardware prevents writes to this register when NVMCON.WR = 1.
2. The bits in this register are reset on a POR only and are unaffected by other Reset sources.

15.21.8 NVMDATA3 – Flash Program Data Register 3

Name: NVMDATA3
Offset: 0x70
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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
	NVMDATA[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 – NVMDATA[31:0] Flash Programming Data bits
The value in this register is written to Flash when a program operation is commanded.

- Single Word program (32-bit)
  - Writes NVMDATA0 to the target Flash address defined in NVMADDR[31:2].
- Quad Word program (128-bit)
  - Writes NVMDATA3:NVMDATA2:NVMDATA1:NVMDATA0 to the target Flash address defined in NVMADDR[31:4]. NVMDATA0 contains the Least Significant Instruction Word.

Notes:

1. Hardware prevents writes to this register when NVMCON.WR = 1.
2. The bits in this register are reset on a POR only and are unaffected by other Reset sources.

15.21.9 NVMSRCADDR – Source Data Address Register

Name: NVMSRCADDR
Offset: 0xC0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NVMSRCADDR[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
	NVMSRCADDR[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
	NVMSRCADDR[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
	NVMSRCADDR[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 – NVMSRCADDR[31:0] Source Data (Word) Address bits
This is the system physical Word address of the data (in DRM) to be programmed into the Flash when NVMCON.NVMOP is set to row programming.
Notes:

1. Hardware prevents writes to this register when NVMCON.WR = 1.
2. The bits in this register are reset on a POR only and are unaffected by other reset sources.

15.21.10 NVMPWPLT – Flash Program Write Protect Lower Register

Name: NVMPWPLT
Offset: 0xD0
Reset: 0x80000000
Property: -

Bit	31	30	29	28	27	26	25	24
	ULOCK							
Access	R/C							
Reset	1							

Bit	23	22	21	20	19	18	17	16
	PWPLT[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
	PWPLT[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
	PWPLT[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 31 – ULOCK NVMPWPLT Register Unlock bit

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied.
2. This field can be cleared at the same time as writing to PWPLT[23:0].

Value	Description
1	NVMPWPLT register is not locked and can be modified
0	NVMPWPLT register is locked and cannot be modified

Bits 23:0 – PWPLT[23:0] Flash Program Write Protect Less Than Address
Pages at Flash addresses less than this value are write-protected.

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied, and ULOCK = 1.
2. This is a byte address force to align to page boundaries.

15.21.11 NVMPWPGTE – Flash Program Write Protect Greater Register

Name: NVMPWPGTE
Offset: 0xE0
Reset: 0x80FFFFFF
Property: -

Bit	31	30	29	28	27	26	25	24
	ULOCK							
Access	R/C							
Reset	1							

Bit	23	22	21	20	19	18	17	16
	PWPGTE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PWPGTE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

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

Bit 31 – ULOCK NVMPWPGTE Register Unlock bit

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied.
2. This field can be cleared at the same time as writing to PWPGTE[23:0].

Value	Description
1	NVMPWPGTE register is not locked and can be modified
0	NVMPWPGTE register is locked and cannot be modified

Bits 23:0 – PWPGTE[23:0] Flash Program Write Protect Address

Pages at Flash addresses greater than or equal to this value are write-protected.

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied and ULOCK = 1.
2. This is a byte address forced to align to page boundaries.

15.21.12 NVMLBWP - Flash Boot Write Protect Register

Name: NVMLBWP
Offset: 0xF0
Reset: 0x80FFFFFF
Property: -

Bit	31	30	29	28	27	26	25	24
	ULOCK							
Access	R/C							
Reset	1							

Bit	23	22	21	20	19	18	17	16
	LBWP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	LBWP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

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

Bit 31 – ULOCK Lower Boot Write Protect (LBWPn) Unlock bit

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied.
2. This field can be cleared at the same time as writing to LBWP[msb:lsb].

Value	Description
1	LBWPn bits are not locked and can be modified
0	LBWPn bits are locked and cannot be modified

Bits 23:0 – LBWP[23:0] Boot Pages Write Protect bits

Notes:

1. This field can only be modified when the NVMKEY unlock sequence is satisfied and ULOCK = 1.
2. The OTP page is always erase protected, and its associated LBWP bit is only for write protection.

Value	Description
1	Erase and write protection for upper boot page n is enabled
0	Erase and write protection for upper boot page n is disabled

16. Device Service Unit (DSU)

16.1 Overview

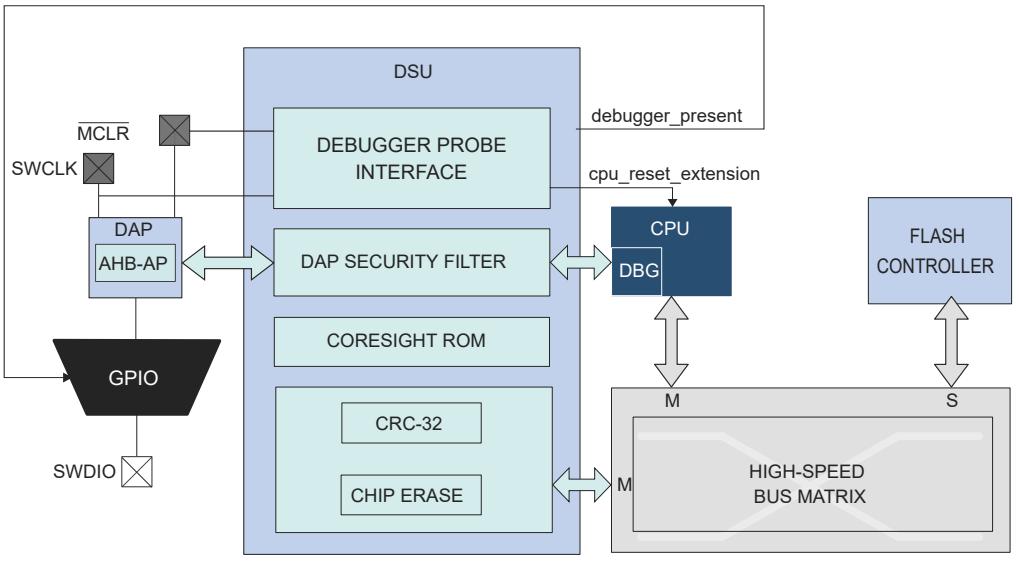
The Device Service Unit (DSU) provides a means of detecting debugger probes. It enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU Reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight debug ROM 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 are unavailable when the Code Protect bit and SECCFG.DEBUG_LCK bit protect the device.

16.2 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 with DMA Connection
- Debug Access Port Security Filter

16.3 DSU Block Diagram

Figure 16-1. DSU Block Diagram



16.4 Signal Description

The DSU uses three signals to function.

Table 16-1. Signal Description

Signal Name	Type	Description
$\overline{\text{MCLR}}$	Digital input	External Reset pin
CM4_SWCLK	Digital input	Software clock pin
CM4_SWDIO	Digital I/O	Software bidirectional data pin

16.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

16.5.1 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 user changes the SWCLK pin function 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, disable it by setting CFGCON0.HPLUGDIS = `1`.

Related Links

16.6.3. Debugger Probe Detection

16.5.2 Power Management

The DSU continues to operate in any Sleep mode (Standby Sleep, Idle) where the selected source clock is running.

16.5.3 Clocks

The DSU bus clocks (CLK_DSU_APB (PB2_CLK) and CLK_DSU_AHB (SYS_CLK)) can be enabled and disabled by the CRU.

16.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). The user must first configure the DMAC to use DMA requests with this peripheral. See *Direct Memory Access Controller (DMAC)* from Related Links.

The CFG.DCCDMALEVEL bit field must be configured depending on the DMA channels access modes (read or write for DCC0 and DCC1).

Related Links

26. Direct Memory Access Controller (DMAC)

16.5.5 Interrupts

Not applicable.

16.5.6 Events

Not applicable.

16.5.7 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)

Notes:

- Optional write protection is indicated by the PAC Write Protection property in the register description.
- If halting the CPU in the Debug mode, all write protection is automatically disabled. Write protection does not apply for accesses through an external debugger.

16.5.8 Analog Connections

Not applicable.

16.6 Debug Operation

16.6.1 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

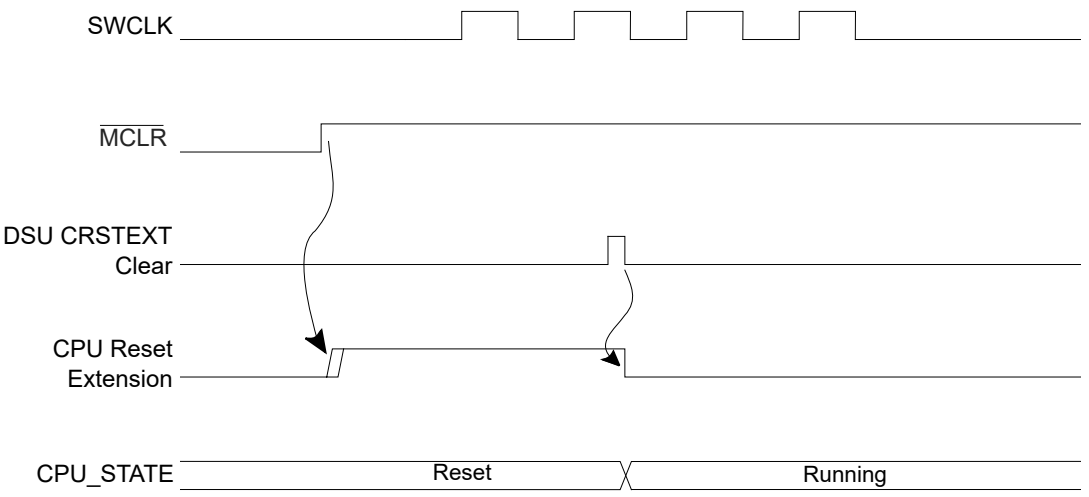
For more details on the ARM debug components, refer to the *ARM Debug Interface v5 Architecture Specification*.

Debug using SWD will be blocked on secured devices (SECCFG.DEBUG_LCK).

16.6.2 CPU Reset Extension

The CPU Reset extension refers to the extension of the Reset phase of the CPU core after releasing the external Reset. This ensures that the CPU is not executing code at start-up while a debugger is connected to the system. The debugger is detected on a $\overline{\text{MCLR}}$ release event when SWCLK is low. At start-up, the 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 Code Protect bit (FCPN0.CP) or the SECCFG.DEBUG_LCK bit is protecting the device. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 16-2. Typical CPU Reset Extension Set and Clear Timing Diagram



16.6.3 Debugger Probe Detection

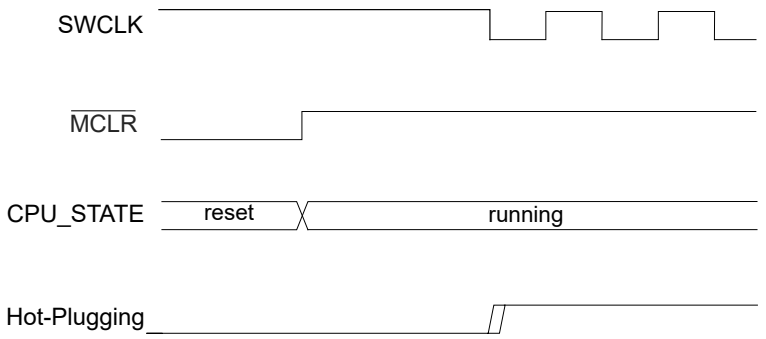
16.6.3.1 Cold-Plugging

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.

16.6.3.2 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 $\overline{\text{MCLR}}$ 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 changing the SWCLK pin function 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. The availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 16-3. Hot-Plugging Detection Timing Diagram



The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. After the detection, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, the Hot-Plugging is not available when the Code Protect bit (FCPN0.CP) or SECCFG.DEBUG_LCK bit is protecting the device.

In this detection, the user must correctly power the pads. Thus, at cold start-up, this detection cannot be done until POR is released. If the device is protected using Code Protect bit (FCPN0.CP) or SECCFG.DEBUG_LCK bit, 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 deasserting the external Reset before POR release, the user must retry the above procedure until it gets connected to the device.

16.7 Chip Erase

Chip erase consists of removing all sensitive information stored in the chip and clearing the Code Protect bit. Therefore, this erases all volatile memories and the Flash memory.
Note: The OTP Boot flash memory page will not be erased.

When the device is protected using the FCPN0.CP or SECCFG.DEBUG_LCK bit, the debugger must first Reset the device to be detected. This ensures that internal registers are reset after removing the Protected state. The user can trigger the chip erase operation 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). After issuing, the module clears volatile memories prior to erasing the Flash array. To ensure the completion of the chip erase operation, 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, the recommendation is to issue a Chip Erase command after a Cold-Plugging procedure to ensure that the device is in a known and safe state.

The following is the recommended sequence:

1. Issue the Cold-Plugging procedure (See *Cold-Plugging* from Related Links), then the device performs the following:
   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 Lock Row and 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
16.6.3.1. Cold-Plugging

16.8 Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the Code Protect bit.

Important: If securing the device using SECCFG.DEBUG_LCK, row programming is not supported because SRAM memory is not accessible by the external debugger. Either Word programming or Quad Word programming is possible based on the CFGCON0.ECCCTL setting.

The programming procedure is as follows:

1. At power-up, $\overline{\text{MCLR}}$ 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)* electrical characteristics from Related Links). The system is going to be in this Static state until the internally regulated supplies have reached a safe Operating state.
2. The power management 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. $\overline{\text{MCLR}}$ 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. The user must issue a Chip Erase command to ensure the Flash is fully erased before programming.
7. Programming is available through the AHB-AP. See *PIC32CX-BZ3 Programming Specification* and *Flash Controller* from Related Links.
   Note: Programming of Boot Flash Memory (BFM) pages is allowed only when the bit 3 at address 0x410001FC is set to '1'.
8. Completing the operation, the chip can be restarted either by asserting $\overline{\text{MCLR}}$, toggling power or writing a '1' to the Status A register CPU Reset Phase Extension bit (STATUSA.CRSTEXT). Ensure that the SWCLK pin is high when releasing $\overline{\text{MCLR}}$ to prevent extending the CPU Reset.

Related Links

17.4.2.1.1. Power-on Reset (POR)

15. Flash Controller (FC)

16.9 Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and the user can accomplish this by setting the Code Protect bit or DEBUG_LCK bit.

The following are the two protection mechanisms in the PIC32CX-BZ3 devices:

1. Code Protect – If enabling the Code Protect bit by configuring FCPN0.CP in Boot Flash memory, the device is locked from programming and debugging. Only chip erase can retrieve the device to the normal programming and debugging condition. When issuing a chip erase, it erases sensitive information from volatile memory and Flash.
2. Secured Device – DEBUG_LCK bits in eFuse (SECCFG.DEBUG_LCK in Root of Trust) determines if the device is locked for debugging. If the DEBUG_LCK bits are non-zero, device is a secured device. Securing of the device implies:
   a. The user cannot execute any unauthenticated firmware.
   b. The debug features of the device are not available and are locked down.
   c. Device programming through SWD interface is still available. Debugger can be plugged in only through the Cold-Plugging procedure. The Hot-Plugging feature is not available (see *Cold-Plugging* and *Hot-Plugging* from Related Links).
   d. The DEBUG_LCK bits are in eFuse (one time programmable memory); therefore, when locked, the device is permanently locked for debug unlike the Code Protect mechanism, which can be cleared on a chip erase.

Important: On a secured device (non-zero value of SECCFG.DEBUG_LCK), if the boot key is zero, it implies that secure boot code need not authorize the firmware code. Therefore, programming through an external debugger is disabled on a device, which is secured, and secure boot key is zero.

When the device is protected, read/write accesses using the AHB-AP is limited to the DSU address range and DSU commands are restricted.

The DSU implements a security filter that monitors the AHB transactions generated by the ARM AHB-AP 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 user can access the DSU 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

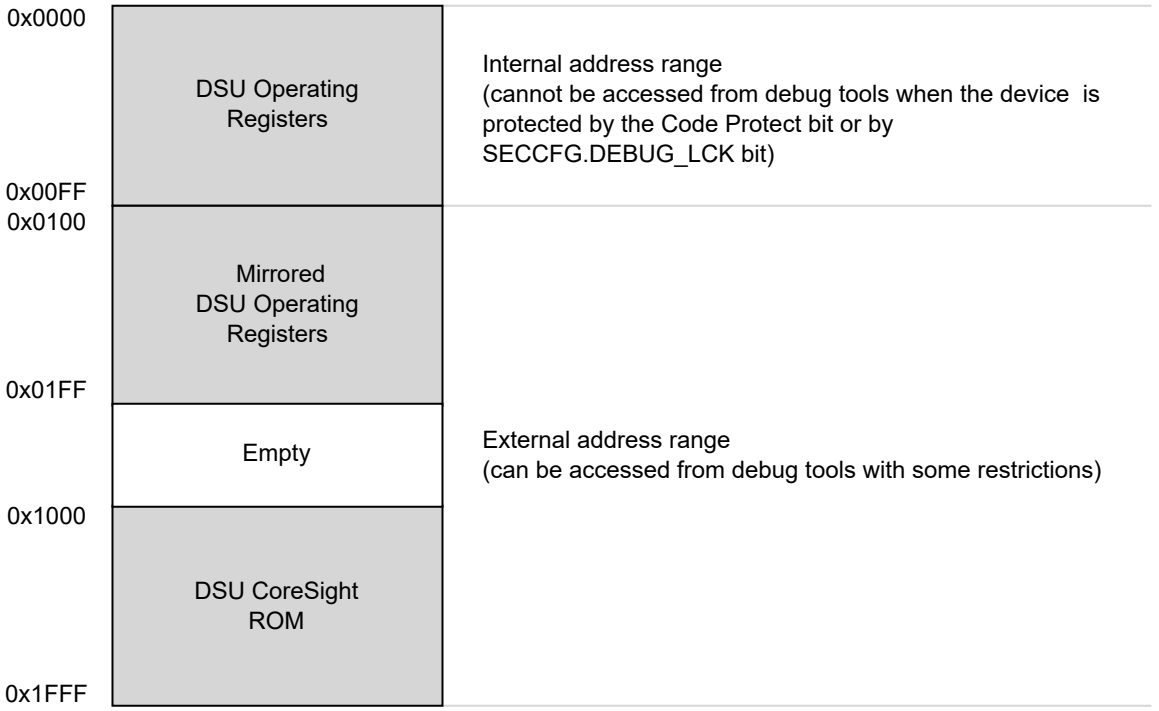
For security reasons, DSU features have limitations when using a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map were mirrored at offset 0x100:

- 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, see Table 16-2.

Figure 16-4. APB Memory Mapping



When the device is protected, some features that are not activated by APB transactions are unavailable.

Table 16-2. Feature Availability Under Protection

Feature	Code Protected and Unsecured	Code Protected and Secured	Not Code Protected and Unsecured	Not Code Protected and Secured
CPU Reset Extension	Yes	Yes	Yes	Yes
Clear CPU Reset Extension	No	No	Yes	No
Debugger Cold-Plugging	Yes	Yes	Yes	Yes
MEM-AP access during Cold-Plugging	No	No	Yes	Yes
Debugger Hot-Plugging	No	No	Yes	No

Notes:

- Code Protected means FCPN0.CP of the Flash fuse configuration bit is set.
- Secured means SECCFG.DEBUG_LCK bits of the eFuse are non zero values.

Related Links

16.6.3.1. Cold-Plugging
16.6.3.2. Hot-Plugging

16.10 Device Identification

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

16.10.1 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 from the PID0 to PID7 CoreSight ROM table registers. The following figure illustrates the conceptual 64-bit peripheral ID.

Figure 16-5. Conceptual 64-bit Peripheral ID

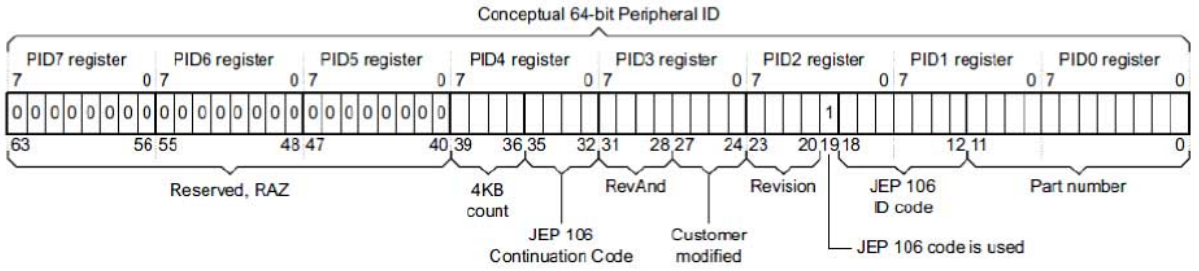


Table 16-3. Conceptual 64-Bit Peripheral ID Bit Descriptions

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*.

16.10.2 Chip Identification Method

The following table shows how the DSU DIS register identifies the device:

Table 16-4. 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	0x9E	Immutable Field = Mask ID [7:0]
DEVSEL	8 bits	RoT eFuse (device_id)	Determines variants of product {VSEL[7:0]}

16.11 Functional Description

16.11.1 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 completion of the current operation. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

16.11.2 Basic Operation

16.11.2.1 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

17. Clock and Reset Unit (CRU)
24. Peripheral Access Controller (PAC)

16.11.2.2 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.

When the device debug is locked through Secure Device, other than DSU external address range, Flash Controller registers, eFuse controller registers and Flash Memory addresses are placed under MEM-AP permissible address range to allow programming.

Table 16-5. Permissible Address Range for External Debugger when Secured Device

Parameter	Description	Value
PERM_ADDR_START1	Permissible address range start	0x4400_0600
PERM_ADDR_END1	Permissible address range end	0x4400_07FF
PERM_ADDR_START2	Second permissible address range start	0x4400_2C00
PERM_ADDR_END2	Second permissible address range end	0x4400_3BFF
FLASH_ADDR_START	Flash address start	0x0080_0000
FLASH_ADDR_END	Flash address end	0x0107_FFFF

Related Links

16.9. Intellectual Property Protection

16.11.2.3 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

16.9. Intellectual Property Protection

16.11.3 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 the following table)

Table 16-6. 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. DATA forced to 0xFFFFFFFF before calculation (no seed).
1-3	Reserved	—

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

16.11.3.1 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). Ensure both are Word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). In general, this value is 0xFFFFFFFF, but it can be, for example, the result of generating a common CRC32 of separate memory blocks through previous CRC32 calculation.

After completion, 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 non inverted if used as the starting point for subsequent CRC32 calculations.

If the device is in Protected state by the Code Protect or SECCFG.DEBUG_LCK security bit, it is only possible to calculate the CRC32 of the whole flash array when operated from the external address space. In most cases, this area is the entire onboard non-volatile memory. The Address, Length and Data registers are forced to predefined values when the CRC32 operation is started, and values written by the user are ignored. This allows the user to verify the contents of a protected device.

The actual test starts 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).

16.11.3.2 Interpreting the Results

The user must monitor the Status A register. After completing the operation, 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.

16.11.4 Debug Communication Channels

The Debug Communication Channels (DCCO 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 or SECCFG.DEBUG_LCK 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 is 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.

16.11.5 System Services Availability when Accessed Externally and Device is Protected

Table 16-7. Available Features when Operated from External Address Range and the Device is Protected

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

16.12 Register Summary

See *DSU* module in the *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0				CE		CRC		SWRST
0x01	STATUSA	7:0				PERR	FAIL	BERR	CRSTEXT	DONE
0x02	STATUSB	7:0				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	PROCESSOR[3:0]				FAMILY[4:1]			
0x1C	CFG	7:0				ETBRAMEN	DCCDMALEVEL[1:0]		LQOS[1:0]	
		15:8								
		23:16								
		31:24								
0x20 ... 0x37	Reserved									
0x38	UUID0	7:0	UUID[7:0]							
		15:8	UUID[15:8]							
		23:16	UUID[23:16]							
		31:24	UUID[31:24]							
0x3C	UUID1	7:0	UUID[7:0]							
		15:8	UUID[15:8]							
		23:16	UUID[23:16]							
		31:24	UUID[31:24]							
0x40	UUID2	7:0	UUID[7:0]							
		15:8	UUID[15:8]							
		23:16	UUID[23:16]							
		31:24	UUID[31:24]							
0x44	UUID3	7:0	UUID[7:0]							
		15:8	UUID[15:8]							
		23:16	UUID[23:16]							
		31:24	UUID[31:24]							
0x48	SECCFG	7:0	DEBUG_LCK[1:0]		UUID_LCK[1:0]					
		15:8					BOOT_KEY_LCK[1:0]		ROOT_KEY_LCK[1:0]	
		23:16								ADD_BOOT_KEY
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	CTR_STAT	7:0	ROLLBACK_CTR[7:0]							
		15:8								
		23:16								
		31:24								
0x50	BOOT_STATUS	7:0	BOOT_STATUS[7:0]							
		15:8								
		23:16								
		31:24								
0x54	BOOT_KEY0	7:0	BOOT_KEY[7:0]							
		15:8	BOOT_KEY[15:8]							
		23:16	BOOT_KEY[23:16]							
		31:24	BOOT_KEY[31:24]							
...										
0x80	BOOT_KEY11	7:0	BOOT_KEY[7:0]							
		15:8	BOOT_KEY[15:8]							
		23:16	BOOT_KEY[23:16]							
		31:24	BOOT_KEY[31:24]							
0x84 ... 0xEF	Reserved									
0xF0	DCFG0	7:0	DCFG[7:0]							
		15:8	DCFG[15:8]							
		23:16	DCFG[23:16]							
		31:24	DCFG[31:24]							
0xF4	DCFG1	7:0	DCFG[7:0]							
		15:8	DCFG[15:8]							
		23:16	DCFG[23:16]							
		31:24	DCFG[31:24]							
0xF8 ... 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									
0x1FE0	PID0	7:0	PARTNBL[7:0]							
		15:8								
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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								

Related Links

8. Product Memory Mapping Overview

16.13 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

16.5.7. Register Access Protection

16.13.1 Control

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

Bit	7	6	5	4	3	2	1	0
				CE		CRC		SWRST
Access				W		W		W
Reset				0		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 2 – CRC 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the cyclic redundancy check algorithm.

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

16.13.2 Status A

Name: STATUSA
Offset: 0x01
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.

16.13.3 Status B

Name: STATUSB
Offset: 0x02
Reset: 0x0x
Property: PAC Write-Protection

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

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-on Reset or an 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 code protected and FCPN0.CP bit in Devconfig Boot Flash memory is set to enable code protection. See *FCPN0* from Related Links.
This bit is never cleared.
Related Links
8.8. FCPN0

16.13.4 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
16.11.3. 32-bit Cyclic Redundancy Check (CRC32)

16.13.5 Length

Name: LENGTH
Offset: 0x08
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.

16.13.6 Data

Name: DATA
Offset: 0x0C
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.

16.13.7 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.

16.13.8 Device Identification

Name: DID
Offset: 0x18
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	PROCESSOR[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 – PROCESSOR[3:0] Processor
The value of this field defines the processor used on the device.

Bits 27:23 – FAMILY[4:0] Product Family
The value of this field corresponds to the product family part of the ordering code.

Bits 21:16 – SERIES[5:0] Product Series
The value of this field corresponds to the product series part of the ordering code.

Bits 15:8 – DIE[7:0] Die Number
Identifies the die family. 0x9E for PIC32CX-BZ3 family of devices.

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. Refer to ordering information for the DEVSEL[7:0] values of different variants.

16.13.9 Configuration

Name: CFG
Offset: 0x1C
Reset: 0x00000002
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				ETBRAMEN	DCCDMALEVEL[1:0]		LQOS[1:0]	
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	1	0

Bit 4 – ETBRAMEN Trace Control
ETB Ram Enable Writing a one to this bit will reserve the first 32 KB of the RAM for the trace ETB ram buffer.

Bits 3:2 – DCCDMALEVEL[1:0] DMA Trigger Level

Value	Description
0x0	DMA trigger rises when DCC is empty
0x1	DMA trigger rises when DCC is full
0x2 – 0x3	Reserved

Bits 1:0 – LQOS[1:0] Latency Quality Of Service
These bits define the priority access during the memory access.

16.13.10 Unique Identifier Register n

Name: UUID
Offset: 0x38 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

These registers contain the unique identifier of the device. It is stored in eFuse memory in Root of Trust module, which is directly driven in this register.

Bit	31	30	29	28	27	26	25	24
	UUID[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
	UUID[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
	UUID[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
	UUID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UUID[31:0] Unique Identifier bits
These bits provide the unique identifier value. The four 32-bit UUID registers contain the 128-bit UUID.

16.13.11 Secure Configuration

Name: SECCFG
Offset: 0x48
Reset: 0x00000000
Property: -

This register contains the secure configuration setting of the device. It is stored in eFuses memory in Root of Trust module, which is directly driven in this register.

Note:
*_LCK bits in this register refer to the program locks of corresponding eFuses.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ADD_BOOT_KEY
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
					BOOT_KEY_LCK[1:0]		ROOT_KEY_LCK[1:0]	
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEBUG_LCK[1:0]		UUID_LCK[1:0]					
Access	R	R	R	R				
Reset	0	0	0	0				

Bit 16 – ADD_BOOT_KEY Additional Boot Key
This bit is the LSB of the Y co-ordinate in the compressed Secure Boot Key.

Bits 11:10 – BOOT_KEY_LCK[1:0] Lock Bits for Secure Boot Key

Value	Description
11	Secure boot key is locked and cannot be programmed
10	Secure boot key is locked and cannot be programmed
01	Secure boot key is locked and cannot be programmed
00	Secure boot key is not locked

Bits 9:8 – ROOT_KEY_LCK[1:0] Lock Bits for Storage Root Key

Value	Description
11	Storage root key is locked and cannot be programmed
10	Storage root key is locked and cannot be programmed
01	Storage root key is locked and cannot be programmed
00	Storage root key is not locked

Bits 7:6 – DEBUG_LCK[1:0] Lock Bits for Debug

Value	Description
11	Debug is locked. Not possible to debug.
10	Debug is locked. Not possible to debug.

Value	Description
01	Debug is locked. Not possible to debug.
00	Debug is not locked.

Bits 5:4 – UUID_LCK[1:0] Programming Lock Bits for Unique ID Fuses

Value	Description
11	Unique ID is locked and cannot be programmed
10	Unique ID is locked and cannot be programmed
01	Unique ID is locked and cannot be programmed
00	Unique ID is not locked

16.13.12 Rollback Counter Status

Name: CTR_STAT
Offset: 0x4C
Reset: 0x0000009C
Property: -

This register has Rollback Counter information. It is stored in eFuse memory in Root of Trust module which is directly driven in this register.

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
	ROLLBACK_CTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ROLLBACK_CTR[7:0] Rollback Counter status
This bit contains the Rollback Counter status bit.

16.13.13 Boot Status

Name: BOOT_STATUS
Offset: 0x50
Reset: 0x0
Property: -

This register reflects the SEC_BOOT.BOOT_STATUS bits in Root of Trust module. The secure boot firmware in ROM manages the SEC_BOOT.BOOT_STATUS bits to indicate secure boot status.

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
	BOOT_STATUS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BOOT_STATUS[7:0]
These bits hold 8-bit code which is written to SEC_BOOT.BOOT_STATUS bits to indicate the secure boot status, such as authentication success, failure or any other indication.

16.13.14 Secure Boot Key n

Name: BOOT_KEY
Offset: 0x54 + n*0x04 [n=0..11]
Reset: 0x00000000
Property: -

These registers read the secure boot key. The secure boot key is stored in eFuse in the Root of Trust module, which is directly driven in this register. The twelve 32-bit BOOT_KEY registers contain the 384-bit secure public boot key.

Bit	31	30	29	28	27	26	25	24
	BOOT_KEY[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
	BOOT_KEY[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
	BOOT_KEY[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
	BOOT_KEY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BOOT_KEY[31:0] Secure Boot key bits
These bits provide the secure boot key.

16.13.15 Device Configuration

Name: DCFG
Offset: 0xF0 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DCFG[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
	DCFG[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
	DCFG[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
	DCFG[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 – DCFG[31:0] Device Configuration

16.13.16 CoreSight ROM Table Entry x

Name: ENTRY
Offset: 0x1000 + n*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 read 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 code protected indicating that the entry is not present.
This bit is cleared at power-up if the device is not code protected indicating that the entry is present.

16.13.17 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.

16.13.18 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: x determined by Debug Access Level (DAL)0x0000000X
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
								SMEMP
Access								R
Reset								x

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 code protected, indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up if the device is code protected, indicating that the system memory is not accessible from a debug adapter.

16.13.19 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] 4 KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4 KB block.

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

16.13.20 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	1	1	0	1	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.

16.13.21 Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x0000009C
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	0	0	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0] Low Part of the JEP-106 Identity Code
These bits will always return 0x9 when read (JEP-106 identity code is 0x29).

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.

16.13.22 Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x0000000A
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	1	0

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 '1' 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 0x2 when read, (JEP-106 identity code is 0x29).

16.13.23 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.

16.13.24 Component Identification 0

Name: CID0
Offset: 0x1FF0
Reset: 0x0D
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 0x0D when read.

16.13.25 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.

16.13.26 Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x05
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 0x05 when read.

16.13.27 Component Identification 3

Name: CID3
Offset: 0x1FFC
Reset: 0xB1
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 0xB1 when read.

17. Clock and Reset Unit (CRU)

17.1 Overview

The Clock and Reset Unit (CRU) provides both clocking and Reset functions. This section describes the clocking and Reset functionality, summarizes the clock distribution and terminology in the PIC32CX-BZ3 device. The CRU handles the clock control to provide system clocks and interface peripheral clocks. The clock is distributed to a different peripheral through peripheral-specific configuration. The CRU controls switching and synchronization of clock sources.

17.2 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)
  - 64 MHz System PLL (RFPLLPGM MHz)
- Provides Control Registers for PLL
- Provides 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 the 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 for the System
- Multiple PB Clock (Peripheral Clock) Dividers
- One System Clock, SYS_CLK, which Almost All Clocks Used Throughout the System are Derived from
- Three Peripheral Clocks, Created by Independent Integer Dividers of the SYS_CLK:
  - PB1_CLK: PB-Bridge-D and PB-Bridge-A
  - 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)
  - 64 MHz system PLL (RFPLL PGM MHz), SPLL_CLK1
  - REFI pin

- Provides Clock Source for Backup Core for Sleep Operations

17.3 Clock System

17.3.1 Block Diagram

The clock system along with the PMD provides gated clock output for all peripheral buses. See *Peripheral Module Disable* from Related Links. The following figure illustrates the clock system block diagram.

Figure 17-1. Clock System Block Diagram

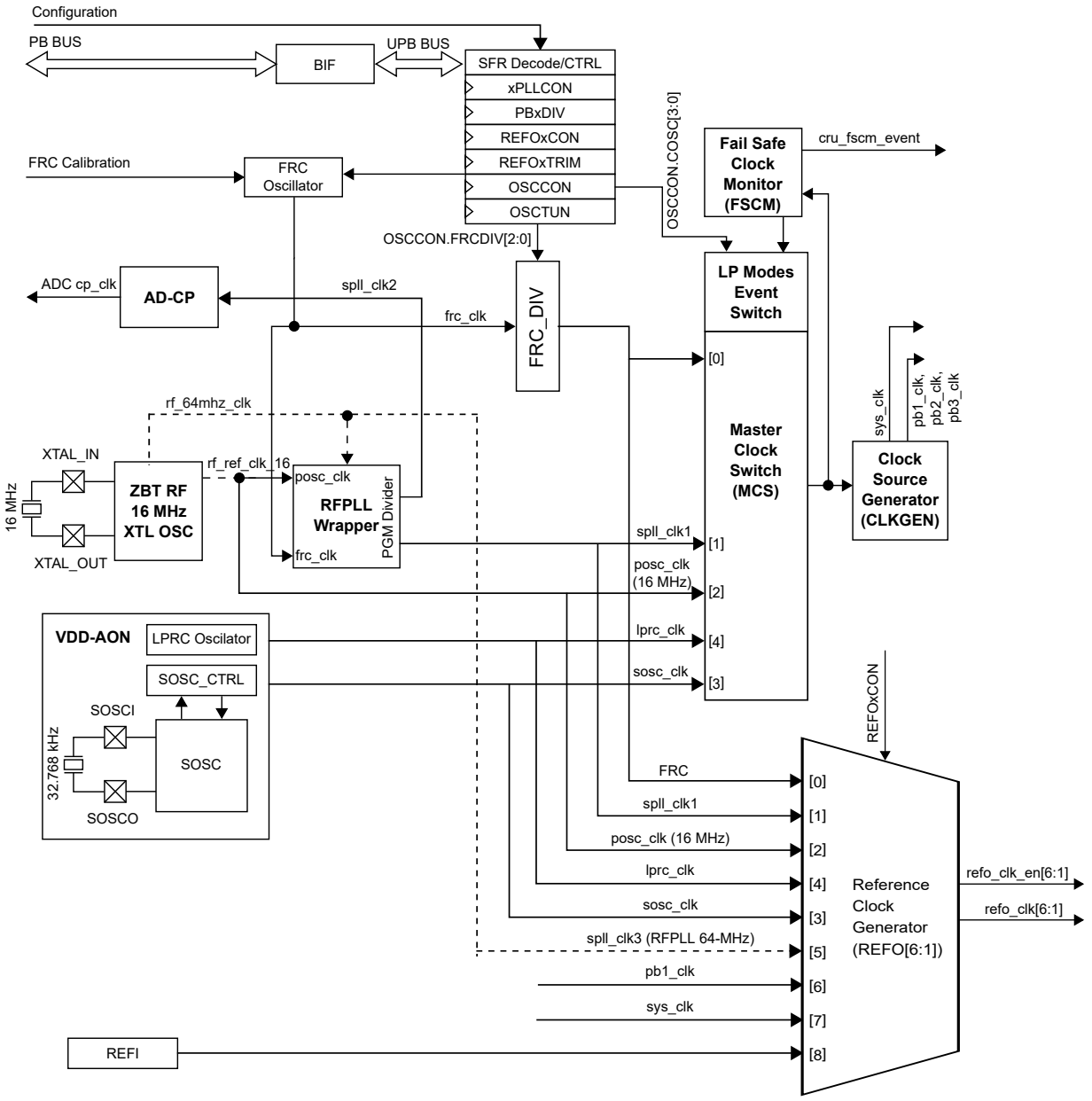


Figure 17-2. RFPLL Wrapper

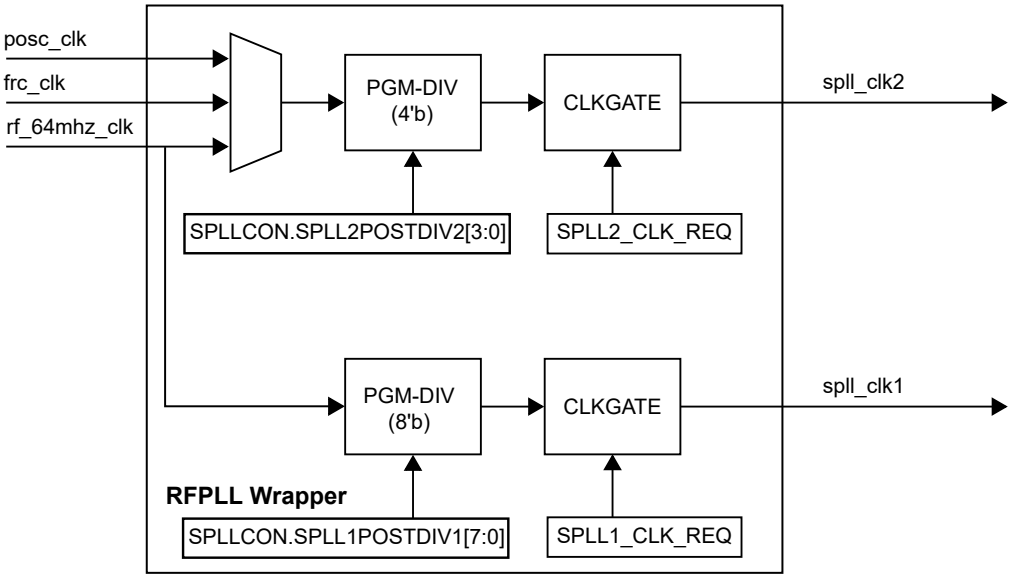


Figure 17-3. Peripheral Clock Generation (GCLK)

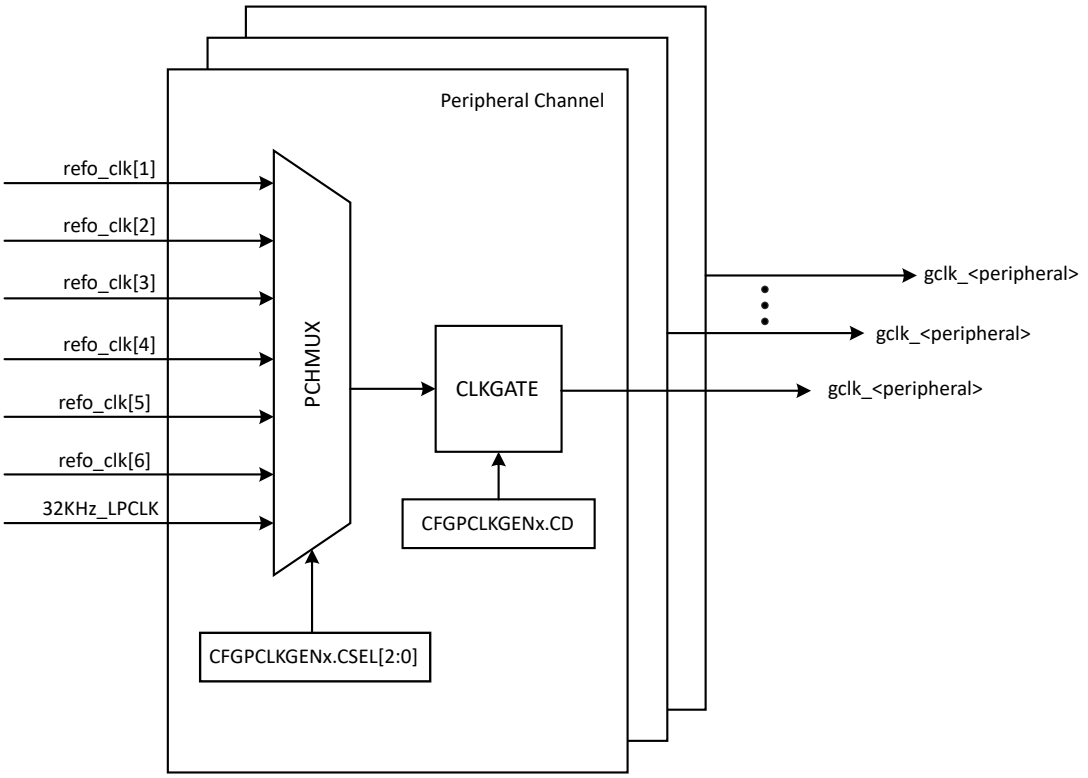
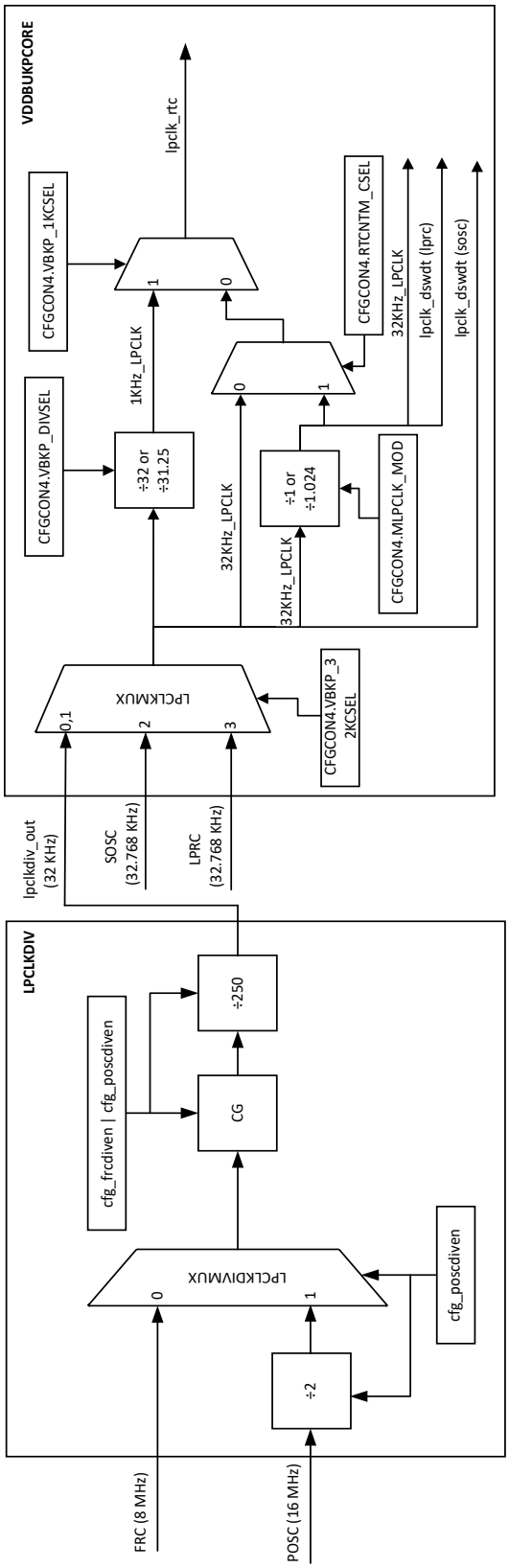


Figure 17-4. Low Power Clock Generation (LPCLK)



The CRU Master Clock Switch (MCS) selects which input clock is 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.

Related Links

23. Peripheral Module Disable (PMD)

17.3.2 Oscillators

17.3.2.1 Fast RC Oscillator (FRC)

The on-chip 8 MHz Fast RC Oscillator (FRC) is a fast, with precise frequency, internal RC oscillator. The FRC oscillator is accurate to provide the clock frequency, which is necessary to maintain the baud rate tolerance for serial data transmissions. Power-on Reset (POR) sets OSCCON.NOSC[3:0] = 0000; hence, the device starts with FRC when powered up.

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

17.3.2.1.1 Enabling the FRC

The FRC oscillator is powered in the following conditions:

- 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.
- When SPLL_CLK2 of ADC PMU Controller, Flash controller and 32KHz_LPCLK (32 KHz) request it as the source.

17.3.2.1.2 Frequency Tuning in User Mode

In addition to the factory calibration, the user can tune base frequency in their 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 OSCTUN.TUN[5:0] register bits.

17.3.2.2 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.

17.3.2.3 Primary Oscillator (POSC)

A 16 MHz ±20 ppm crystal/resonator oscillator or external clock is the primary oscillator for the 2.4 GHz RF transceiver. This is the input clock source for the system PLL, providing up to a 64 MHz clock.

17.3.2.4 Secondary Oscillator (SOSC)

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

The following are the features of the secondary oscillator:

- 32.768 kHz Operation
- Provides System Clock (SYS_CLK) Output
- Provides Source to CRU or LPCLKGEN on Request
- Provides Clock for the Low Power Mode

17.3.3 System and Peripheral Bus Clock Generation (CLKGEN)

The CLKGEN module generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB), as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains prescalers for the CPU and bus clocks.

There are two types of clocks generated by this module, called system clocks and peripheral clocks:

- The system clock (SYS_CLK used by the CPU supports components such as memory subsystems and fast peripherals.
- The peripheral bus clocks (PB1_CLK, PB2_CLK and PB3_CLK) are used to clock slow peripheral devices attached to the pb_bus.

The PBx_CLK 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 bus clocks are stopped when in Sleep mode. The clocks are restarted by disabling the sleep enable.

17.3.4 FRC Divider (FRCDIV)

The OSCCON.FRCDIV[2:0] register enables the division of the Fast RC (FRC) oscillator, allowing it to be used as a system clock and REFO clock through divider settings. The divisor is configured for eight divider selections: /1, /2, /4, /8, /16, /32, /64, /256.

17.3.5 RFPLL Wrapper

The output of RFPLL wrapper is used as a system clock or REFO clock source. Selection of the system clock source is performed with the OSCCON.NOSC[3:0] register field.

The RFPLL wrapper generates two clocks:

- SPLL_CLK1 (PGM MHz)
  - Clock frequencies = 64 MHz/(1-255) frequency choices
  - Clock ready indication
- SPLL_CLK2 (PGM MHz)
  - Clock frequencies = 64 MHz/(1-15) and optional clock disable option
  - Clock ready indication

Clocks are produced only when the consumer generates a request; CRU is the consumer for SPLL_CLK1 (SPLL_CLK1 is chosen through OSCCON.NOSC[3:0]) and ADC charge pump is the consumer for SPLL_CLK2, (SPLLCON.SPLL_BYP[1:0]). Along with the clocks, individual clock ready is also generated, which indicates that clocks are ready for consumption.

17.3.6 Reference Clock (REFO_CLK) Generation

The reference clock generator module uses multiple clock sources as input source and generates six different reference clock outputs.

The REFOxCON registers are used to configure the input clock source and divisor.

The clock sources for the reference clock generator:

- 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)
- 64 MHz system PLL (RFPLL PGM MHz, SPLL_CLK1, SPLL_CLK3)
- REFI pin
  - External clock input (REFI) is provided on anyone of the supported I/O pins. For details on supported input pins, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

The following table lists the clock's source mapping for both the CLKGEN generator and REFO_CLK generator.

Table 17-1. CRU Source and Output Clock Mapping

Clock Source	CLKGEN Selection (OSCCON.NOSC[3:0])	REFO Selection (REFOxCON.ROSEL[3:0])
FRC	0000	0000
SPLL_CLK1	0001	0001
POSC (16 MHz)	0010	0010
SOSC	0011	0011
LPRC	0100	0100
SPLL_CLK3 (RFPLL, 64 MHz)	—	0101
PB1_CLK	—	0110
SYS_CLK	—	0111
REFI Pin	—	1000

Related Links

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

17.3.7 Peripheral Clock Generation (GCLK)

All six reference clock generator outputs are given as input for GCLK generator module for peripheral clock generation. The GCLK module provides selection among following clocks:

- REFO1 to REFO6 clocks
  Note: Only REFO1 – REFO4 can get routed to chip IOs.
- Low-power clock (32KHz_LPCLK) (either LPRC, SOSC or 32 KHz clock derived from POSC/FRC)

The GCLK generator provides the Generic Clocks (GCLK_<Periperhal>) for system peripherals via peripheral channels. There are a total of 26 peripheral channels with the mapping as shown in the following table. The peripheral channels are fixed configuration and mapped in the CFGPCLKGENx register. CFGPCLKGENx dictates the peripheral clock selection and enables the clock for a specific peripheral.

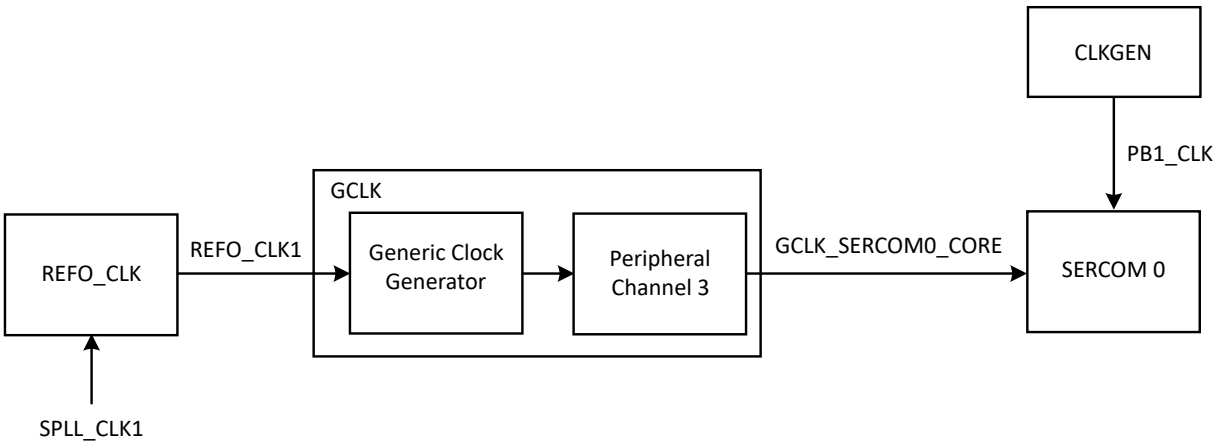
Table 17-2. Peripheral Clock Generation

Peripheral Clock	Channel Index	Position in CFGPCLKGENx
GCLK_EIC, GCLK_CCL	0	0
GCLK_FREQM_MSR	1	2
GCLK_FREQM_REF	2	1
GCLK_SERCOM0_CORE, GCLK_SERCOM1_CORE	3	3
GCLK_SERCOM2_CORE	4	4
GCLK_TC0	5	24
GCLK_TC1	6	25
GCLK_TC2, GCLK_TC3	7	26
GCLK_TC4, GCLK_TC5	8	27
GCLK_TC6, GCLK_TC7	9	28
GCLK_EVSYS_CH_0	10	8
GCLK_EVSYS_CH_1	11	9
GCLK_EVSYS_CH_2	12	10
GCLK_EVSYS_CH_3	13	11
GCLK_EVSYS_CH_4	14	12
GCLK_EVSYS_CH_5	15	13

.....continued		
Peripheral Clock	Channel Index	Position in CFGPCLKGENx
GCLK_EVSYS_CH_6	16	14
GCLK_EVSYS_CH_7	17	15
GCLK_EVSYS_CH_8	18	16
GCLK_EVSYS_CH_9	19	17
GCLK_EVSYS_CH_10	20	18
GCLK_EVSYS_CH_11	21	19
GCLK_TCC0	22	21
GCLK_TCC1, GCLK_TCC2	23	5
GCLK_AC	24	20
GCLK_CM4_TRACE	25	7

The following figure illustrates an example, where SPLL_CLK1 clocks the SERCOM0. The SPLL_CLK1 is input to the REFO generator. The Generic Clock Generator uses the REFO_CLK1 as its clock source and feeds into Peripheral Channel 3. The Generic Clock channel 3, also called GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface is clocked by PB1_CLK bus clock.

Figure 17-5. Example of SERCOM0 Clock



17.3.8 LPCLK Divider

The low-power clock divider module provides the clock source for low-power domain (VDDBUKUPCORE) modules. There are two sources of sleep clock sources, such as 32 KHz and 32.768 KHz clock. These clock sources are either from LPRC, SOSC or derived from POSC/FRC modules, such as RTCC, which requires 32.768 KHz in the RTC mode; whereas, the Bluetooth link controller requires a 32 KHz clock to maintain the Bluetooth clock in the Standby Sleep mode. Therefore, if the LPCLK source is 32 KHz, the RTC divider must be 31.25, which the user can select using CFGCON4.VBKP_DIVSEL. If the LPCLK source is 32.768 KHz, program CFGCON4.MLPCLK_MOD to divide it by 1.024. See *Power Management Unit (PMU)* for low power mode from Related Links.

Related Links

18. Power Management Unit (PMU)

17.3.9 Start-Up Considerations

The presence of hardware NVR configuration fuses on the PIC32CX-BZ3 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.

- When

  the device is active, the user can change the primary system clock source from FRC to SPLL by using the OSCCON register.

17.3.10 Fail-Safe Clock Monitor (FSCM)

The clock system includes a Fail-Safe Clock Monitor (FSCM). The FSCM monitors the SYS_CLK for continuous operation. If the SYS_CLK fails, 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. If executing the NMI, software can restart the main oscillator or shut down the system.

The SYS_CLK and the FSCM halt in the Sleep modes to prevent FSCM detection.

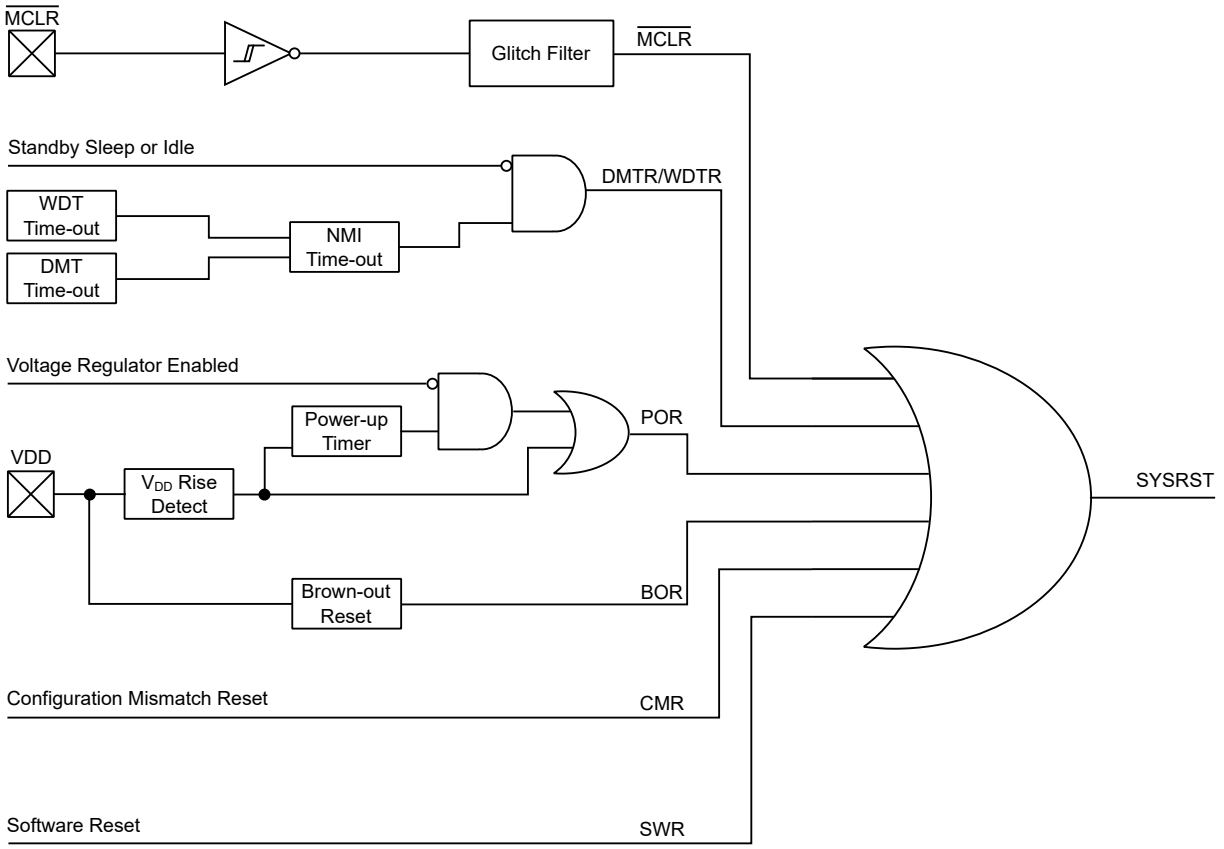
17.4 Resets

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}, I/O, 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)
- 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 (SYSRST) signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state.

Figure 17-6. CRU-System Reset Block Diagram



17.4.1 Control Registers

Most types of device resets set its corresponding status bits in the RCON register to indicate the type of Reset. The one exception is for the Non-Maskable Interrupt (NMI) time-out Reset. A POR clears all RCON bits, except the BOR and POR bits (RCON[1:0]), which are set. The user software can 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 does not allow system Reset.

The RCON register 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.

The system clock begins after a delay equal to the duration of the value of RNMICON.NMICNT 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 resets at the end of the interval. The RNMICON.NMICNT value can be set to zero for no delay and up to 255 SYS_CLK cycles.

The user can also trigger the NMI interrupt by setting the RNMICON.SWNMI bit in software or if the RNMICON.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

17.4.2 Modes of Operation

17.4.2.1 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 ($\overline{MCLR}$)(EXTR)
- Watchdog Time-out Reset (WDTO)
- Deadman Timer Reset (DMTR)
- Software Reset (SWR)
- Configuration Mismatch Reset (CMR)

17.4.2.1.1 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 initiating a new POR. 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.

When the POR event expires, but the device Reset is still asserted while the device configuration settings load and the clock oscillator sources configure, the clock monitoring circuitry waits for the oscillator source to become stable. The clock source of this device when exiting from Reset is always OSCCON.NOSC.

After these delays expire, the System Reset, SYSRST, is de-asserted. Before allowing the CPU to start code execution, eight system clock cycles (SYS_CLK) are required before deasserting the synchronized Reset to the CPU core. When the device is active, the user can change the primary system clock source from FRC to SPLL by using the OSCCON register

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.

Related Links

43. Electrical Characteristics

17.4.2.1.2 Master Clear Reset (EXTR)

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

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

EXTR is not a true POR. The user can configure the $\overline{MCLR}$ pin to generate a POR event by configuring the CFGCON1.SMCLR bit rather than an EXTR event.

17.4.2.1.3 Software Reset (SWR)

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

An SWR is performed as follows:

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

Setting the SWRST bit (RSWRST[0]) will arm the SWR. The subsequent read of the RSWRST register triggers the SWR, which will 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 is placed after the `READ` instruction.

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

17.4.2.1.4 Watchdog Timer Reset (WDTR)

A WDTR event is synchronized with the system clock (SYS_CLK), before asserting the SYSRST.

The RNMICON.WDTR flag is going to be set if there is a WDT event when in the CPU Run mode. The device Reset happens after the NMI counter expires and RCON.WDTO flag is set.

A WDT timeout during the Standby Sleep or Idle mode is going to wake up the processor. The WDTR flag will be set if there is a WDT event. The RNMICON.WDTS flag will be set if there is a WDT event during the Standby Sleep/Idle mode (WDTS). The WDTS flag triggers the NMI interrupt, but does not start the NMI counter, nor cause a device Reset. There is no affect on the RCON.WDTO bit. See *Power Management Unit (PMU)* for power modes from Related Links.

Related Links

18. Power Management Unit (PMU)

17.4.2.1.5 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 generates a BOR event, which is synchronized with the system clock, SYS_CLK, before asserting the SYSRST. The BOR flag bit (RCON[1]) captures this event. For more information, see *Electrical Characteristics* from Related Links.

Related Links

43. Electrical Characteristics

17.4.2.1.6 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 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.

17.4.2.1.7 Deadman Timer Reset (DMTR)

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

The primary function of the DMTR 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 Standby Sleep mode.

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

In general, a DMTR is useful in mission-critical and safety-critical applications, where it must detect any single failure of the software functionality and sequencing. For more information on the DMTR, see *Deadman Timer (DMT)* from Related Links.

Related Links

20. Deadman Timer (DMT)

17.4.2.2 Non-Maskable Interrupt (NMI)

The NMI timer provides a delay between DMT or WDT events and a device Reset. The delay set in the System Clock counts from 0-255 in the NMICNT[15:0] bits (RNMICON[15:0]). If these bits are set to '0', there is no delay between the RNMICON.DMTO or RNMICON.WDTR 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 in RNMICON is not cleared 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, then it waits for another NMI event to occur. In this case, a device Reset is not asserted and the software can return from this interrupt.

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

The RNMICON.WDTR flag will be set if there is a WDT event. The device will be reset after the NMI counter expires.

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

The Fail-Safe Clock Monitor (FSCM) sets the RNMICON.CF bit in case of a clock failure. The CF flag triggers the NMI interrupt but does not start the timer nor cause a Reset.

The RNMICON.SWNMI bit can be set in software to cause an NMI interrupt but does not start the NMI counter nor cause a Reset.

17.4.2.3 Determining the Source of Device Reset

After a device Reset, examine the RCON register to confirm the source of the Reset. Clear all Reset Status bits in the RCON register after reading them to ensure the RCON value provides meaningful results after the next device Reset.

17.4.3 Effects of Various Resets

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

Table 17-3. Status Bits, Their Significance and Initialization Condition for RCON Register(1)

Condition	EXTR	SWR	WDTO	DMTO	STANDBY SLEEP(2)	IDLE (2)	CMR	BOR	POR
Power-on Reset or $\overline{\text{MCLR}}$ set as POR	0	0	0	0	0	0	0	1	1

.....continued

Condition	EXTR	SWR	WDTO	DMTO	STANDBY SLEEP(2)	IDLE (2)	CMR	BOR	POR
Brown-out Reset	0	0	0	0	0	0	0	1	*
$\overline{MCLR}$ Reset during the Run mode	1	*	*	*	*	*	*	*	*
$\overline{MCLR}$ Reset during the Idle mode	1	*	*	*	*	1	*	*	*
$\overline{MCLR}$ Reset during the Standby Sleep mode	1	*	*	*	1	*	*	*	*
Software Reset command	*	1	*	*	*	*	*	*	*
Configuration Word mismatch Reset	*	*	*	*	*	*	1	*	*
WDT Time-out Reset during the Run mode and NMI counter expires	*	*	1	*	*	*	*	*	*
WDT Time-out Reset during the Idle mode	*	*	*	*	*	1	*	*	*
WDT Time-out Reset during the Standby Sleep mode	*	*	*	*	1	*	*	*	*
DMT Time-out Reset and NMI counter expires	*	*	*	1	*	*	*	*	*
Interrupt exit from the Idle mode	*	*	*	*	*	1	*	*	*
Interrupt exit from the Standby Sleep mode	*	*	*	*	1	*	*	*	*

1. Legends: * = unchanged
2. The device enters the Standby Sleep or Idle states when it executes a WAIT (WFI) instruction along with the sleep sequence. See *Power Management Unit (PMU)* for Sleep/Idle modes from Related Links.

Related Links

18. Power Management Unit (PMU)

17.4.3.1 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

17.4.3. Effects of Various Resets

17.4.3.2 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.

17.4.3.3 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: Clear the status bits in the RCON register after reading them so that the next RCON register value after a device Reset is meaningful.

Table 17-4. 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
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])	$\overline{\text{MCLR}}$ 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

17.4.4 CRU Clock Configuration Registers

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.

17.5 Register Summary

See *CRU* module in the *Product Memory Mapping Overview* from Related Links for base address.

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

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		2SPDSLP					
		31:24						FRCDIV[2:0]		
0x04 ... 0x0F	Reserved									
0x10	OSCTUN	7:0			TUN[5:0]					
		15:8								
		23:16								
		31:24								
0x14 ... 0x1F	Reserved									
0x20	SPLLCON	7:0				SPLLFLOCK	SPLLPWDN			
		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 ... 0x5F	Reserved									
0x60	RNMICON	7:0	NMICNT[7:0]							
		15:8	NMICNT[15:8]							
		23:16	SWNMI				EXT	PLVD	CF	WDTS
		31:24							DMTO	WDTR
0x64 ... 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

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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
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									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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									
0x0120	REFO6TRIM	7:0								
		15:8								
		23:16	ROTRIM0							
		31:24	ROTRIM8	ROTRIM7	ROTRIM6	ROTRIM5	ROTRIM4	ROTRIM3	ROTRIM2	ROTRIM1
0x0124 ... 0x012F	Reserved									
0x0130	PB1DIV	7:0		PB1DIV[6:0]						
		15:8	PB1DIVON				PB1DIVRDY			
		23:16								
		31:24								
0x0134 ... 0x013F	Reserved									
0x0140	PB2DIV	7:0		PB2DIV[6:0]						
		15:8	PB2DIVON				PB2DIVRDY			
		23:16								
		31:24								
0x0144 ... 0x014F	Reserved									
0x0150	PB3DIV	7:0		PB3DIV[6:0]						
		15:8	PB3DIVON				PB3DIVRDY			
		23:16								
		31:24								
0x0154 ... 0x015F	Reserved									
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 ... 0x016F	Reserved									
0x0170	CLKSTAT	7:0			SPLL3RDY	LPRCRDY	SOSCRDY	POSCRDY	SPLL1RDY	FRCRDY
		15:8								
		23:16								
		31:24								
0x0174 ... 0x018F	Reserved									
0x0190	CLK_DIAG	7:0			SPLL3_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

Related Links

6.4.1.9. CLR, SET and INV Registers
8. Product Memory Mapping Overview

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 the user can only write them by disabling the peripheral. Enable protection is denoted by the Enable-Protected property in each individual 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

17.6.1 CRU Oscillator Control

Name: OSCCON
Offset: 0x00
Reset: 0x00200003

Note: Perform the system unlock sequence before writing this register.

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		2SPDSLP					
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 is divided by 1 (default value)
001	FRC is divided by 2
010	FRC is divided by 4
011	FRC is divided by 8
100	FRC is divided by 16
101	FRC is divided by 32
110	FRC is divided by 64
111	FRC is divided by 256

Bit 23 – DRMEN Enable Dream Mode bit

Value	Description
1	When the SLEEP/WAIT (WFI) instruction is executed and SLPEN = 1, DMA transfer complete causes the device to enter the Sleep mode.
0	DMA transfer has no effect

Bit 21 – 2SPDSLP 2-Speed start-up enabled in the Standby Sleep mode bit
Note: CFGCON2.WAKE2SPD specifies the default Reset value.

Value	Description
1	When the device exits the Standby Sleep mode, the SYS_CLK is going to be from FRC until the selected clock is ready.
0	When the device exits the Standby Sleep mode, the SYS_CLK is going to be from the selected clock.

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

Notes:

- Default value on reset is 4'b0000
- 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 (SPLL_CLK1 module) (input clock is 64 MHz from RFPLL wrapper 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.

Value	Description
0000	Fast RC oscillator (FRC) divided by OSCCON.FRCDIV
0001	System PLL Clock-1 (SPLL_CLK1 module) (input clock is 64 MHz from RFPLL wrapper 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:

- When 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, OSCTUN, SPLLCON, PBxDIV
0	Clock and PLL selection registers are not locked; configurations can be modified.

Bit 4 – SLPEN Enable Sleep Mode bit

Value	Description
1	When a WAIT Instruction is executed, the device enters the Standby Sleep mode.
0	When a WAIT instruction is executed, the device enters the IDLE mode.

Bit 3 – CF Clock Fail Detect bit (Read/Writable/Clearable by application)

Notes:

- Writing '1' to this bit initiates the clock-switching sequence by the clock switch state machine.
- A Reset occurs when the clock switch state machine initiates a valid clock switching sequence.
- This bit is set when the clock fail event is detected.

Value	Description
1	FSCM detected a clock failure
0	FSCM did not detected a clock failure

Bit 1 – SOSCEN Low Power Secondary Oscillator Enable bit

Note: Set the value specified by the SOSCEN configuration bits in CFGCON2.SOSCSEL on RESET.

Value	Description
1	Enable Secondary Oscillator
0	Disable Secondary Oscillator

Bit 0 – OSWEN Oscillator Switch Enable bit

Notes:

- Writing '0' to this bit has no effect.
- Hardware clears this bit after a successful clock switch.
- Hardware clears this bit after a redundant clock switch (NOSC = COSC).
- Hardware clears this bit after FSCM switches the oscillator to the fail-safe clock source.

Value	Description
1	Requests the oscillator switch to select as specified by NOSC[3:0] bits
0	Oscillator switch is complete

17.6.2 CRU Oscillator Trimming

Name: OSCTUN
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								
Reset								

Bit	7	6	5	4	3	2	1	0
			TUN[5:0]					
Access			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

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

17.6.3 SPLL Control

Name: SPLLCON
Offset: 0x20
Reset: 0xC0010028

Note: Perform the system unlock sequence before this register is 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
				SPLLFLOCK	SPLLPWDN			
Access				R/W/L	R/W/L			
Reset				0	1			

Bits 31:30 – SPLL_BYP[1:0] SPLL Bypass; SPLL_CLK2 clock source selection

Notes:

- Dictates the clock source for ADC CP (Analog-to-Digital Converter Charge Pump) (SPLL_CLK2) Clock generation only
- Preselect the clock sources and keep them ready before the need of ADC CP arrives. Failure to do so results in 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 (SPLL2) Post Divide Value

Value	Description
1 ≤ SPLLPOSTDIV2 ≤ 15	Divide by SPLL2POSTDIV2
0	No Clock; Clock disabled

Bits 15:8 – SPLL1POSTDIV1[7:0] SPLL1 Post Divide Value

Value	Description
2 ≤ SPLLPOSTDIV ≤ 255	Divide by SPLL1POSTDIV1
0,1	Divide by 1

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

17.6.4 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

- 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.

This bit is set by hardware at detection of an I/O POR event. User software must clear this bit to view the next detection.
Note: Writing '1' to this bit does not cause POR_IO reset.

Bit 30 – POR_CORE Core Voltage POR Flag bit

- 1 = A Power-on Reset has occurred due to core voltage.
- 0 = A Power-on Reset has not occurred due to core voltage.

This bit is set by hardware at detection of a core POR event. User software must clear this bit to view the next detection.
Note: Writing '1' to this bit does not cause POR_IO reset.

Bit 27 – BCFGERR BCFG Error Flag bit

- 0 = A BCFG error has not occurred.
- 1 = A BCFG error has occurred.

This bit is set when a primary BCFG value has an error but the secondary BCFG value is valid and used.

Bit 26 – BCFGFAIL BCFG Failure Flag bit

- 0 = A BCFG failure has not occurred.

- 1 = A BCFG failure has occurred.

This bit is set when both the Primary and Secondary BCFG values has an unrecoverable error. Only the default values are in effect.

Bit 25 – NVMLTA NVM Life Time Alert Flag bit

- 0 = A NVM LTA error has not occurred.
- 1 = A NVM LTA error has occurred.

This bit is set due to charge leakage, and the NVM (Flash) is nearing EOL.

Bit 24 – NVMEOL NVM End of Life Flag bit

- 0 = A NVM EOL failure has not occurred.
- 1 = A NVM EOL failure has occurred.

This bit may not be visible to the user because the part does not come out of Reset if the bit is asserted.

Bit 16 – VBAT VBAT Mode Flag bit

- 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

- 1 = Deep Sleep mode has occurred.
- 0 = Deep Sleep mode has not occurred.

This bit is set by hardware at the time of entry into the Deep Sleep mode. User software must clear this bit to view next detection.

Bit 9 – CMR Configuration Mismatch Reset Flag bit

- 1 = A CMR event has occurred.
- 0 = A CMR event has not occurred.

Note: Writing '1' to this bit does not cause Mismatch Reset.

Bit 7 – EXTR External Reset $\overline{\text{MCLR}}$ Status bit

- 1 = A Master Clear (pin) Reset has occurred.
- 0 = A Master Clear (pin) Reset has not occurred.

Note: Writing '1' to this bit does not cause a ($\overline{\text{MCLR}}$).

Bit 6 – SWR Software Reset Flag bit

- 1 = A SWR has occurred.
- 0 = A SWR has not occurred.

Note: Writing '1' to this bit does not cause SWR.

Bit 5 – DMTO Deadman Timer Time-out Flag bit

- 1 = DMT Time-out has occurred and caused a Reset.
- 0 = DMT Time-out has not occurred.

Note: Writing '1' to this bit does not cause DMT Reset.

Bit 4 – WDTO Watchdog Timer Time-Out Flag bit

- 1 = WDT Time-out has occurred and caused a Reset.
- 0 = WDT Time-out has not occurred.

Note: Writing '1' to this bit does not cause WDTR.

Bit 3 – SLEEP Wake from Standby Sleep Flag bit

- 1 = Device has been in Standby Sleep mode.
- 0 = Device has not been in Standby Sleep mode.

Note: Writing '1' to this bit does not invoke the Standby Sleep mode.

Bit 2 – IDLE Wake from Idle Flag bit

- 1 = Device was in Idle mode.
- 0 = Device was not in Idle mode.

Note: Writing '1' to this bit does not invoke the Idle mode.

Bit 1 – BOR BOR Flag bit

- 1 = A BOR occurred.
- 0 = A BOR did not occur.

This bit is set by hardware at detection of a BOR event. User software must clear this bit to view the next detection.
Note: Writing '1' to this bit does not cause a BOR.

Bit 0 – POR POR Flag bit

- 1 = A Power-on Reset occurred.
- 0 = A Power-on Reset did not occur.

This bit is set by hardware at detection of a POR event. User software must clear this bit to view the next detection.
Note: Writing '1' to this bit does not cause a POR.

17.6.5 Software Reset Register

Name: RSWRST
Offset: 0x40
Reset: 0x00000000
Property: -

Note: The system unlock sequence must be done before writing this register. For more details, see *Register Locking* 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
Access								
Reset								

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

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'.

Related Links
22.6. Register Locking

17.6.6 NMI Control Register

Name: RNMICON
Offset: 0x60
Reset: 0x00000000
Property: -

Note: The system unlock sequence must be done before writing this register. See *System Configuration and Register Locking (CFG)* from Related Links.

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 causes a Reset when NMICNT expires)

- 1 = DMT Time-out occurred and caused an NMI.
- 0 = DMT Time-out did not occur.

Note: Writing '1' to this bit cause a user-initiated DMT NMI event and NMICNT start.

Bit 24 – WDTR Watchdog Timer Time-Out in Run Flag bit

- 1 = WDT Time-out occurred and caused an NMI (this may cause a Reset if NMICNT expires).
- 0 = WDT Time-out did not occur.

Note: Writing '1' to this bit cause a user initiated WDT NMI event and NMICNT start.

Bit 23 – SWNMI Software NMI Trigger bit

- 1 = Writing '1' to this bit generates an NMI.
- 0 = Writing '0' to this bit has no effect.

Bit 19 – EXT External / Generic NMI Event bit

- 1 = A general NMI event was detected and caused an NMI.
- 0 = A general NMI event was not detected.

Note: Writing '1' to this bit cause a user initiated EXT NMI event.

Bit 18 – PLVD Programmable Low Voltage Detect Event bit

- 1 = PLVD detected a low voltage condition and caused an NMI.
- 0 = PLVD did not detect a low voltage condition.

Note: Writing '1' to this bit causes a user-initiated PLVD NMI event.

Bit 17 – CF Clock Fail Detect bit (Read/Clear-able by application)

- 1 = FSCM detected clock failure and caused an NMI.
- 0 = FSCM did not detect clock failure.

Note: Writing '1' to this bit causes a user-initiated clock failure NMI event but does not cause a clock switch.

Bit 16 – WDTS Watch-Dog Timer Time-out in Standby Sleep Flag bit

- 1 = WDT Time-out occurred during the Standby Sleep mode and caused a wake-up from sleep.
- 0 = WDT Time-out did not occur during the Standby Sleep mode.

Note: Writing '1' to this bit causes a user-initiated WDT NMI event.

Bits 15:0 – NMICNT[15:0] NMI Reset counter value bit

This bit field specifies the reload value used by the NMI Reset counter. The following events generate an NMI event and/or reset when the NMI_CNT expires on:

- WDT event
- DMT event

Values	Description
0x0000	No delay between NMI assertion and device Reset event.
0x0001	—
0x0002	—
.....	—
.....	—
.....	—
0xFFFE	—
0xFFFF	Number of SYSCLK cycles that the software has to clear the NMI event before a device Reset is performed. If the NMI event is cleared before the counter reaches zero, no device Reset is asserted.

Related Links

22. System Configuration and Register Locking (CFG)

17.6.7 Reference Oscillator x Control

Name: REFOxCON
Offset: 0x70 + (x-1)*0x20 [x=1..6]
Reset: 0x00000000

Notes:

- Do not write REFOCON.ROSEL while the REFOCON.ACTIVE bit is '1'. This results in undefined behavior.
- Do not write REFOCON when REFOCON.ON != REFOCON.ACTIVE. This results in undefined behavior.
- This register can always be accessed regardless of the SYSKEY 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.
For example, period of REFO_CLK = [Reference source * 2] * RODIV.

Value	Description
0x7FFF	REFOx clock is Base clock frequency divided by 65,534 (32,767 *2)
0x7FFE	REFOx clock is Base clock frequency divided by 65,532 (32,766 * 2)
...	
...	
...	
0x0003	REFOx clock is Base clock frequency divided by 6 (3*2)
0x0002	REFOx clock is Base clock frequency divided by 4 (2*2)
0x0001	REFOx clock is Base clock frequency divided by 2 (1*2)
0x0000	REFOx clock is same frequency as Base clock (no divider)

Bit 15 – ON Output Enable bit

Value	Description
1	Enables the Reference Oscillator Module
0	Disables the Reference Oscillator Module

Bit 14 – FRZ Freeze in Debug mode bit

Value	Description
1	When emulator is in the Debug mode, module freezes operation
0	When emulator is in the Debug mode, module continues operation

Bit 13 – SIDL Peripheral Stop in Idle Mode bit

Value	Description
1	Discontinue 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 Standby Sleep bit

Note: This bit is ignored when ROSEL[3:0] = (0000 or 0001).

Value	Description
1	Reference Oscillator output continues to run in Standby Sleep
0	Reference Oscillator output is disabled in Standby Sleep

Bit 9 – DIVSW_EN Clock RODIV/ROTRIM switch enabled

Value	Description
1	Clock Divider Switching is in progress
0	Clock Divider Switch is 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, SYS_CLK (reference clock reflects any device clock switching)
0110	Peripheral clock, PB1_CLK (reference clock reflects any peripheral clock switching)
0101	System PLL (Clock-3), SPLL_CLK3
0100	LPRC
0011	SOSC
0010	POSC
0001	System PLL (Clock-1), SPLL_CLK1
0000	FRC

17.6.8 Reference Oscillator x Trim

Name: REFOxTRIM
Offset: 0x80 + (x-1)*0x20 [x=1..6]
Reset: 0x00000000

Notes:

- Do not write REFOxTRIM when REFOxCON.ON != REFOxCON.ACTIVE. This results in undefined behavior.
- This register can always be accessed regardless of the SYSKEY 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 is added to RODIV value
0000_0000_1	1/512 (0.001953125) divisor is added to RODIV value
0000_0001_0	2/512 (0.00390625) divisor is added to RODIV value
...	...
...	...
100000000	256/512 (0.5000) divisor is added to RODIV value
...	...
...	...
1111_1111_0	510/512 (0.99609375) divisor is added to RODIV value
1111_1111_1	511/512 (0.998046875) divisor is added to RODIV value

17.6.9 PBx Clock Divisor Control

Name: PBxDIV
Offset: 0x0130 + (x-1)*0x10 [x=1..3]
Reset: 0x00008800

Note: Perform the system unlock sequence before this register can be written. The Reset value for PB3DIV[6:0] is 0x09.

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
	PBxDIVON				PBxDIVRDY			
Access	R				R			
Reset	1				1			

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

Bit 15 – PBxDIVON (x=1 to 3) Output Enable bit

Value	Description
1	Enables PBx Output clock
0	Disables PBx Output clock **Note:** Do not write PB1DIV.PB1DIVON bit as '0', as the CRU system uses this clock, and it is one of the source for REFO.

Bit 11 – PBxDIVRDY (x=1 to 3) PBx Peripheral Clock Divisor Ready

Value	Description
1	Indicates the PB clock divisor logic is not switching divisors and the PBxDIV may be written.
0	Indicates the PB clock divisor logic is currently switching values and the PBxDIV cannot be written.

Bits 6:0 – PBxDIV[6:0] (x=1 to 3) PBx Peripheral Clock Divisor Control value

Value	Description
000_0000	Divide by 1 PBx Clock same frequency as SYS_CLK
000_0001	Divide by 2 PBx Clock is 1/2 of SYS_CLK
000_0010	Divide by 3 PBx Clock is 1/3 of SYS_CLK
000_0011	Divide by 4 PBx Clock is 1/4 of SYS_CLK
. . .	...
. . .	...
000_1111	Divide by 16 PBx Clock is 1/16 of SYS_CLK
001_0000	Divide by 17 PBx Clock is 1/17 of SYS_CLK
. . .	...
. . .	...
111_1110	Divide by 127 PBx Clock is 1/127 of SYS_CLK
111_1111	Divide by 128 PBx Clock is 1/128 of SYS_CLK

17.6.10 Slew Rate Control for Clock Switching

Name: SLEWCON
Offset: 0x160
Reset: 0x00000000

Notes:

- Perform the system unlock sequence before this register is written.
- Updates to this register do not take affect 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 input, cfg_slewcon_sel[] defines the reset value of this register field.

Value	Description
0000	1 clock is generated at each slew step
0001	2 clocks is generated at each slew step
. . .	...
1111	16 clocks are 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 are used when doing slewed clock switches
Note: Each Divisor step lasts for four clocks

Value	Description
000	No Divisor is used
001	Divide by 2 (2^1), then no divisor is used
010	Divide by 4 (2^2), then by 2, then no divisor is used
011	Divide by 8 (2^3), then by 4, then by 2, then no divisor is used

Value	Description
100	Divide by 16 (2^4), then by 8, then by 4, then by 2, then no divisor is used
...	...
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 is used

Bit 2 – SLW_UP Enables clock slew for switching up to faster clocks

Value	Description
0	Disables Clock Slewing
1	Enables Clock Slewing on a clock switch or exits from the Standby Sleep mode

Bit 1 – SLW_DN Enables clock slew for switching down to slower clocks

Value	Description
0	Disables Clock Slewing
1	Enables Clock Slewing 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

17.6.11 Clock Status

Name: CLKSTAT
Offset: 0x170
Reset: 0x00000000

Note: The corresponding RDY bits are updated only after 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 5 – SPLL3RDY System PLL (Clock-3) Ready Status value

Value	Description
1	SPLL_CLK3 is stable and ready
0	SPLL_CLK3 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	SPLL_CLK1 is stable and ready
0	SPLL_CLK1 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

17.6.12 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	SPLL1_STOP	LPRC_STOP	FRC_STOP	SOSC_STOP	POSC_STOP
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			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 internal NMI counter.

Bit 5 – SPLL3_STOP SPLL Clock Stop Control value

Note: This gating logic is outside the CRU module.

Value	Description
0	SPLL_CLK3 clock source runs as normal
1	SPLL_CLK3 clock source is stopped

Bit 4 – SPLL1_STOP SPLL Clock Stop Control value

Note: This gating logic is outside the CRU module.

Value	Description
0	SPLL_CLK1 clock source runs as normal
1	SPLL_CLK1 clock source is stopped

Bit 3 – LPRC_STOP LPRC Clock Stop Control value

Note: This gating logic is outside the CRU module.

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: This gating logic is outside the CRU module.

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: This gating logic is outside the CRU module.

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: This gating logic is outside the CRU module.

Value	Description
0	POSC clock source runs as normal
1	POSC clock source is stopped

18. Power Management Unit (PMU)

18.1 Overview

This section describes the Power Management Unit (PMU) in the PIC32CX-BZ3 wireless SoC with the various power saving modes. The PMU controls the sleep modes and the power domain gating of the device. Various sleep modes are provided to fit power consumption requirements. This allows the PMU to disable unused modules to save power. In the Active mode, the CPU is executing application code. When the device enters the Sleep mode, program execution is stopped and some modules and clock domains are automatically switched off according to the Sleep mode. The application code helps the device to choose the type and timing of the Sleep mode to enter. Interrupts from enabled peripherals and all enabled Reset sources can restore the device from the Sleep mode to Active mode.

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 SDK and operational stacks provided by Microchip handle the start-up and operation of the PMU. Therefore, Microchip highly recommends using this software framework for all application development on the device.

18.2 Features

- Low Power Modes: Idle, Standby Sleep, Deep Sleep, Extreme Deep Sleep
- Sleep Walking/Dream Available in the Standby Sleep Mode
- FlexRAM (SRAM) Retention in the Standby Sleep, and Deep Sleep Mode
- I/O Lines Retention in the Deep Sleep Mode

18.3 Functional Description

18.3.1 Power Modes

The power modes and power management of different modules of the PIC32CX-BZ3 are shown in the following table.

Table 18-1. Power Modes and Power Management Features

Functions	Device Power Modes					
	Run	Idle	SleepWalking/Dream(1)	Standby Sleep	Deep Sleep	Extreme Deep Sleep
CPU	ON	Clock Gated	Clock Gated	Clock Gated	OFF	OFF
Peripherals	ON	Idle	On Demand	Clock Gated	OFF	OFF
Flash	ON	Idle	On Demand	Clock Gated	OFF	OFF
Core System Memory	ON	Idle	On Demand	Retention Mode	OFF	OFF
Radio	ON	Idle	On Demand	Clock Gated	OFF	OFF
DSWDT	ON	ON	ON	ON	ON	OFF
RTCC	ON	ON	ON	ON	ON	OFF
FlexRAM (SRAM)	ON	Idle	On Demand	Retention Mode	Retention Mode (On Demand)	OFF
XTAL(16 MHz)	ON	ON	ON	ON	OFF	OFF
FRC	ON	ON	On Demand	OFF	OFF	OFF
LPRC	ON	ON	ON	ON	ON	OFF
SPLL	ON	ON	On Demand	OFF	OFF	OFF
SOSC	ON	ON	ON	ON	ON	OFF

Note:

1. Dream (Sleep Walking) is not a mode by itself but is a companion mode to the Standby Sleep mode. This mode cannot be entered directly through a software command.

All transitions from the Run state to any of the low-power states is simply initiated by the WFI command from the CPU, however, prior to initiating the WFI command.

18.3.1.1 Operation

All power-saving modes are controlled by sub-systems: XDS/DS Controller, CRU, PMU Controller and Wireless Subsystem. To enter into the different low power modes, the software performs the following actions:

- Disabling all interrupts
- Setting up the DSCON.DSEN, OSCCON.SLPEN, OSCCON.DRMEN and Wireless Subsystem Sleep mode control bits
- 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 mode and service the interrupt
- If no pending interrupts, issue a SLEEP/WFI command from the CPU to get into the appropriate Low Power mode

The Low-Power modes entry and exit commands and wake-up resources are shown in the following table.

Table 18-2. Low-Power Saving Modes 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 + Standby Sleep mode	IRQ, Reset, Others
Standby 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, Reset
Extreme Deep Sleep	DSCON.DSEN + {RTCC (Disabled) and DSWDT (Disabled) and INT0 (Enabled)} + Wireless Sleep followed by WFI	INT0, Reset

Note:

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.

18.3.1.2 Run Mode

In the Run mode, the CPU is actively executing code. This mode provides normal operation of the processor and all peripherals that are currently enabled. On power-up, the device automatically gets into the Run mode. There are no special instructions required to get into the Run mode

18.3.1.3 Idle Mode

In the Idle mode, all active peripherals can be clocked but the CPU core is clock gated off and no code is executed. This mode can be useful for applications that only require the processor to run when an interrupt occurs.

The device enters the Idle mode, when the OSCCON.SLPEN bit is low and the CPU executes a WFI instruction.

Note: When exiting from Standby Sleep mode, the CRU transitions 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 the Run mode. Therefore, when exiting the Standby Sleep mode, both the RCON.SLEEP and RCON.IDLE bits are going to be set.

18.3.1.4 Dream Mode

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.

When the DRMEN bit in the OSCCON register is set, it allows the DMA controller to switch between the Idle mode and the Standby Sleep mode. When the OSCCON.SLPEN bit is set and the OSCCON.DRMEN bit is set, the CRU monitors the DMAC to ensure all transfers are complete before going into the Standby 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 Standby 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 wakes 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 go backs to the Standby 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 exits the Standby Sleep mode and transitions back to the Run mode.

18.3.1.5 Standby Sleep Mode

In the Standby Sleep mode, the FlexRAM (SRAM) is in the Retention mode while the CPU and most peripherals are clock gated off and no code is executed.

Standby Sleep Entry

The Standby 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').

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

Note: If software writes to the CRU SFR before going into the Standby Sleep mode, it is recommended to perform read on the SFR to flush the write before executing the WFI instruction that initiates the Standby Sleep mode.

Standby Sleep Exit

The device comes out of Standby Sleep mode and starts running with the FRC as the clock source and performs an automatic clock switch to the selected clock source.

18.3.1.6 Deep Sleep Mode

In the Deep Sleep mode, the FlexRAM (SRAM) is in the Retention mode (on demand) while the CPU and peripherals (see Table 18-2 for available peripherals as wake-up source) are OFF, and no code is executed.

Deep Sleep Mode Entry

The Deep Sleep mode is entered by performing the following steps:

1. Disable all interrupts.
2. Set Wireless Subsystem into the Low-Power mode.
3. Set the DSEN bit in the DSCON register.

4. Enable the Deep Sleep wake-up source (See Table 18-2).
5. Check for any pending interrupts and if present, abort Deep Sleep mode and service the interrupt.
6. If there are no pending interrupts, then issue a SLEEP/WFI command from the CPU.

To minimize the chance that Deep Sleep is spuriously entered, the SLEEP/WFI command must be issued as the next instruction following the setting of the DSCON.DSEN bit. This sequence can still be interrupted by interrupts and other system latencies but does not prevent the Deep Sleep mode being entered once the SLEEP/WFI command is executed. The DSEN bit is then automatically cleared when exiting the Deep Sleep mode.

Note: The DSWSRC register clears automatically when the DSEN bit is set, regardless of whether the Deep Sleep mode is actually entered or not. Therefore, software must read this entire register after exiting the Deep Sleep mode and before re-enabling the Deep Sleep mode.

Deep Sleep Mode Exit

The Deep Sleep mode exits on any of the following events:

1. Device exits Deep Sleep due to a wake-up event.
   a. POR event (de-assertion) on the VDD supply.
   b. DSWDT time-out (if DSWDT is enabled). When the DSWDT timer times out, the Deep Sleep mode will be
      exited.
   c. RTCC alarm (if RTCC is enabled).
   d. Assertion (0) of the ($\overline{MCLR}$) pin.
   e. Assertion of the INT0 pin (if the interrupt was enabled before the Deep Sleep mode was entered). The polarity configuration (refer to CFGCON0.INT0P) is used to determine the assertion level (0 or 1) of the pin that causes an exit from the Deep Sleep mode.
      Note: Any interrupts pending when entering the Deep Sleep mode are cleared, and exiting from the Deep Sleep mode requires a change on the INT0 pin while in the Deep Sleep mode.
2. The DSEN bit is automatically cleared.
3. Read the Deep Sleep Status bit and clear it.
4. Read the DSSEMA1 registers (optional).
5. When
   all state-related configuration is complete, clear the DSSR bit in the DSCON register.
6. Device releases all held logic and/or I/Os. Until this occurs, the control and data bits for the I/Os have no effects on the actual I/O state.
7. Software resumes normal operation.

18.3.1.6.1 Enabling Retention RAM in the Deep Sleep Mode

There is no separate backup SRAM available for the PIC32CX-BZ3 family of devices. Any context saving needs to be done in the data FlexRAM (SRAM). Depending on the size requirement of the data to be retained while in the Deep Sleep mode, the right size FlexRAM (SRAM) may be selected in the WCMSIZ register. See *WCMSIZ* from Related Links.

Related Links

18.5.2. WCMSIZ

18.3.1.6.2 Zero-Power BOR (ZPBOR)

The ZPBOR is a low-power BOR used during the Deep Sleep operation. The ZPBOR is enabled only when the CFGCON4.DSZPBOREN configuration bit is set as '1'. See *CFGCON4(L)* from Related Links.

Related Links

22.9.4. CFGCON4(L)

18.3.1.6.3 Deep Sleep Watchdog Timer (DSWDT)

The Deep Sleep Watchdog Timer is a configurable timer with a time-out range of 1 ms to over 24 days. The DSWDT is configured using the CFGCON4.DSWDTEN, CFGCON4.DSWDTPS[3:0] and CFGCON4.DSWDTLPRC. See *CFGCON4(L)* from Related Links. The DSWDT is a separate timer from the device's WDT that is used in the Run mode. The device's WDT does not have to be enabled for the DSWDT to function. Entry into the Deep Sleep mode automatically resets the DSWDT.

Related Links

22.9.4. CFGCON4(L)

18.3.1.7 Extreme Deep Sleep (XDS) Mode

In the Extreme Deep Sleep mode, INT0, which is the external interrupt pin, alone is active and can wake up the device. Exiting XDS is equivalent to POR. The RCON register provides the status whether it is a normal power up or exiting from XDS.

Extreme Deep Sleep Mode Entry

1. Disable all the interrupts except INT0 (as desired).
2. Disable RTCC, DSWDT and WCM Retention.
3. Set the DSEN bit in the DSCON register.
4. Check for the pending interrupts, then, if present, abort the Extreme Deep Sleep mode and service the interrupt.
5. If there are no pending interrupts, issue a WFI command from the CPU.
   To minimize the chance of entering the Extreme Deep Sleep mode spuriously, it is recommended that the WFI command be issued as the next instruction following the setting of the DSCON.DSEN bit. This sequence can still be interrupted by interrupts and other system latencies, but it does not prevent the system entering the Extreme Deep Sleep mode once the WFI command is executed.

   The DSEN bit is, then, automatically cleared when exiting the Extreme Deep Sleep mode.

Extreme Deep Sleep Mode Exit

The Extreme Deep Sleep mode exits on any of the following events:

1. POR event on the VDD supply.
2. Assertion of the ($\overline{MCLR}$) pin or INT0 pin.

18.4 Register Summary

See *PMU* module in the *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x53	Reserved									
0x54	WCMCFG	7:0								
		15:8		SRAM2_CFG[2:0]				SRAM1_CFG[2:0]		
		23:16								
		31:24								
0x58 ... 0x5F	Reserved									
0x60	WCMSIZ	7:0								
		15:8							SRAM1_SIZE[1:0]	
		23:16								
		31:24								

Related Links

8. Product Memory Mapping Overview

18.5 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.

18.5.1 WCMCFG Register

Name: WCMCFG
Offset: 0x54
Reset: 0x00001100
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
		SRAM2_CFG[2:0]				SRAM1_CFG[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	1		0	0	1

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 14:12 – SRAM2_CFG[2:0] CMCC Memory Configuration in the Standby Sleep Mode

Notes:

1. This field is only writable when CFGCON0.PMULOCK = 0.
2. These bits are only applicable for controlling the state of memory in the Standby Sleep mode.
3. Memories cannot be completely turned off dynamically in the PIC32CX-BZ3 (unless in DS and CLDO is OFF). Therefore, option '000' is unavailable.
4. The power consumption of these modes are: ON > NAP > RET > RET+NAP.

Value	Description
1xx	CMCCRAM in the ON mode
011	CMCCRAM in the NAP mode
010	CMCCRAM in the RET mode
001	CMCCRAM in the RET + NAP mode
000	CMCCRAM is powered OFF (Option is unavailable)

Bits 10:8 – SRAM1_CFG[2:0] FLEXRAM (SRAM) Configuration in the Standby Sleep Mode

Notes:

1. This field is only writable when CFGCON0.PMULOCK = 0.
2. These bits are only applicable for controlling the state of memory in the Standby Sleep mode.
3. Memories cannot be completely turned off dynamically in the PIC32CX-BZ3 (unless in DS and CLDO is OFF). Therefore, option '000' is unavailable.
4. The power consumption of these modes are : ON > NAP > RET > RET+NAP.

Value	Description
1xx	FLEXRAM in the ON mode

Value	Description
011	FLEXRAM in the NAP mode
010	FLEXRAM in the RET mode
001	FLEXRAM in the RET + NAP mode
000	FLEXRAM is powered OFF (Option is unavailable)

18.5.2 WCMSIZ Register

Name: WCMSIZ
Offset: 0x60
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
							SRAM1_SIZE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 9:8 – SRAM1_SIZE[1:0] Flex RAM Retention Size Configuration in the Deep Sleep Mode
Note: This field is only writable when CFGCON0.PMULOCK = 0.

Value	Description
11	Not available
10	32K Flex RAM is available in retention
01	16K Flex RAM is available in retention
00	Flex RAM is completely powered OFF in the Deep Sleep mode

18.6 Register Summary

See *DSCON* module in the *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DSCON	7:0							ZPBOR	DSSR
		15:8	DSEN		XSEMAEN	RTCPWREQ				RTCCWDIS
		23:16								
		31:24								
0x04	DSWAKE	7:0				DSWDT	RTCC	MCLR		
		15:8								INT0
		23:16								
		31:24								
0x08	DSSEMA1	7:0	DSSEMA1[7:0]							
		15:8	DSSEMA1[15:8]							
		23:16	DSSEMA1[23:16]							
		31:24	DSSEMA1[31:24]							

Related Links

8. Product Memory Mapping Overview

18.7 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.

18.7.1 DS Control

Name: DSCON
Offset: 0x00
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
	DSEN		XSEMAEN	RTCPWREQ				RTCCWDIS
Access	R/W/HC		R/W	R/W				R/W
Reset	0		0	0				0

Bit	7	6	5	4	3	2	1	0
							ZPBOR	DSSR
Access							R/W/C/HS	R/C/HS/HC
Reset							0	0

Bit 15 – DSEN Deep Sleep Enable bit

Value	Description
0	Enters the Standby Sleep mode on a WFI command
1	Enters the Deep Sleep mode on a WFI command

Bit 13 – XSEMAEN XSEMA General Purpose Registers Enable bit

Value	Description
0	No general purpose register retention in the Deep Sleep mode
1	Enables the general purpose register retention in the Deep Sleep mode

Bit 12 – RTCPWREQ RTCC Module Disable bit

To enable RTCC function, RTCC module level registers must be programmed in addition to the DSCON.RTCPWREQ bit.

Value	Description
0	Enables the RTCC module
1	Disables the RTCC module

Bit 8 – RTCCWDIS RTCC Wake-up Disable bit

Value	Description
0	Enables the wake-up from RTCC
1	Disables the wake-up from RTCC

Bit 1 – ZPBOR Deep Sleep BOR Event Status bit

Value	Description
0	CFGCON4.DSZPBOREN is disabled or VDD did not drop below the DSBOR threshold during the Deep Sleep mode
1	CFGCON4.DSZPBOREN is enabled and VDD dropped below the DSBOR threshold during the Deep Sleep mode **Note:** Unlike all other events, a DSBOR event does not cause a wake-up from the Deep Sleep mode. This bit is present only as a status bit.

Bit 0 – DSSR I/O pin State Release bit

Value	Description
0	Release I/O pins and allow their respective TRIS and LAT bits to control their states
1	Upon waking from Deep Sleep, the I/O pins maintain their previous states (Not user settable). **Note:** The DSSR register bit is readable and can be cleared (but not set) by the software. This bit is automatically set when entering the Deep Sleep mode. While exiting the Deep Sleep mode, due to any wake-up source other than $\overline{\text{MCLR}}$, when all state-related configuration is complete, the software must clear the DSSR bit. When the DSSR bit is cleared, the I/Os will be controlled by their I/O registers.

18.7.2 DSWAKE

Name: DSWAKE
Offset: 0x04
Reset: 0x00000000

The DSWAKE register is only writable by software when DSCON.DSSR = 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
								INT0
Access								R/W/C/HS
Reset								0

Bit	7	6	5	4	3	2	1	0
				DSWDT	RTCC	MCLR		
Access				R/W/C/HS	R/W/C/HS	R/W/C/HS		
Reset				0	0	0		

Bit 8 – INT0 Deep Sleep Interrupt-on-change #0 bit

Value	Description
0	Interrupt-on-change #0 (INT0) is not asserted during the Deep Sleep
1	Interrupt-on-change #0 (INT0) is asserted during the Deep Sleep

Bit 4 – DSWDT Deep Sleep Watchdog Timer Time-out bit

Value	Description
0	DSWDT does not time out during the Deep Sleep
1	DSWDT timed out during the Deep Sleep

Bit 3 – RTCC Deep Sleep Real Time Clock and Calendar Alarm bit

Value	Description
0	Deep Sleep RTC and calendar does not trigger an alarm during the Deep Sleep
1	Deep Sleep RTC and calendar triggers an alarm during the Deep Sleep

Bit 2 – $\overline{\text{MCLR}}$ Deep Sleep MCLR Event bit

Value	Description
0	($\overline{\text{MCLR}}$) pin is not active or is active but not asserted during the Deep Sleep
1	($\overline{\text{MCLR}}$) pin is active and is asserted during the Deep Sleep

18.7.3 DSSEMA1 Register

Name: DSSEMA1
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DSSEMA1[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
	DSSEMA1[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
	DSSEMA1[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
	DSSEMA1[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 – DSSEMA1[31:0] Deep Sleep Persistent General Purpose Register bits
The contents of the DSSEMA1 register are retained, even in the Deep Sleep mode. The DSSEMA1 is disabled by default in the Deep Sleep mode but can be enabled with the XSEMAEN bit (DSCON[13]). All register bits are reset only in the case of a VDD POR event outside of the Deep Sleep mode.

19. Watchdog Timer (WDT)

19.1 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 the 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 the Standby Sleep or Idle mode.

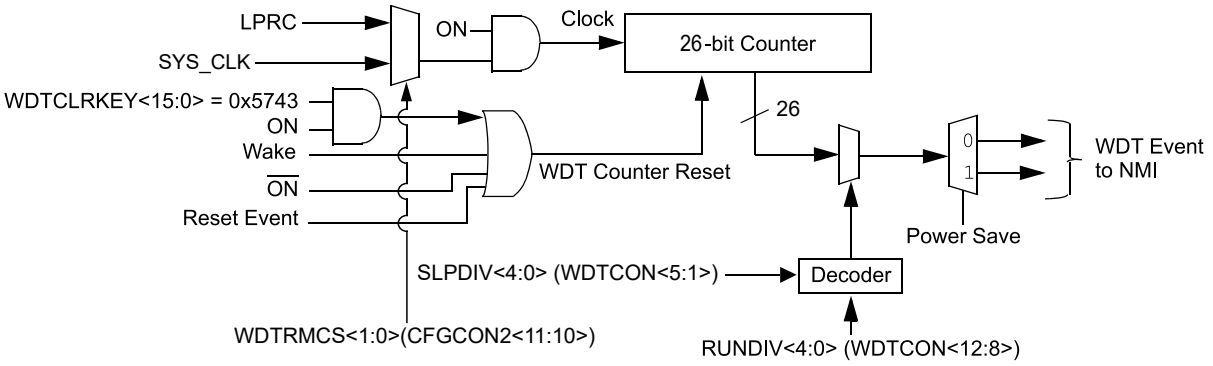
19.2 Features

- Configuration Using Fuses (DEVCFG) or Software Controlled
- Up to 32 Configurable Time-Out Periods
- Can Wake the Device from the Standby Sleep or Idle Mode
- Independent Run and the Standby Sleep Mode Counters
- WDT May Use Alternate Clock Source and Postscaler for Run Mode Counter
- Independent 5-Bit Postscalers for Run and the 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.

19.3 Block Diagram

Figure 19-1. Block Diagram



19.4 Watchdog Timer Operation

The primary function of the WDT is to reset the processor in the event of a software malfunction or to wake up the processor in the event of a time-out while in the Standby Sleep mode.

If enabled, the WDT increments until it overflows or times out. A WDT time-out will force a device Reset, except during the Standby Sleep or Idle mode. To prevent a WDT time-out Reset, the user application must periodically clear the WDT by writing a keyword 0x5743 to the WDTCLRKEY[15:0] bits (WDTCON[31:16]) through a single 16-bit write.

Related Links

19.9.1. WDTCON

19.4.1 Modes of Operation

The WDT has two modes of operation:

- Non-Windowed
- Programmable Windowed

The Programmable Windowed mode can be enabled by setting the Watchdog Window Enable (WDTWINEN) bit (WDTCON[0]). In Programmable 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. This window is active when the timer counter is greater than a predetermined value for each option. Any attempts to clear the WDT when the window is not active causes a device Reset. In Non-Windowed mode, software can clear the WDT anytime before the period match occurs.

19.4.2 Enabling and Disabling the WDT

The WDT is enabled or disabled by the device configuration or controlled through software by writing to the WDTCON register. See *WDTCON* from Related Links for more details.

Related Links

19.9.1. WDTCON

19.4.3 Device Configuration Controlled WDT

If the DEVCFG2.WDTEN Configuration bit is set, the WDT is always enabled. The ON control bit (WDTCON[15]) reflects this by reading a '1'. In this mode, the ON bit cannot be cleared in software. The DEVCFG2.WDTEN Configuration bit will not be cleared by any form of Reset. To disable the WDT, the configuration must be rewritten to the device.

Note: The WDT is enabled by default on an unprogrammed device.

The DEVCFG2.WINDIS Configuration bit can be used to enable or disable the Programmable Windowed mode. The window size for the WDT Programmable Windowed mode can be configured using the DEVCFG2.WINSZ Configuration bits.

19.4.4 Software Controlled WDT

If the DEVCFG2.WDTEN Configuration bit is '0', the WDT module can be enabled or disabled (the default condition) by software. In this mode, the ON bit (WDTCON[15]) reflects the status of the WDT under software control. A '1' indicates the WDT module is enabled and a '0' indicates it is disabled. If the DEVCFG2.WINDIS Configuration bit is '0', the WDT Programmable Windowed mode can be enabled or disabled by software. The Programmable Windowed mode can be configured using the WDTWINEN bit (WDTCON[0]). A '1' indicates that Programmable Windowed mode is enabled and '0' indicates it is disabled. The window sizes can be configured by setting the DEVCFG2.WINSZ Configuration bits only and cannot be set in software.

The WDT is enabled in software by setting the Watchdog Timer control bit, ON (WDTCON[15]). The ON control bit is cleared on any device Reset. The bit is not cleared upon a wake from Standby Sleep or exit from Idle mode. The software WDT option allows the user to enable the WDT for critical code segments and to disable the WDT during noncritical segments for maximum power savings. This bit can also be used to disable the WDT while the device is awake to eliminate the need for WDT servicing, then re-enable it before the device is put into the Idle mode or Standby Sleep mode to wake the device at a later time.

19.4.4.1 Watchdog Timer Programmable Window

The window size is determined by the Configuration bits, DEVCFG2.WINSZ and WDTPS. In the Programmable Windowed mode (WDTCON.WDTWINEN = 1), the WDT must be cleared based on the setting of the Window Size Configuration bits (DEVCFG2.WINSZ[1:0]), see the following figure. These bit settings are:

- 11

= WDT window is 25% of the WDT period

- 10
  = WDT window is 37.5% of the WDT period
- 01
  = WDT window is 50% of the WDT period
- 00
  = WDT window is 75% of the WDT period

If the WDT is cleared before the allowed window, a system Reset is immediately generated. See *Clock and Reset Unit (CRU)* from Related Links for more information on which type of Reset occurs.

The Windowed mode is useful for resetting the device during unexpected quick or slow execution of a critical portion of the code.

Figure 19-2. Programmable Windowed WDT

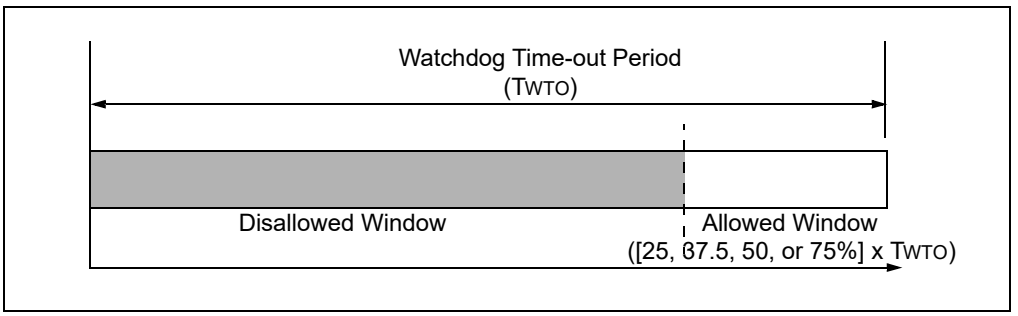
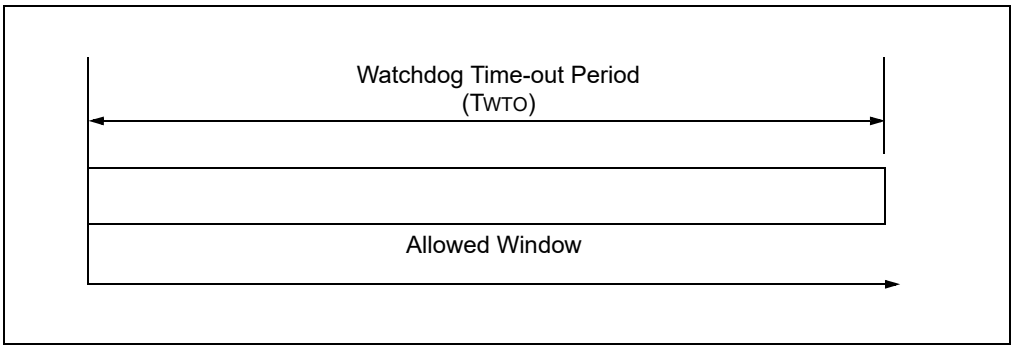


Figure 19-3. Non-Windowed WDT



Related Links

17. Clock and Reset Unit (CRU)

19.4.5 WDT Operation in Power-Saving Modes

The WDT if enabled, will continue operation in the Standby Sleep mode or Idle mode. The WDT module may be used to wake the device from Standby Sleep mode or Idle mode. When the WDT times out in a Standby Sleep mode or Idle mode, a Non-Maskable Interrupt (NMI) is generated and the WDTO bit (RCON[4]) is set. The NMI vectors execution to the CPU start-up address, but does not Reset registers or peripherals. If the device is in Standby Sleep, the SLEEP status bit (RCON[3]) will also be set. If the device is in Idle, the IDLE status bit (RCON[2]) will also be set. These bits allow the start-up code to determine the cause of the wake-up.

19.4.6 WDT NMI Reset Delay

It is possible to program a delay time between a WDT event and a device Reset using the NMI reset counter. See *Resets* from Related Links for more details.

Related Links

17.4. Resets

19.4.7 Resetting the WDT

The following events will Reset both internal WDT counters:

- Disabling the WDT via the ON bit on any device Reset
- Any counter value greater than the selected WDT period

The following event will Reset the Run Mode Counter:

- Detection of a correct write value (0x5743) to the WDTCLRKEY[15:0] bits (WDTCON[31:16])

The following event will Reset the Standby Sleep Mode Counter:

- Exiting from Idle or Standby Sleep due to WDT event

Note: The WDT is not Reset when the device enters a Power-Saving mode. The WDT module must be serviced prior to entering a Power-Saving mode.

19.4.8 WDT Period Selection

The Sleep mode counter always uses the 32 kHz LPRC clock source. The Run mode counter uses the clock source selected by the DEVCFG2.WDTRMCS[1:0] Configuration Bits in fuses to be either the 32 kHz LPRC clock or the SYS_CLK.
Note: The WDT module time-out period is directly related to the frequency of the LPRC Oscillator when clock source is 32 kHz LPRC. The frequency of the LPRC Oscillator varies as a function of the device operating voltage and temperature. See *Electrical Characteristics* from Related Links for LPRC clock frequency specifications.

Related Links

43. Electrical Characteristics

19.4.9 WDT Postscalers

The WDT has a 5-bit postscaler to create a wide variety of time-out periods. This postscaler provides 1:1 through 1:1048576 divider ratios (see the following table). Time-out periods that range between 1 ms and 1048.576 seconds (nominal) can be achieved using the postscaler.

Table 19-1. WDT Time-out Period versus Postscaler Settings(1),(2)

WDTPSR[4:0]/WDTPSS[4:0]	Postscaler Ratio	Time-out Period (Non-windowed Mode)	Time-out Period (Programmable Windowed Mode)(3)
00000	1:1	1 ms	0.75 ms
00001	1:2	2 ms	1.5 ms
00010	1:4	4 ms	3 ms
00011	1:8	8 ms	6 ms
00100	1:16	16 ms	12 ms
00101	1:32	32 ms	24 ms
00110	1:64	64 ms	48 ms
00111	1:128	128 ms	96 ms
01000	1:256	256 ms	192 ms
01001	1:512	512 ms	384 ms
01010	1:1024	1.024s	0.768s

.....continued			
WDTPSR[4:0]/WDTPSS[4:0]	Postscaler Ratio	Time-out Period (Non-windowed Mode)	Time-out Period (Programmable Windowed Mode)(3)
01011	1:2048	2.048s	1.536s
01100	1:4096	4.096s	3.072s
01101	1:8192	8.192s	6.144s
01110	1:16384	16.384s	12.228s
01111	1:32768	32.768s	24.576s
10000	1:65536	65.536s	49.152s
10001	1:131072	131.072s	98.304s
10010	1:262144	262.144s	196.608s
10011	1:524288	524.288s	393.216s
10100	1:1045876	1048.576s	786.432s

Notes:

1. All other combinations result in operation as if the postscaler was set to '10100'.
2. The periods listed are based on a 32 kHz (nominal) input clock.
3. In this case, DEVCFG2.WINSZ = 00. The WDT window is 75% of the selected WDT period.

The settings are chosen using the DEVCFG2.WDTPSR[4:0] inputs for the Run mode counter and the DEVCFG1.WDTPSS[4:0] inputs for the Standby Sleep mode counter. The time-out period of the WDT is calculated as shown in the following equation:

$$WDTPeriod = 1\ ms \bullet 2^{Postscaler}$$

19.5 Interrupt and Reset Generation

The NMI timer provides a delay between WDT events and a device Reset. Set the delay in the System Clock counts from 0 to 255 in the NMICNT[15:0] bits (RNMICON[15:0]). If these bits are set to zero, there is no delay between the 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 (RNMICON.WDTR) is not cleared in RNMICON before the counter reaches zero, a device Reset is issued.

If the corresponding NMI flag in RNMICON is cleared before the counter reaches zero, 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 will be able to return from this NMI interrupt.

The WDTS flag is set if there is a WDT event during the Standby Sleep/Idle mode. The WDTS flag will wake the CPU from Standby Sleep/Idle mode, but will not start the NMI counter, nor cause a Reset.

To detect a WDT Reset, the WDTO bit (RCON[4]), SLEEP bit (RCON[3]) and IDLE bit (RCON[2]) must be tested. If the WDTO bit is '1', the event was due to a WDT time-out. The SLEEP and IDLE bits can, then, be tested to determine if the WDT event occurred while the device was awake or if it was in the Standby Sleep or Idle mode.

19.6 Operation in Debug and Power-Saving Modes

19.6.1 WDT Operation in Power-Saving Modes

The WDT can be used to wake the device from the Standby Sleep or Idle modes. The WDT continues to operate in the Power-Saving modes. A time-out can, then, be used to wake the device. This allows the device to remain in Standby Sleep mode until the WDT expires or another interrupt wakes the device.

If the device does not re-enter the Standby Sleep or Idle mode following a wake-up, the WDT must be disabled or periodically serviced to prevent a device Reset.

19.6.2 WDT Operation in Standby Sleep Mode

The WDT, if enabled, continues operation in the Standby Sleep mode. The WDT may be used to wake the device from the Sleep mode. When the WDT times out in Sleep, an NMI is generated and the WDTO bit (RCON[4]) is set. The NMI vectors execution to the CPU start-up address but does not Reset registers or peripherals. The Standby Sleep status bit (RCON[3]) is set indicating the device was in the Standby Sleep mode. These bits allow the start-up code to determine the cause of the wake-up.

19.6.3 WDT Operation in Idle Mode

The WDT, if enabled, continues operation in the Idle mode. The WDT may be used to wake the device from the Idle mode. When the WDT times out in Idle, an NMI is generated and the WDTO bit (RCON[4]) is set. The NMI vectors execution to the CPU start-up address but does not Reset registers or peripherals. The IDLE status bit (RCON[2]) is set, indicating the device was in the Idle mode. These bits allow the start-up code to determine the cause of the wake-up.

19.6.4 Time Delays During Wake-up

The delay between a WDT time-out and the beginning of code execution depends on the Power-Saving mode.

There is a time delay between the WDT event in the Standby Sleep mode and the beginning of code execution. The duration of this delay consists of the start-up time for the oscillator in use and the PWRT delay, if it is enabled. Unlike a wake-up from the Standby Sleep mode, there are no time delays associated with wake-up from the Idle mode. The system clock is running during the Idle mode; therefore, no start-up delays are required at wake-up.

19.6.5 WDT Operation in Debug Mode

The WDT is always suspended in the Debug mode, and therefore does not time-out.

19.7 Effects of Various Resets

Any form of device Reset clears the WDT. The Reset returns the WDTCON register to the default value and the WDT is disabled unless it is enabled by the device configuration.

Note: After a device Reset, the WDT ON bit (WDTCON[15]) reflects the state of the DEVCFG2.WDTEN.

19.8 Register Summary

See *WDT* module in the *Product Memory Mapping Overview* from Related Links for the base address.

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	WDTCON	7:0			SLPDIV[4:0]					WDTWINEN
		15:8	ON			RUNDIV[4:0]				
		23:16	WDTCLRKEY[7:0]							
		31:24	WDTCLRKEY[15:8]							

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

19.9 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.

19.9.1 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 WDTPSR[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 WDTPSS[4:0] Configuration bits in CFGCON1.

Bit 0 – WDTWINEN Watchdog Timer Window Enable bit

Value	Description
1	Enable windowed WDT
0	Disable windowed WDT

20. Deadman Timer (DMT)

20.1 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 the instructions fetched 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.

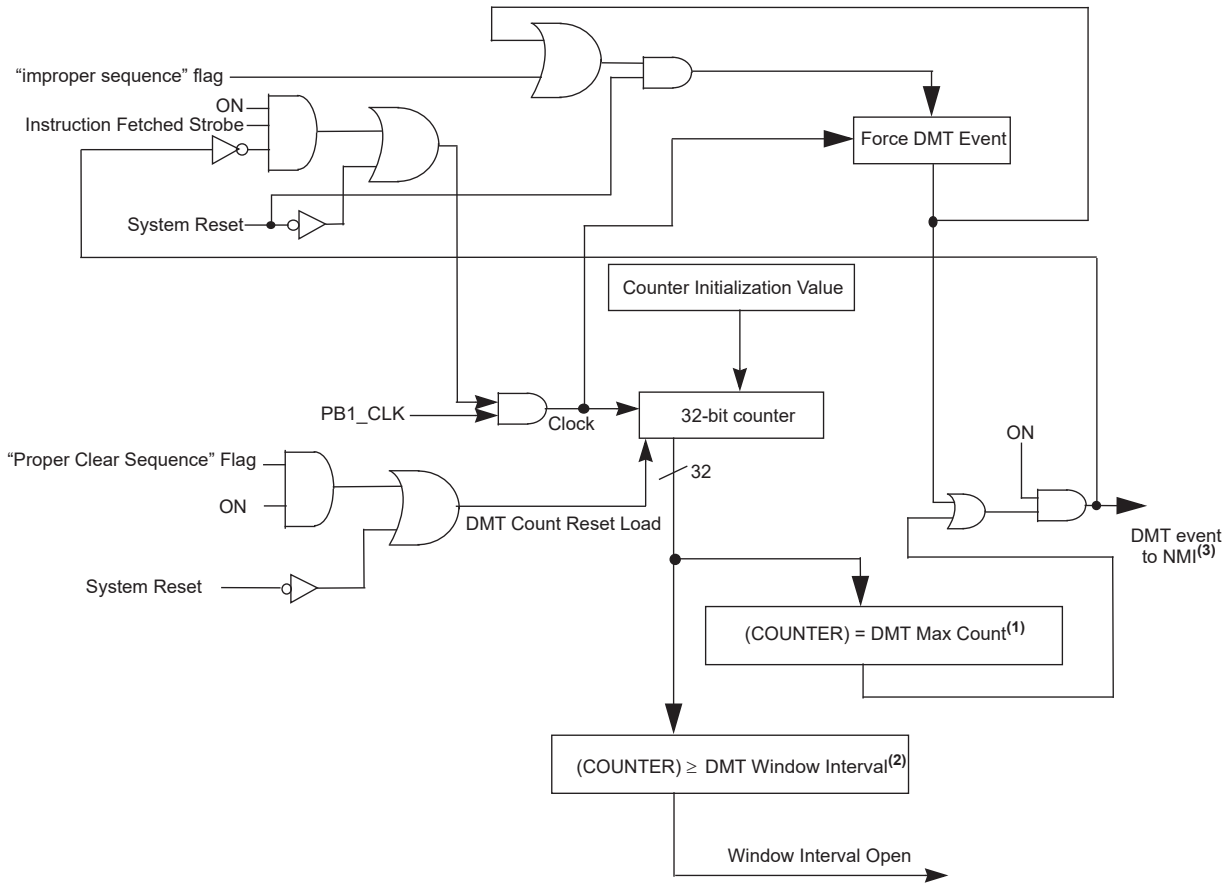
20.2 Features

- 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

20.3 Block Diagram

The following figure illustrates the block diagram of the Deadman Timer.

Figure 20-1. 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.

20.4 DMT Operation

20.4.1 Mode of Operation

The primary function of the 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 that 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 is 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.

20.4.2 Enabling and Disabling the DMT Module

Because of the nature of the DMT, 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. When 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 the DMTCON register will reflect this by reading a ‘1’. In this mode, the ON bit (DMTCON[15]) cannot be cleared in the 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’.

20.4.3 DMT Count Windowed Interval

The DMT module has the Windowed Operation mode. The DMT INTERVAL (DMTINV[2:0]) bits in the CFGCON2 Fuse register sets the window interval value. The PSINTV[31:0] bits in the 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 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, only when the DMT counter value is greater than or equal to the value written to the window interval value, can the DMT clear sequence be executed in the DMT module. If the DMT is cleared before the allowed window, a DMT Reset event is immediately generated.

20.4.4 DMT Count Selection

The DMT count is set by the DMT COUNT (DMTCNT[4:0]) Configuration bits in the CFGCON2 Fuse register. The current DMT count value can be obtained by reading the DMTCNT register.

The PSCNT[31:0] bits in the DMT COUNT POST STATUS (DMTPSCNT) register allow 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 (DMTHOLDREG) register holds the value of the last read DMT upper count values whenever DMTCNT is last read.

20.4.5 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.

20.4.6 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 (DMTPRECLR) register and DMT CLEAR (DMTCLR) register 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.

When 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 DMTEVENT bit in the DMTSTAT register will be set. Refer to the flowchart illustrated in the following figure.

Figure 20-2. Flowchart for Clearing the DMT

20.5 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]							

Related Links

8. Product Memory Mapping Overview

20.6 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

6.4.1.9. CLR, SET and INV Registers

20.6.1 Deadman Timer Control

Name: DMTCON
Offset: 0x00
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
	ON							
Access	HC							
Reset	0							

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

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.

20.6.2 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 when Write Pattern is:

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.

20.6.3 Deadman Timer Clear

Name: DMTCLR
Offset: 0x20
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
	STEP2[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 7:0 – STEP2[7:0] Clear Timer bit when Write Pattern is:

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.

20.6.4 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.

20.6.5 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.

20.6.6 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.

20.6.7 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.

20.6.8 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.

21. RAM Error Correction Code (RAMECC)

21.1 Overview

For safety applications, the PIC32CX-BZ3 family can embed error correction codes (ECC) to detect and correct single bit errors or to enable dual error detection in SRAM. As discussed in the Memories chapter, when the RAMECC is enabled, the top half of the SRAM memory will be reserved to store error correction codes and will not be available for the application. See *SRAM Memory Configuration* from Related Links.

ECC calculation is software-selectable through the CFGCON0.FRECCDIS bit in the Boot Flash Configuration. For additional information, see *System Configuration and Register Locking (CFG)* from Related Links.

Related Links

8.6. SRAM Memory Configuration

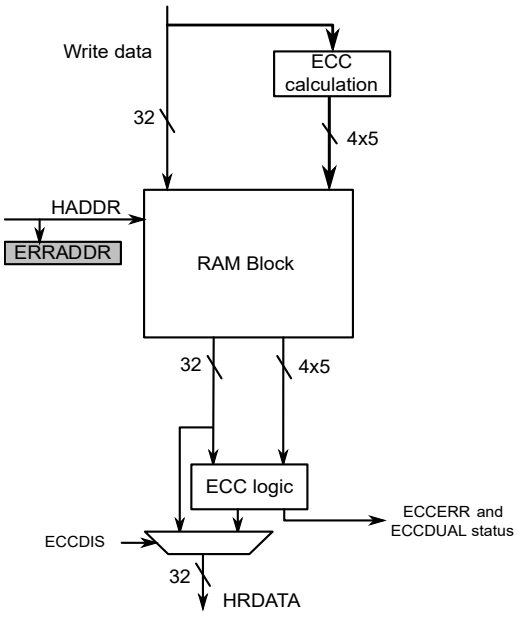
22. System Configuration and Register Locking (CFG)

21.2 Features

- Single Bit Correction and Dual Bit Detection
- Error Interrupt
- Operates Idle and Standby Sleep Mode
- Interrupts Generated by RAMECC Can be Used to Wake-up the Device from the Standby Sleep Mode

21.3 Block Diagram

Figure 21-1. RAMECC Block Diagram



21.4 Signal Description

Not applicable.

21.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

21.5.1 I/O Lines

Not applicable.

21.5.2 Power Management

The RAMECC continues to operate in any Sleep mode (Standby Sleep, Idle) 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

18. Power Management Unit (PMU)

21.5.3 DMA

Not applicable.

21.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the RAMECC interrupt(s) requires the interrupt controller to be configured first.

21.5.5 Events

Not applicable.

21.5.6 Debug Operation

When the CPU is halted in the Debug mode, the RAMECC corrects and logs ECC errors based on the following table.

Table 21-1. 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.

21.5.7 Register Access Protection

All registers with write access are optionally write protected by the 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
24. Peripheral Access Controller (PAC)

21.6 Functional Description

21.6.1 Interrupts

The RAMECC has the following interrupt sources:

- Dual Bit Error (DUALE) – Indicates that a dual bit error was detected
- Single Bit Error (SINGLEE) – Indicates that a single-bit error was 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 OR'ed together at the 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.

21.6.2 Principle of Operation

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.SINGLEE). 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.SINGLEE are cleared. If a dual-bit error occurs while INTFLAG.SINGLEE is set, the ERRADDR register is updated with the dual bit error information and INTFLAG.DUALE is also set.

The INTFLAG.SINGLEE and INTFLAG.DUALE bits are both cleared on the ERRADDR read.

The block diagram illustrates 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.

21.7 Register Summary

See the *RAMECC* 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	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[17:16]	
		31:24								
0x08 ... 0x0E	Reserved									
0x0F	DBGCTRL	7:0							ECCELOG	ECCDIS

Related Links

8. Product Memory Mapping Overview

21.8 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 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

21.5.7. Register Access Protection

21.8.1 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	SINGLEE
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 – SINGLEE 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.

21.8.2 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.

21.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x02
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							DUALE	SINGLEE
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 – SINGLEE 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.

21.8.4 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 start-up based on the DEVCFG0/CFGCON0.FRECCDIS bit in the Boot Flash device configuration. When enabled, the calculated ECC is written to SRAM 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.

21.8.5 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[17:16]	
Access							R	R
Reset							0	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 17:0 – ERRADDR[17:0] 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 is updated with the address of the dual-bit error, otherwise, it stalls on the first error occurrence. This register reads as '0' unless INTFLAG.SINGLEE and/or INTFLAG.DUALE are '1'.

21.8.6 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 disables 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.

22. System Configuration and Register Locking (CFG)

22.1 Overview

A PIC32CX-BZ3 family device includes several non-volatile (programmable) configuration words that define the device's behavior. The device configuration words are located in the Boot Flash device config memory. These configuration words are loaded on equivalent system configuration registers after the device Reset. The write access to the system configuration registers is controlled through locking registers.

22.2 Features

This PIC32CX-BZ3 device provides several user-writable configuration registers related to the configuration and operation of the system.

- Permission Group Configuration Register (CFGPG) Defines the Permission Group
- System Key Register (SYSKEY) Defines the System Key
- Configuration Control Register 0 (CFGCON0(L)) Provides Control, Selection and Locking for Various Features of the Device
  Note: The registers marked with (L) are loadable from Flash.
  - PPS register locking
  - PMD register locking
  - CFGPG register locking
  - Config register locking
  - Trace port enable
  - Flash, SRAM ECC control
  - RTCC, AC alternate pinout selection
  - SERCOM slew rate enable
- Configuration Control Register 1 (CFGCON1(L)) Provides Control, Selection and Locking for Various Features of the Device
  - QSPI DDR mode clock enable
  - High-speed SERCOM, QSPI enable
  - WDT Sleep mode prescale configuration
  - I^2C slew rate control
- Configuration Control Register 2 (CFGCON2(L)) Provides Control, Selection and Locking for Various Features of the Device
  - DMT enable and configuration
  - WDT enable and configuration
  - Clock monitoring and control
  - Oscillator enable and configuration
  - Two-speed start-up enabled in Sleep mode bit
- Configuration Control Register 4 (CFGCON4(L)) Provides Control, Selection and Locking for Various Features of the Device
  - Deep sleep modules control
  - SOSC configuration control
  - RTCC event control

- User Unique ID Register (USERID(L)) Provides the End User with a 16-Bit ID Field that Can be Read Out Directly Through the SWD Interface Via the USERID SWD Instruction

22.3 Modes of Operation

22.3.1 System Configuration Words

The device configuration words programmed in Boot Flash memory (NVR pages) get loaded on equivalent registers after the device Reset. The following table shows the mapping between the Boot Flash memory region and loading registers. Registers marked with (L) are loadable from Flash, and they can be controlled by software after the boot with the correct unlock sequence. After programming the configuration words, the user must Reset the device to ensure the configuration data is reloaded with the new programmed values.

Table 22-1. Device Configuration in Flash vs Register

Device Configuration (Flash)	Physical Address in Flash	Reloaded Register
FBCFG0	0x0080_5F9C	BCFG0 (0x4400_0200)
FBCFG1/DEVCFG0	0x0080_5F98	CFGCON0(L) (0x4400_0000)
FBCFG2/DEVCFG1	0x0080_5F94	CFGCON1(L) (0x4400_0010)
FBCFG3/DEVCFG2	0x0080_5F90	CFGCON2(L) (0x4400_0020)
FBCFG4/DEVCFG4	0x0080_5F8C	CFGCON4(L) (0x4400_0040)
FBCFG5/FUSERID	0x0080_5F88	USERID(L) (0x4400_00A0)

Other than device configurations in the Boot Flash region, there are some more system configuration registers. They are run-time programmable and do not have associated Flash region.

- CFGPGQOS – This register defines the permission group settings for various bus hosts on the device bus matrix.
- CFGPCLKGENx (x = 1, 2, 3, and 4) – These registers dictate the peripheral clock selection and enable the clock for the specific peripheral. See *Clock and Reset Unit (CRU)* from Related Links for more details.

Related Links

17. Clock and Reset Unit (CRU)

22.3.1.1 System Configuration Register Protection

To ensure data integrity of each system configuration word, a comparison is continuously made between each configuration bit and its stored complement. If a mismatch is detected, a Configuration Mismatch Reset is generated causing a device Reset.

22.3.2 Alternate System Configuration Words

In the PIC32CX-BZ3 family of devices, the configuration words select various device configurations and are located at physical addresses from 0x00805F80 (FBCFG7).

If an unrecoverable ECC error occurs when reading the configuration words, the alternate configuration words are used to configure the device from Boot Flash memory. This configuration can be identical to the primary configuration words or different to operate in another condition. The alternate configuration words are located at physical addresses from 0x00805E80 (ALTFBCFG7). To flag that an ECC error has occurred, the BCFGERR (RCON[27]) bit is set.

If uncorrectable ECC errors are found in both primary and alternate words, the BCFGFAIL (RCON[26]) bit is set and the default configuration is used.

After programming the configuration words, the user application must reset the device to ensure the configuration data is reloaded with the new programmed values.

22.4 Locking and Unlocking the System Configuration Registers

Write access to the system configuration registers is controlled via the CFGCON0.CFGLOCK[1:0] register bits.

22.5 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.

22.6 Register Locking

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, system lock-protected registers are not writable. The system lock feature protects registers that are system-critical. The PIC32CX-BZ3 provides different methods of register level locking:

22.6.1 One-Way Lock

This mechanism provides a one-way lock (when locked, only a Reset can unlock) using the CFGCON0.IOLOCK, CFGCON0.PMDLOCK, CFGCON0.PMULOCK and CFGCON0.PGLOCK register bits. This method includes protection for the following registers:

- All PPS registers using CFGCON0.IOLOCK
- All PMD registers using CFGCON0.PMDLOCK
- CFGPGQOS register using CFGCON0.PGLOCK
- All PMU registers using CFGCON0.PMULOCK

22.6.2 One-Way or Two-Way Lock (Software Selectable)

If CFGCON0.CFGLOCK[1:0] is '10', it locks the registers, but it is also possible to unlock the registers by writing '00' to CFGCON0.CFGLOCK[1:0]. If CFGCON0.CFGLOCK[1:0] is '11', it locks the registers, and they are no longer possible to unlock as CFGCON0.CFGLOCK[1:0] bits are also locked. Only a system Reset can unlock them. This mechanism provides one-way or two-way lock (software selectable) using the CFGCON0.CFGLOCK[1:0] register bits. This method applies to the following registers:

- BCFG0
- CFGCON0
- CFGCON1
- CFGCON2
- CFGCON4
- CFGPCLKGENx
- USER_ID

22.6.3 Two-Way Lock and Unlock

Each module that uses the system lock feature describes register bits and functions, which are affected by this system lock feature. A specific sequence of writes to the SYSKEY register unlock the access to those register bits and features.

This locking method provides a two-way (locking and unlocking) write lock of system-critical registers. It includes protection for the following registers:

- CRU.OSCCON
- CRU.OSCTUN

- CRU.SPLLCON
- CRU.RSWRST
- CRU.RNMICON
- CRU.PB1DIV
- CRU.PB2DIV
- CRU.PB3DIV
- CRU.SLEWCON
- CRU.CLK_DIAG

22.6.3.1 Unlock Requirements

The unlock sequence must be atomic. If any other peripheral bus access occurs on the same peripheral bus where SYSKEY resides during the unlock attempt sequence, the unlock fails. Therefore, turn OFF all bus initiators, such as DMA, and disable interrupts.

22.6.3.2 Unlock Sequence

The following steps to be followed to unlock the CFG registers. The unlock sequencer looks for steps 3 and 4 to be atomic. For this sequence, atomic means that there is no other activity on the peripheral bus between steps 3 and 4. Step 2 is only needed to ensure that the sequence starts from a known locked state.

1. Suspend all other peripheral bus accesses.
2. Write SYSKEY = 0x00000000
3. Write SYSKEY = 0xAA996655
4. Write SYSKEY = 0x556699AA

22.6.3.3 Lock Sequence

When the system is unlocked, any write to the SYSKEY register causes the system lock to become active.

22.6.3.4 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.

22.7 Effects of Various Resets

The configuration data is reloaded from the corresponding Boot Flash memory configuration words on the following types of Reset:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- External Reset (NMCLR)
- Configuration Mismatch Reset (CM)
- Watchdog Timer Reset (WDTR)
- Software Reset (SWR)
- NMI Time-out Reset (NMITR)

22.8 Register Summary

See *CFG* module in the *Product Memory Mapping Overview* from Related Links for the base address.

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	CFGCON0(L)	7:0	CPENFILT	ACCMP1_ALTEN		ADCPOVR	JTAGEN	TROEN	SWOEN	
		15:8	CFGLOCK[1:0]		IOLOCK	PMDLOCK	PGLOCK	PMULOCK	RTCOUT_ALTEN	RTCIN0_ALTEN
		23:16	SLRTEN2	SLRTEN1	SLRTEN0	HPLUGDIS	SMBUSEN2	SMBUSEN1	SMBUSEN0	
		31:24		FRECCDIS	ECCCTL[1:0]			INT0P	INT0E	PCM
0x04 ... 0x0F	Reserved									
0x10	CFGCON1(L)	7:0	ZBTWKSYS	ECC_SEL_MEM	TRCEN					
		15:8	QSCHE_EN	SMCLR	SLRCTRL2	SLRCTRL1	SLRCTRL0		CMP1_OE	CMP0_OE
		23:16	I2CDSEL2	I2CDSEL1	I2CDSEL0	CCL_OE		SCOM_HSEN[1:0]		QSPI_HSEN
		31:24		CLKZBREF	QSPIDDRM	WDTPSS[4:0]				
0x14 ... 0x1F	Reserved									
0x20	CFGCON2(L)	7:0			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_MOD	VBKP_DIVSEL	VBKP_32KCSEL[1:0]		VBKP_1KCSEL	RTCEVENT_EN	RTCEVENTSEL[1:0]	
		23:16	DSWDTPS[2:0]			DSZPBOREN	CPEN_DLY[2:0]			RTCEVTYPE
		31:24	RTCNTM_CSEL	LPOSCEN	UVREGROVR	DSBITEN	DSWDTEN	DSWDTLPRC	DSWDTPS[4:3]	
0x44 ... 0x4F	Reserved									
0x50	CFGPGQOS	7:0					CPUQOS[1:0]		CPUPG[1:0]	
		15:8							DMAPG[1:0]	
		23:16	CRYPTOQOS[1:0]		CRYPTOPG[1:0]					
		31:24	WISIBQOS[1:0]		FCQOS[1:0]				DSUPG[1:0]	
0x54 ... 0x5F	Reserved									
0x60	CFGPCLKGEN1	7:0	FREQMRCD	FREQMRCSEL[2:0]			EICCD	EICCSEL[2:0]		
		15:8	SERCOM01CD	SERCOM01CSEL[2:0]			FREQMMCD	FREQMMCSEL[2:0]		
		23:16	TCC12CD	TCC12CSEL[2:0]			SERCOM2CD	SERCOM2CSEL[2:0]		
		31:24	CM4TCD	CM4TCSEL[2:0]						
0x64 ... 0x6F	Reserved									
0x70	CFGPCLKGEN2	7:0	EVSYSC2CD	EVSYSC2SEL[2:0]			EVSYSC1CD	EVSYSC1SEL[2:0]		
		15:8	EVSYSC4CD	EVSYSC4SEL[2:0]			EVSYSC3CD	EVSYSC3SEL[2:0]		
		23:16	EVSYSC6CD	EVSYSC6SEL[2:0]			EVSYSC5CD	EVSYSC5SEL[2:0]		
		31:24	EVSYSC8CD	EVSYSC8SEL[2:0]			EVSYSC7CD	EVSYSC7SEL[2:0]		
0x74 ... 0x7F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80	CFGPCLKGEN3	7:0	EVSYSC10CD	EVSYSC10SEL[2:0]			EVSYSC9CD	EVSYSC9SEL[2:0]		
		15:8	EVSYSC12CD	EVSYSC12SEL[2:0]			EVSYSC11CD	EVSYSC11SEL[2:0]		
		23:16	TCC0CD	TCC0CSEL[2:0]			ACCD	ACCSEL[2:0]		
		31:24								
0x84 ... 0x8F	Reserved									
0x90	CFGPCLKGEN4	7:0	TC1CD	TC1CSEL[2:0]			TC0CD	TC0CSEL[2:0]		
		15:8	TC45CD	TC45CSEL[2:0]			TC23CD	TC23CSEL[2:0]		
		23:16					TC67CD	TC67CSEL[2:0]		
		31:24								
0x94 ... 0x9F	Reserved									
0xA0	USER_ID	7:0	USER_ID[7:0]							
		15:8	USER_ID[15:8]							
		23:16								
		31:24								
0xA4 ... 0xAF	Reserved									
0xB0	SYSKEY	7:0	SYSKEY[7:0]							
		15:8	SYSKEY[15:8]							
		23:16	SYSKEY[23:16]							
		31:24	SYSKEY[31:24]							
0xB4 ... 0x01FF	Reserved									
0x0200	BCFG0	7:0							PCSCMODE	
		15:8								
		23:16								
		31:24	BINFOVALID0		SIGN	CP				

Related Links

6.4.1.9. CLR, SET and INV Registers

8. Product Memory Mapping Overview

22.9 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 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.

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
- HS = Hardware Set
- HC = Hardware Cleared

22.9.1 Configuration Control Register 0

Name: CFGCON0(L)
Offset: 0x00
Reset: 0x7100000b
Property: -

The CFGLOCK[1:0] register bits are writable only when CFGLOCK[0] = 0.

The IOLOCK, PMDLOCK and PGLOCK register bits can only be cleared on a system Reset. Thereafter, these bits are writable using CFGLOCK.

This register is loaded with trusted data from FBCFG1/DEVCFG0 during the pre-boot period. Trusted data from Flash means that when there is no BCFG* fail status and BINFOVALID = 0 during Flash, the configuration word reads. If accompanied by the fail status BCFGFAIL (RCON[26]) or blank/erase indication, Reset values (described in the following register description) are retained and new values from FBCFG1 are not loaded.

Under all conditions, Flash loading is omitted for the following bits in the CFGCON0 register:

- IOLOCK
- CFGLOCK[1:0]
- PMDLOC
- PGLOCK
- PMULOCK
- JTAGEN
- HPLUGDIS

Hence, writing these bits in Boot Flash does not have an effect on the configuration register.

Bit	31	30	29	28	27	26	25	24
		FRECCDIS	ECCCTL[1:0]			INT0P	INT0E	PCM
Access		R/L	R/W/L	R/W/L		R/W/L	R/W/L	R/W/L
Reset		1	1	1		0	0	1

Bit	23	22	21	20	19	18	17	16
	SLRTEN2	SLRTEN1	SLRTEN0	HPLUGDIS	SMBUSEN2	SMBUSEN1	SMBUSEN0	
Access	R/W/L	R/W/L	R/W/L	R/W	R/W/L	R/W/L	R/W/L	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
	CFGLOCK[1:0]		IOLOCK	PMDLOCK	PGLOCK	PMULOCK	RTCOUT_ALTEN	RTCIN0_ALTEN
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	ACCMP1_ALTEN		ADCPOVR	JTAGEN	TROEN	SWOEN	
Access	R/W/L	R/W/L		R/W/L	R/W/L	R/W/L	R/W/L	
Reset	0	0		0	1	0	1	

Bit 30 – FRECCDIS Flex RAM (SRAM) ECC Control
Note: Only a read-only fuse bit sets the initialization value of RAMECC Control. The true RAMECC override is available in the RAMECC module.

Value	Description
1	ECC is disabled
0	ECC is enabled

Bits 29:28 – ECCCTL[1:0] Flash ECC Control

Note: These bits are 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 (NVMCON.NVMOP[3:0] != 1 (Word programming))

Bit 26 – INT0P INT0P Polarity

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	INT0 Polarity (High)
0	INT0 Polarity (Low)

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] of ARM protection control bus.

Bit 23 – SLRTEN2 Slew Rate Enable for SERCOM2

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 22 – SLRTEN1 Slew Rate 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 Slew Rate 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

Value	Description
0	Hot plugging is enabled

Bit 19 – SMBUSEN2 SMBus Enable for SERCOM2
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

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

Bits 15:14 – CFGLOCK[1:0] Configuration Register Lock
Note: These bits are 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 bits 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

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

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

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- For the 32-pin variant PIC32CX5109BZ31032 device, this bit is set to be '0'.

Value	Description
1	AC_CMP1 Out is available on PA6
0	AC_CMP1 Out is available on PA13

Bit 4 – ADCPOVR ADC Charge Pump Override

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Overridden (Software controlled)
0	Hardware controlled

Bit 3 – JTAGEN JTAG Enable

Note: JTAG functionality is not available in the PIC32CX-BZ3 devices. The default value of this bit is '1'. It is recommended to write '0' to this bit during Application initialization to use JTAG pins for regular GPIO functionality. For pin details, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

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	Start Trace Clock and enable Trace Outputs (Trace probe must be present)
0	Stop Trace Clock and disable Trace Outputs

Bit 1 – SWOEN SWO Enable on two-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

Related Links

5. Pinout and Signal Descriptions List
6. I/O Ports and Peripheral Pin Select (PPS)

22.9.2 Configuration Control Register 1

Name: CFGCON1(L)
Offset: 0x10
Reset: 0x1f00443b
Property: -

This register is loaded with trusted data from FBCFG2/DEVCFG1 during the pre-boot period.

Trusted data from Flash means that when there is no BCFG* fail status during Flash, configuration word reads. If accompanied by fail status BCFGFAIL (RCON[26]) or blank/erase indication, Reset values (described in the following register description) are retained, and new values from FBCFG2 are not loaded.

Under all conditions, Flash loading is omitted for the ZBTWKSYS bit in CFGCON1 register. Hence, writing this bit in Boot Flash will not have an effect on the configuration register.

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		SCOM_HSEN[1:0]		QSPI_HSEN
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	15	14	13	12	11	10	9	8
	QSCHE_EN	SMCLR	SLRCTRL2	SLRCTRL1	SLRCTRL0		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
Reset	0	1	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
	ZBTWKSYS	ECC_SEL_MEM	TRCEN					
Access	R/W/L	R/W/L	R/W/L					
Reset	0	0	1					

Bit 30 – CLKZBREF External Reference Clock
The external reference clock output from the Zigbee wireless subsystem on the REFO1 pin, which is configurable through PPS
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Clock from Zigbee wireless subsystem on PPS.REFO1 is enabled
0	No clock from Zigbee wireless subsystem on PPS.REFO1

Bit 29 – QSPIDDRM QSPI Double Data Rate (DDR) Mode Clock Enable
Notes:

- When using the QSPI DDR mode, the System Clock (SYS_CLK) must be <= 32 MHz.
- This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	QSPI DDR mode clock is enabled
0	QSPI DDR mode clock is disabled

Bits 28:24 – WDTPSS[4:0] Watchdog Timer Post-scale Select Sleep bits
Note: These bits are 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 I^2C Delay Select for SERCOM2
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I^2C delay is enabled.
0	I^2C delay is disabled.

Bit 22 – I2CDSEL1 I^2C Delay Select for SERCOM1
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I^2C delay is enabled
0	I^2C delay is disabled

Bit 21 – I2CDSEL0 I^2C Delay Select for SERCOM0
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	I^2C delay is enabled
0	I^2C 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

Bits 18:17 – SCOM_HSEN[1:0] SERCOM (Direct) Enable, 17 = SERCOM0 and 18 = SERCOM1

Notes:

- These bits are only writable when CFGLOCK[1:0] is '00'.
- For the 32-pin variant PIC32CX5109BZ31032 device, QSPI works only with the PPS configuration. It does not work in Direct mode.

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'.
- For the 32-pin variant PIC32CX5109BZ31032 device, QSPI works only with the PPS configuration. It does not work in Direct mode.

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 NMCLR Control

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Does not reset all device NMCLR Reset states
0	NMCLR external Reset causes a faux POR

Bit 13 – SLRCTRL2 I^2C Slew Rate Control for SERCOM2

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 12 – SLRCTRL1 I^2C Slew Rate Control 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 I^2C Slew Rate Control 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 9 – CMP1_OE Analog Comparator-1 Output Enable

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- Not available in the 32-pin variant PIC32CX5109BZ31032 device

Value	Description
1	AC_CMP1 output is enabled
0	AC_CMP1 output is disabled

Bit 8 – CMP0_OE Analog Comparator-0 Output Enable

Notes:

- This bit is only writable when CFGLOCK[1:0] is '00'.
- Not available in the 32-pin variant PIC32CX5109BZ31032 device

Value	Description
1	AC_CMP0 output is enabled
0	AC_CMP0 output is disabled

Bit 7 – ZBTWKSYS ZBT Subsystem External Wake-up source

Notes:

- Write-only bit, with read-as zero; when '1' is written, creates one pulse on the ZBT subsystem.external_NMI0 pin. This enables external system wake-up to ZBT subsystem. This allows CPU and ZBT subsystem wake-up/sleep to be independent of each other.
- Flash fuse loading is excluded for this bit.

Bit 6 – ECC_SEL_MEM ECC Row Selection

This bit comes into effect only for 96K memory variant and if CFGCON0.FRECCDIS = 0. For other cases, this bit setting has no effect.

Note: This bit is only writable when CFGLOCK[1:0] = 00.

Value	Description
1	RowC ECC is applied for Row B
0	RowC ECC is applied for Row A

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

22.9.3 Configuration Control Register 2

Name: CFGCON2(L)
Offset: 0x20
Reset: 0x00
Property: -

This register is loaded with trusted data from FBCFG3/DEVCFG2 during the pre-boot period.

Trusted data from Flash means that when there is no BCFG* fail status during Flash, configuration word reads. If accompanied by fail status BCFGFAIL (RCON[26]) or blank/erase indication, Reset values (described in the following register description) are retained and new values from FBCFG3 are not loaded.

Under all conditions, Flash loading is omitted for POSCMD[1:0] bits in CFGCON2 register. Hence, writing these bits in Boot Flash must not have an effect on the configuration register.

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
			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
Reset			1	1	1	0	0	0

Bit 31 – DMTEN Dead Man Timer Enable
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	DMT is enabled always and DMTCON.ON bit does not have control
0	DMT disabled (control is placed on the DMTCON.ON bit)

Bits 30:26 – DMTCNT[4:0] Dead Man Timer Count Select
Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
00000	Counter value is 2^8 for DMTPSCNT[31:0]
00001	Counter value is 2^9 for DMTPSCNT[31:0]
. . .	...
10100	Counter value is 2^28 for DMTPSCNT[31:0]
10101	Counter value is 2^29 for DMTPSCNT[31:0]
10110	Counter value is 2^30 for DMTPSCNT[31:0]
10111	Counter value is 2^31 for DMTPSCNT[31:0]
11000 – 11111	Reserved

Bits 25:24 – WINSZ[1:0] Watchdog Timer Window Size
Note: These bits are 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 – WDTEN Watchdog Timer Enable
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	WDT is enabled always and WDTCON.ON bit does not have control
0	WDT is disabled (control is placed on the WDTCON.ON bit)

Bit 22 – WINDIS Windowed Watchdog Timer Disable
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Standard WDT selected; windowed WDT disabled
0	Windowed WDT enabled

Bit 21 – WDTSPGM Watchdog Timer Stop during Flash Programming
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	The WDT stops during NVR programming
0	The WDT runs during NVR programming

Bits 20:16 – WDTPSR[4:0] Watchdog Timer Post-scale Select Run bits
Note: These bits are 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 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 enabled
0	Software clock switching disabled

Bit 13 – WAKE2SPD Two-Speed Start-up Enabled in the Sleep mode
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	When the device EXITS Sleep mode, the SYS_CLK is from FRC until the selected clock is ready.
0	Reserved

Bit 12 – SOSCSEL SOSC Selection Configuration
Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Crystal (SOSCI/SOSCO) mode
0	Digital (SCLKI) mode

Bits 11:10 – WDTRMCS[1:0] WDT RUN Mode Clock Select
Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	LPRC
10	Reserved
01	Reserved
00	WDT PB Clock (PB1_CLK)

Bits 9:8 – POSCMD[1:0] Primary Oscillator Configuration
Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	Primary Oscillator mode is disabled
10	Reserved
01	Reserved
00	Primary Oscillator mode is selected

Bits 5:3 – DMTINTV[2:0] Dead Man Timer Count Window Interval
Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
000	Window/Interval value is zero for DMTPSINTV[31:0] - Windowed mode is disabled
001	Window/Interval value is 1/2 Counter value for DMTPSINTV[31:0]
010	Window/Interval value is 3/4 Counter value for DMTPSINTV[31:0]
011	Window/Interval value is 7/8 Counter value for DMTPSINTV[31:0]
100	Window/Interval value is 15/16 Counter value for DMTPSINTV[31:0]
101	Window/Interval value is 31/32 Counter value for DMTPSINTV[31:0]
110	Window/Interval value is 63/64 Counter value for DMTPSINTV[31:0]
111	Window/Interval value is 127/128 Counter value for DMTPSINTV[31:0]

Bits 2:0 – ACMP_CYCLE[2:0] AC Comparator Result Wait Cycles
Note: These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
n	Wait for 32 µs* ACMP_CYCLE[2:0]+1 cycles to generate comparator done indication

22.9.4 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 that when there is no BCFG* fail status during Flash, configuration word reads. If accompanied by fail status BCFGFAIL (RCON[26]) or blank/erase indication, 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_MOD	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	Enable Low Power (Secondary) Oscillator, also at Reset
0	Disable Low Power (Secondary) Oscillator

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	Enable DS bit in DSCON
0	Disable DS bit in DSCON

Bit 27 – DSWDTEN Deep Sleep Watchdog Timer Enable

Note: This bit is only writable when CFGLOCK[1:0] is '00'.

Value	Description
1	Enable DSWDT during deep sleep
0	Disable DSWDT during deep sleep

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	Enable ZPBOR during deep sleep
0	Disable ZPBOR during deep sleep

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_32KCSEL[1:0] VDDBUKPCORE 32 KHz Clock Source Selection
Notes:

- When '00' or '01', the Deep Sleep mode is entered and it falls back to '11' before entering Deep Sleep. Any change of clock source results in gaps in the LPCLK output.
- These bits are only writable when CFGLOCK[1:0] is '00'.

Value	Description
11	LPRC
10	SOSC
01	POSC
00	FRC

Bit 11 – VBKP_1KCSEL 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	Enables RTCC-Event output

Value	Description
0	Disables RTCC-Event output

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
Gain configuration for SOSC Oscillator:
G3>G2>G1>G0

- 11 = Gain is G3
- 10 = Gain is G2
- 01 = Gain is G1
- 00 = Gain is G0

Note: These bits are only writable when CFGLOCK[1:0] is '00'.

22.9.5 Permission Group Configuration

Name: CFGPGQOS
Offset: 0x50
Reset: 0xe044004c
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 to provide software based protection mechanism to device memory mapped region.

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
	CRYPTOQOS[1:0]		CRYPTOPG[1:0]					
Access	R/W/L	R/W/L	R/W/L	R/W/L				
Reset	0	1	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
					CPUQOS[1:0]		CPUPG[1:0]	
Access					R/W/L	R/W/L	R/W/L	R/W/L
Reset					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
The DSU bus host has access to Access Controlled memory regions in the Bus Structure's Permission Groups SFRs.

- DSUPG[1:0] == 2'b11 : Initiator is assigned to Permission Group 3
- DSUPG[1:0] == 2'b10 : Initiator is assigned to Permission Group 2

- DSUPG[1:0] == 2'b01 : Initiator is assigned to Permission Group 1
- DSUPG[1:0] == 2'b00 : Initiator is assigned to Permission Group 0

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

Bits 23:22 – CRYPTOQOS[1:0] Crypto 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 – CRYPTOPG[1:0] Crypto Permission Group

The Crypto bus host has access to Access Controlled memory regions in the Bus Structure's Permission Groups SFRs.

- CRYPTOPG[1:0] == 2'b11 : Initiator is assigned to Permission Group 3
- CRYPTOPG[1:0] == 2'b10 : Initiator is assigned to Permission Group 2
- CRYPTOPG[1:0] == 2'b01 : Initiator is assigned to Permission Group 1
- CRYPTOPG[1:0] == 2'b00 : Initiator is assigned to Permission Group 0

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

Bits 9:8 – DMAPG[1:0] DMA (Rd/Wr) Permission Group

The DMA bus host has access to Access Controlled memory regions in the Bus Structure's Permission Groups SFRs.

- DMAPG[1:0] == 2'b11 : Initiator is assigned to Permission Group 3
- DMAPG[1:0] == 2'b10 : Initiator is assigned to Permission Group 2
- DMAPG[1:0] == 2'b01 : Initiator is assigned to Permission Group 1
- DMAPG[1:0] == 2'b00 : Initiator is assigned to Permission Group 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

The CPU Bus host has access to Access Controlled memory regions in the Bus Structure's Permission Groups SFRs.

- CPUPG[1:0] == 2'b11 : Initiator is assigned to Permission Group 3
- CPUPG[1:0] == 2'b10 : Initiator is assigned to Permission Group 2
- CPUPG[1:0] == 2'b01 : Initiator is assigned to Permission Group 1
- CPUPG[1:0] == 2'b00 : Initiator is assigned to Permission Group 0

Notes:

- CPUPG[1:0] automatically reverts to `2'b00` when the CPU acknowledges entering into an NMI exception
- This field is only writable when CFGCON0.PGLOCK = 0

.

22.9.6 Peripheral Clock Generator 1

Name: CFGPCLKGEN1
Offset: 0x60
Reset: 0x00000000
Property: -

The CFGPCLKGEN1 dictates the peripheral clock selection described in the Clock and Reset Unit chapter. See *Clock and Reset Unit (CRU)* from Related Links.

There is no Flash location for this register because the purpose of this register is to provide application-based peripheral clocking selection. This is best handled in the application software drivers.

Bit	31	30	29	28	27	26	25	24
	CM4TCD	CM4TCSEL[2:0]						
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
	TCC12CD	TCC12CSEL[2:0]			SERCOM2CD	SERCOM2CSEL[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
	SERCOM01CD	SERCOM01CSEL[2:0]			FREQMMCD	FREQMMCSEL[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
	FREQMRCD	FREQMRCSEL[2:0]			EICCD	EICCSEL[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 – CM4TCD CM4_Trace Peripheral Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 CFGCON0.PGLOCK is '0'.

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 Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – SERCOM2CD SERCOM2 Peripheral Clock Enable
Notes:

- This field is only writable when CFGCON0.PGLOCK is '0'.
- For the 32-pin variant PIC32CX5109BZ31032 device, the SERCOM2 is not available

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – SERCOM2CSEL[2:0] SERCOM2 Peripheral Clock Selection
Notes:

- This field is only writable when CFGCON0.PGLOCK is '0'.
- For the 32-pin variant PIC32CX5109BZ31032 device, the SERCOM2 is not available

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – SERCOM01CD SERCOM0 and SERCOM1 Peripheral Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – FREQMMCD FREQM Measurement Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – FREQMMCSEL[2:0] FREQM Measurement Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – FREQMRCD FREQM Reference Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – FREQMRCSEL[2:0] FREQM Reference Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Related Links
17. Clock and Reset Unit (CRU)

22.9.7 Peripheral Clock Generator 2

Name: CFGPCLKGEN2
Offset: 0x70
Reset: 0x00000000
Property: -

The CFGPCLKGEN2 dictates the peripheral clock selection described in the Clock and Reset Unit chapter. See *Clock and Reset Unit (CRU)* from Related Links.

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
	EVSYSC8CD	EVSYSC8SEL[2:0]			EVSYSC7CD	EVSYSC7SEL[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
	EVSYSC6CD	EVSYSC6SEL[2:0]			EVSYSC5CD	EVSYSC5SEL[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
	EVSYSC4CD	EVSYSC4SEL[2:0]			EVSYSC3CD	EVSYSC3SEL[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
	EVSYSC2CD	EVSYSC2SEL[2:0]			EVSYSC1CD	EVSYSC1SEL[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 – EVSYSC8CD EVSYS Channel 8 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 30:28 – EVSYSC8SEL[2:0] EVSYS Channel 8 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 27 – EVSYSC7CD EVSYS Channel 7 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 26:24 – EVSYSC7SEL[2:0] EVSYS Channel 7 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 23 – EVSYSC6CD EVSYS Channel 6 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 22:20 – EVSYSC6SEL[2:0] EVSYS Channel 6 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – EVSYSC5CD EVSYS Channel 5 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – EVSYSC5SEL[2:0] EVSYS Channel 5 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – EVSYSC4CD EVSYS Channel 4 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – EVSYSC4SEL[2:0] EVSYS Channel 4 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – EVSYSC3CD EVSYS Channel 3 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – EVSYSC3SEL[2:0] EVSYS Channel 3 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – EVSYSC2CD EVSYS Channel 2 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – EVSYSC2SEL[2:0] EVSYS Channel 2 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – EVSYSC1CD EVSYS Channel 1 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – EVSYSC1SEL[2:0] EVSYS Channel 1 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Related Links
17. Clock and Reset Unit (CRU)

22.9.8 Peripheral Clock Generator 3

Name: CFGPCLKGEN3
Offset: 0x80
Reset: 0x00000000
Property: -

The CFGPCLKGEN3 dictates the peripheral clock selection described in the Clock and Reset Unit chapter. See *Clock and Reset Unit (CRU)* from Related Links.

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
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TCC0CD	TCC0CSEL[2:0]			ACCD	ACCSEL[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
	EVSYSC12CD	EVSYSC12SEL[2:0]			EVSYSC11CD	EVSYSC11SEL[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
	EVSYSC10CD	EVSYSC10SEL[2:0]			EVSYSC9CD	EVSYSC9SEL[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 – TCC0CD TCC0 Peripheral Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

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 CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 19 – ACCD Analog Comparator Peripheral Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – ACCSEL[2:0] Analog Comparator Peripheral Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – EVSYSC12CD EVSYS Channel 12 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – EVSYSC12SEL[2:0] EVSYS Channel 12 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – EVSYSC11CD EVSYS Channel 11 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – EVSYSC11SEL[2:0] EVSYS Channel 11 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – EVSYSC10CD EVSYS Channel 10 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – EVSYSC10SEL[2:0] EVSYS Channel 10 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – EVSYSC9CD EVSYS Channel 9 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – EVSYSC9SEL[2:0] EVSYS Channel 9 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Related Links
17. Clock and Reset Unit (CRU)

22.9.9 Peripheral Clock Generator 4

Name: CFGPCLKGEN4
Offset: 0x90
Reset: 0x00000000
Property: -

The CFGPCLKGEN4 dictates the peripheral clock selection described in the Clock and Reset Unit chapter. See *Clock and Reset Unit (CRU)* from Related Links.

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
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					TC67CD	TC67CSEL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TC45CD	TC45CSEL[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	7	6	5	4	3	2	1	0
	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 19 – TC67CD TC6 and TC7 Peripheral Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 18:16 – TC67CSEL[2:0] TC6 and TC7 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 15 – TC45CD TC4 and TC5 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 14:12 – TC45CSEL[2:0] TC4 and TC5 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 11 – TC23CD TC2 and TC3 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 10:8 – TC23CSEL[2:0] TC2 and TC3 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 7 – TC1CD TC1 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 6:4 – TC1CSEL[2:0] TC1 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 clock is selected
7	Low power clock is selected

Bit 3 – TC0CD TC0 Clock Enable
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	Clock is disabled
1	Clock is enabled

Bits 2:0 – TC0CSEL[2:0] TC0 Clock Selection
Note: This field is only writable when CFGCON0.PGLOCK is '0'.

Value	Description
0	No clock is selected
1-6	REFO1-6 is clock selected
7	Low power clock is selected

Related Links
17. Clock and Reset Unit (CRU)

22.9.10 User Unique ID

Name: USER_ID
Offset: 0xA0
Reset: 0x00000000
Property: -

The User ID is a 16-bit ID that can 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 SWD interface.
There is no dedicated status bit to indicate when the User ID value is loaded into the USER_ID register and is ready to be read from SWD. It is assumed that a non-zero value for the User ID is 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 that when there is no BCFG* fail status during Flash, configuration word reads. If accompanied by fail status BCFGFAIL (RCON[26]) or blank/erase indication, Reset values (described in the following register description) are retained and new values from FBCFG5/ FUSERID 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
Note: This field is only writable when CFGLOCK[1:0] is '00'.

22.9.11 Boot Configuration 0

Name: BCFG0
Offset: 0x200
Reset: 0x00
Property: -

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 know if the information from Flash is valid or invalid. The BCFG area is critical to device boot-up.
2. Trusted FBCFG* data = (BINFOVALID = 0) and (BCFGFAIL = 0).
3. It is recommended for the application to program this bit to '0' for proper operation.

Value	Description
1	FBCFG0 to FBCFG7 is not valid (Untrusted, flash values are ignored and safe values are used)
0	FBCFG0 to FBCFG7 is valid (Trusted and loaded from Flash)

Bit 29 – SIGN Flash SIGN Bit

This bit is a read only bit. Reading this bit returns the value of the SIGN bit in the FSIGN0 fuse location (invisible to user) in the NVR memory.

Value	Description
1	Unsigned
0	Signed

Bit 28 – CP Code Protect

The CP bit is a read-only bit. Reading this bit returns the inverted value of the CP bit in the FCPN0 Flash location (~FCPN0.CP && ~FSIGN0.SIGN). To set Code Protect, the CP bit in the FCPN0 fuse location in the NVR memory must be set to '1'.

Value	Description
1	Protection enabled
0	Protection disabled

Bit 1 – PCSCMODE PCHE Single Cache Mode

Note: This bit must be changed only when there are no active accesses to Flash from CPU. If this bit is changed while the CPU is accessing Flash, a system hang may result.

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.

22.9.12 System Key Register

Name: SYSKEY
Offset: 0xB0
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	SYSKEY[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
	SYSKEY[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
	SYSKEY[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
	SYSKEY[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 – SYSKEY[31:0] 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.

23. Peripheral Module Disable (PMD)

23.1 Overview

The Peripheral Module Disable (PMD) registers provide a method to disable a peripheral module by stopping all clock sources supplied to that module. When a peripheral is disabled using an appropriate PMD control bit, the peripheral is in a minimum power consumption state. The control and status registers associated with the peripheral are also disabled; therefore, writing to those registers does not have effect and read values are invalid.

23.2 Enabling Peripherals

The PMD register bits control the operation of individual peripherals on the device. When a peripheral's associated PMD bit is '0', the peripheral is enabled and operates as programmed. However, when the associated PMD bit is '1', the peripheral logic, memory map and SFR bits are 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, clear the corresponding PMD register bit to enable the peripheral.

There are some caveats to use PMD bits. The following must be observed:

1. Disabling a peripheral while its ON bit is '0' results in an undefined behavior of the external interface.
2. For bus initiators, the software must verify that the module is not busy after setting the ON bit to '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 before setting the associated PMD bit.

23.3 Controlling Configuration Changes

Because peripherals can be disabled during run time, some restrictions on disabling peripherals are needed to prevent accidental configuration changes. The PIC32CX-BZ3 devices include a Control register lock sequence feature to prevent alterations to the enabled or disabled peripherals.

23.3.1 Control Register Lock

Under normal operation, writing to the PMDx registers is not allowed. Attempted writes appear to execute normally, but the contents of the registers remain unchanged. To change these registers, they must be unlocked in the hardware. The CFGCON0.PMDLOCK Configuration bit controls the register lock. Setting CFGCON0.PMDLOCK prevents writes to the control registers, and clearing CFGCON0.PMDLOCK allows write operation.

23.4 PMD Register Summary

See the *PMD* module in the *Product Memory Mapping Overview* from Related Links for the base address.

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						
		15:8							CVDMD	ADCSARMD
		23:16								RTCCMD
		31:24			QSPIMD					
0xC4 ... 0xCF	Reserved									
0xD0	PMD2	7:0								
		15:8								
		23:16								
		31:24	REFO4MD	REFO3MD	REFO2MD	REFO1MD			REFO6MD	REFO5MD
0xD4 ... 0xDF	Reserved									
0xE0	PMD3	7:0	TC3MD	TC2MD	TC1MD	TC0MD	DACMD	SER2MD	SER1MD	SER0MD
		15:8		TCC2MD	TCC1MD	TCC0MD	TC7MD	TC6MD	TC5MD	TC4MD
		23:16								
		31:24								

Related Links

6.4.1.9. CLR, SET and INV Registers
8. Product Memory Mapping Overview

23.5 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 23-1. Module Enable/Disable Controls

Module	PMD Control	Module Control	Enable/Disable Strategy
AC	Present	Present	Disable at PMD or Module
ADC	Present	Present	Disable at PMD or Module
CCL	NA	Present	Disable at Module
CVD	Present	Present	Disable at PMD or Module
CMCC	NA	Present	Disable at Module
DAC	Present	Present	Disable at PMD or 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

.....continued			
Module	**PMD Control**	**Module Control**	**Enable/Disable Strategy**
PAC	NA	NA	Always Enabled (Dynamic ON/OFF)
QSPI	Present	Present	Disable at PMD or Module
RAMECC	NA	NA	Disabled by default
RTCC	Present	Present	Disable at PMD or Module
SERCOM	Present	Present	Disable at PMD or Module
TC	Present	Present	Disable at PMD or Module
TCC	Present	Present	Disable at PMD or Module

Note: For modules with both PMD control and Module control, Enable = PMDx=0 AND Module Enable=1, Disable =PMDx=1 OR Module Enable=0.

Related Links

24. Peripheral Access Controller (PAC)

23.5.1 PMD1 - Peripheral Module Disable 1 Register

Name: PMD1
Offset: 0x00C0
Reset: 0x00000000
Property: -

Note: This register's bits are only writable when CFGCON0.PMDLOCK = 0.

Bit	31	30	29	28	27	26	25	24
			QSPIMD					
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
							CVDMD	ADCSARMD
Access							R/W/L	R/W/L
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	ADCMD	ACMD						
Access	R/W/L	R/W/L						
Reset	0	0						

Bit 29 – QSPIMD QSPI Module Disable

Value	Description
1	Disables the QSPI module
0	Enables the QSPI module

Bit 16 – RTCCMD RTCC Module Disable (Unused at top level, part of XDS controller SFR)

Value	Description
1	Disables the RTCC module
0	Enables the RTCC module

Bit 9 – CVDMD Shared CVD Module Disable Bit

Value	Description
1	Disables the corresponding shared CVD module
0	Enables the corresponding shared CVD module

Bit 8 – ADCSARMD Shared ADC SAR Core Module Disable Bit

Value	Description
1	Disables the shared ADC SAR Core module
0	Enables the shared ADC SAR Core module

Bit 7 – ADCMD ADC Controller Module Disable

Value	Description
1	Disables the ADC Controller module
0	Enables the ADC Controller module

Bit 6 – ACMD AC Module Disable

Value	Description
1	Disables the AC module
0	Enables the AC module

Note: PMD1.ACMD is not available in the 32-pin variant PIC32CX5109BZ31032 device.

23.5.2 PMD2 - Peripheral Module Disable 2 Register

Name: PMD2
Offset: 0x00D0
Reset: 0x00000000
Property: -

Note: This register bits are only writable when CFGCON0.PMDLOCK = 0.

Bit	31	30	29	28	27	26	25	24
	REFO4MD	REFO3MD	REFO2MD	REFO1MD			REFO6MD	REFO5MD
Access	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

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
Access								
Reset								

Bit 31 – REFO4MD Reference (Clock) Out 4 Disable

Value	Description
1	Disables the Reference (clock) out 4
0	Enables the Reference (clock) out 4

Bit 30 – REFO3MD Reference (Clock) Out 3 Disable

Value	Description
1	Disables the Reference (clock) out 3
0	Enables the Reference (clock) out 3

Bit 29 – REFO2MD Reference (Clock) Out 2 Disable

Value	Description
1	Disables the Reference (clock) out 2
0	Enables the Reference (clock) out 2

Bit 28 – REFO1MD Reference (Clock) Out 1 Disable

Value	Description
1	Disables the Reference (clock) out 1
0	Enables the Reference (clock) out 1

Bit 25 – REFO6MD Reference (Clock) Out 6 Disable

Value	Description
1	Disables the Reference (clock) out 6
0	Enables the Reference (clock) out 6

Bit 24 – REFO5MD Reference (Clock) Out 5 Disable

Value	Description
1	Disables the Reference (clock) out 5
0	Enables the Reference (clock) out 5

23.5.3 PMD3 - Peripheral Module Disable 3 Register

Name: PMD3
Offset: 0x00E0
Reset: 0x00000000
Property: -

Note: This register's bits are only writable when CFGCON0.PMDLOCK = 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
		TCC2MD	TCC1MD	TCC0MD	TC7MD	TC6MD	TC5MD	TC4MD
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	7	6	5	4	3	2	1	0
	TC3MD	TC2MD	TC1MD	TC0MD	DACMD	SER2MD	SER1MD	SER0MD
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 14 – TCC2MD TCC2 Module Disable

Value	Description
1	Disables the TCC2 module
0	Enables the TCC2 module

Bit 13 – TCC1MD TCC1 Module Disable

Value	Description
1	Disables the TCC1 module
0	Enables the TCC1 module

Bit 12 – TCC0MD TCC0 Module Disable

Value	Description
1	Disables the TCC0 module
0	Enables the TCC0 module

Bit 11 – TC7MD TC7 Module Disable

Value	Description
1	Disables the TC7 module
0	Enables the TC7 module

Bit 10 – TC6MD TC6 Module Disable

Value	Description
1	Disables the TC6 module
0	Enables the TC6 module

Bit 9 – TC5MD TC5 Module Disable

Value	Description
1	Disables the TC5 module
0	Enables the TC5 module

Bit 8 – TC4MD TC4 Module Disable

Value	Description
1	Disables the TC4 module
0	Enables the TC4 module

Bit 7 – TC3MD TC3 Module Disable

Value	Description
1	Disables the TC3 module
0	Enables the TC3 module

Bit 6 – TC2MD TC2 Module Disable

Value	Description
1	Disables the TC2 module
0	Enables the TC2 module

Bit 5 – TC1MD TC1 Module Disable

Value	Description
1	Disables the TC1 module
0	Enables the TC1 module

Bit 4 – TC0MD TC0 Module Disable

Value	Description
1	Disables the TC0 module
0	Enables the TC0 module

Bit 3 – DACMD DAC Module Disable

Value	Description
1	Disables the DAC module
0	Enables the DAC module

Bit 2 – SER2MD SERCOM 2 Module Disable

Value	Description
1	Disables the SERCOM 2 module
0	Enables the SERCOM 2 module

Bit 1 – SER1MD SERCOM 1 Module Disable

Value	Description
1	Disables the SERCOM 1 module
0	Enables the SERCOM 1 module

Bit 0 – SER0MD SERCOM 0 Module Disable

Value	Description
1	Disables the SERCOM 0 module
0	Enables the SERCOM 0 module

24. Peripheral Access Controller (PAC)

24.1 Overview

The Peripheral Access Controller (PAC) provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations that might happen when accessing a peripheral:

- Write-protected access
- Illegal access
- Enable-protected access
- Access when clock synchronization or software reset is ongoing

These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the client bus level when an access to a non-existent address is detected.

Note: The modules attached to the PB-Bridge-D bridge and wireless subsystem, as well as RTCC and DSCON are excluded from PAC. The protection mechanism described in CFG protects critical system registers. See *System Configuration and Register Locking (CFG)* from Related Links.

Related Links

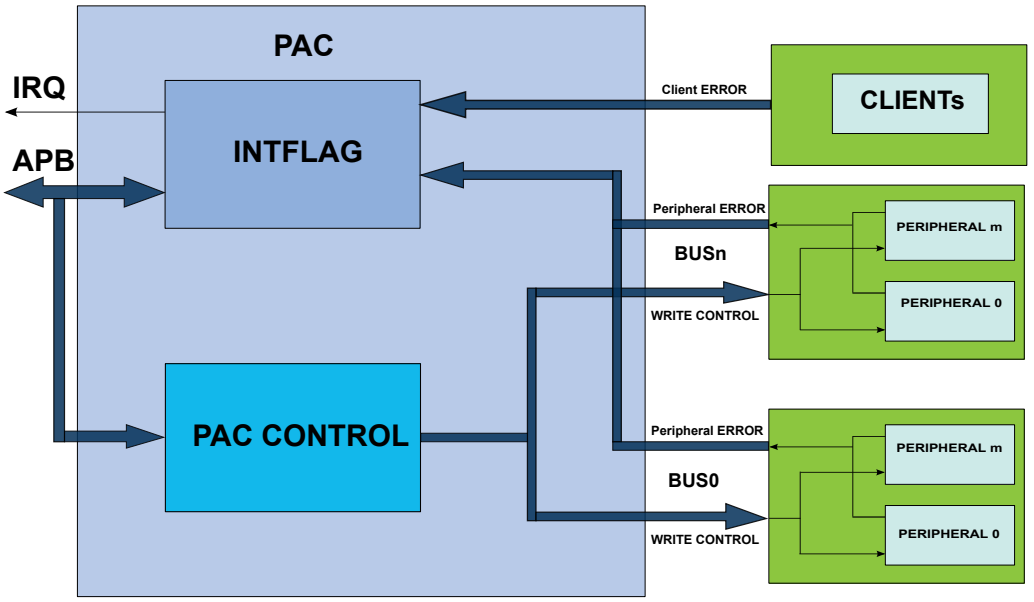
22. System Configuration and Register Locking (CFG)

24.2 Features

- Manages Write Protection Access and Reports Access Errors for the Peripheral Modules or Bridges
- Manages Security Attribution for the Peripheral Modules

24.3 Block Diagram

Figure 24-1. PAC Block Diagram



24.4 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

24.4.1 IO Lines

Not applicable.

24.4.2 Power Management

The PAC can continue to operate in any sleep modes (Idle, Standby Sleep) where the selected source clock is running. The PAC interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting Sleep modes.

24.4.3 DMA

Not applicable.

24.4.4 Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). Using the PAC interrupt requires the NVIC to be configured first.

Table 24-1. Interrupt Lines

Instances	NVIC Line
PAC	PACERR

24.4.5 Events

The events are connected to the Event System, which may need configuration. See *Event System (EVSYS)* from Related Links.

Related Links

30. Event System (EVSYS)

24.4.6 Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

24.4.7 Register Access Protection

All registers with write access can be write protected optionally by the PAC, except for the following PAC registers:

- Write Control (WRCTRL) register
- AHB Client Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write protection by the 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.

24.5 Functional Description

24.5.1 Principle of Operation

The Peripheral Access Control module allows the user to set a write protection on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral's protection can be set, cleared or locked at the user discretion. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, client bus errors can be also reported in the cases where reserved area is accessed by the application.

24.5.2 Basic Operation

24.5.2.1 Initialization, Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

24.5.2.2 Operations

The PAC module allows the user to set, clear or lock the write protection status of all peripherals on all Peripheral Bridges, except the peripherals on PB-Bridge-D bus.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user of the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. See *Peripheral Access Errors* from Related Links.

The PAC module also reports the errors occurring at the client bus level when an access to a reserved area is detected. The AHB Subordinate Bus Interrupt Flag register (INTFLAGAHB) informs the user of the status of the violation in the corresponding client. See *AHB Subordinate Bus Errors* from Related Links.

Related Links

24.5.2.6. AHB Subordinate Bus Errors
24.5.2.3. Peripheral Access Errors

24.5.2.3 Peripheral Access Errors

The following events generate a Peripheral Access Error:

- Protected write – To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as PAC Write-Protection in the module's data sheet can be protected. If a peripheral is not write protected, write data accesses are performed as usual. If a peripheral is write protected and if a write access is attempted, data will not be written and the peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register is set.
- Illegal access – Access to an unimplemented register within the module
- Synchronized write error – For write-synchronized registers, an error is reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt is requested if the PAC interrupt enable bit is set.

24.5.2.4 Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields:

- WRCTRL.PERID – An unique identifier corresponding to a peripheral
- WRCTRL.KEY – A key value that defines the operation to be done on the control access bit These operations can be:
  - Clear protection – Removes the write-access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.
  - Set protection – Sets the write-access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with the write-protection property in this peripheral.
  - Set and lock protection bit – Sets the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

24.5.2.5 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register effectively changes the access protection. Other types of accesses have no effect and cause a PAC write access error. This error is reported in the INTFLAGA.PAC bit.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected, the PAC returns an error and does so similarly for a double clear protection operation.

In addition, an error is generated when writing a set and lock protection to a write-protected peripheral or when a write access is done to a locked set protection. This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect. However, in applications where a write-protected peripheral is used in several contexts, for example, interrupt, care must be taken so that either the interrupt cannot happen while the main application or other interrupt levels manipulate the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (for example, key error, double protect error and so on) set the INTFLAGA.PAC flag.

24.5.2.6 AHB Subordinate Bus Errors

The PAC module reports errors occurring at the AHB Subordinate bus level. These errors are generated when an access is performed at an address where no subordinate (bridge or peripheral) is mapped. These errors are reported in the corresponding bits of the INTFLAGAHB register.

24.5.2.7 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag register bits is set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set to '1'.

24.5.3 DMA Operation

Not applicable.

24.5.4 Interrupts

The PAC has the following interrupt source:

- Error (ERR) – Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC.
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers 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. 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 PAC is reset. All interrupt requests from the peripheral are ORed together at the system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

24.5.5 Events

The PAC generates the following output event:

- Error (ERR) – Generated when one of the interrupt flag registers bits is set

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

24.5.6 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus host (CPU, DMA) is running. The PAC continues to catch access errors from the module and generate interrupts or events.

24.5.7 Synchronization

Not applicable.

24.6 Register Summary

See the *PAC* module in *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	WRCTRL	7:0	PERID[7:0]							
		15:8	PERID[15:8]							
		23:16	KEY[7:0]							
		31:24								
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	7:0	PB-B	PB-A	PFLASH	CFLASH	SRAM3	SRAM2	SRAM1	SRAM0
		15:8				CRYPTO	BOOTROM	QSPI	PB-D	PB-C
		23:16								
		31:24								
0x14	INTFLAGA	7:0	TC2	TC1	TC0	SERCOM1	SERCOM0	EIC	FREQM	PAC
		15:8	TCC2	TCC1	TCC0	TC7	TC6	TC5	TC4	TC3
		23:16								
		31:24								
0x18	INTFLAGB	7:0				RAMECC	EVSYS	DMAC		DSU
		15:8								
		23:16								
		31:24								
0x1C	INTFLAGC	7:0	AC	CCL					SERCOM2	QSPI
		15:8							HMTX	
		23:16								
		31:24								
0x20 ... 0x33	Reserved									
0x34	STATUSA	7:0	TC2	TC1	TC0	SERCOM1	SERCOM0	EIC	FREQM	PAC
		15:8	TCC2	TCC1	TCC0	TC7	TC6	TC5	TC4	TC3
		23:16								
		31:24								
0x38	STATUSB	7:0				RAMECC	EVSYS	DMAC		DSU
		15:8								
		23:16								
		31:24								
0x3C	STATUSC	7:0	AC	CCL					SERCOM2	QSPI
		15:8							HMTX	
		23:16								
		31:24								

Related Links

8. Product Memory Mapping Overview

24.7 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 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

24.4.7. Register Access Protection

24.7.1 Write Control

Name: WRCTRL
Offset: 0x00
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0] Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock the peripheral write control until the next hardware reset

Bits 15:0 – PERID[15:0] Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. Formula to calculate the peripheral identifier:
$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$

Where, BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number. For example, PAC peripheral belongs to Peripheral Bridge A at the '0' bit position (see *INTFLAGA* from Related Links). Therefore, PERID = 32*0+0 = 0 for PAC peripheral.

Table 24-2. PERID Values

Peripheral Bridge Name	BridgeNumber	PERID Values
A	0	0+N
B	1	32+N
C	2	64+N

Related Links
24.7.6. INTFLAGA

24.7.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								ERREO
Access								RW/-/RW
Reset								0

Bit 0 – ERREO Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, an event is generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set.

Value	Description
0	Peripheral Access Error Event Output is disabled.
1	Peripheral Access Error Event Output is enabled.

24.7.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
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 can also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								RW
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

24.7.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
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 can also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								RW
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Enable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request is generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

24.7.5 Bridge Interrupt Flag Status

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

These flags are cleared by writing a '1' to the corresponding bit.

These flags are set when an access error is detected by the corresponding AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

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
				CRYPTO	BOOTROM	QSPI	PB-D	PB-C
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PB-B	PB-A	PFLASH	CFLASH	SRAM3	SRAM2	SRAM1	SRAM0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 12 – CRYPTO Interrupt Flag for Crypto
This flag is set when an access error is detected by the Crypto AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the crypto interrupt flag.

Bit 11 – BOOTROM Interrupt Flag for Boot ROM
This flag is set when an access error is detected by the Boot ROM Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the Boot ROM interrupt flag.

Bit 10 – QSPI Interrupt Flag for QSPI
This flag is set when an access error is detected by the QSPI AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the QSPI interrupt flag.

Bit 9 – PB-D Interrupt Flag for PB-Bridge-D
This flag is set when an access error is detected by the PB-D AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.

Writing a '1' to this bit clears the PB-D interrupt flag.

Bit 8 – PB-C Interrupt Flag for PB-C (PB-Bridge-C)
This flag is set when an access error is detected by the PB-C Bridge AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the PB-C interrupt flag.

Bit 7 – PB-B Interrupt Flag for PB-B (PB-Bridge-B)
This flag is set when an access error is detected by the PB-B Bridge AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the PB-B interrupt flag.

Bit 6 – PB-A Interrupt Flag for PB-A (PB-Bridge-A)
This flag is set when an access error is detected by the PB-A Bridge AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the PB-A interrupt flag.

Bit 5 – PFLASH Interrupt Flag for PFLASH (Peripheral Flash)
This flag is set when an access error is detected by the PFLASH AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the PFLASH interrupt flag.

Bit 4 – CFLASH Interrupt Flag for CFLASH (CPU Flash)
This flag is set when an access error is detected by the CFLASH AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the CFLASH interrupt flag.

Bit 3 – SRAM3 Interrupt Flag for SRAM3
This flag is set when an access error is detected by the SRAM3 AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the SRAM3 interrupt flag.

Bit 2 – SRAM2 Interrupt Flag for SRAM2
This flag is set when an access error is detected by the SRAM2 AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the SRAM2 interrupt flag.

Bit 1 – SRAM1 Interrupt Flag for SRAM1
This flag is set when an access error is detected by the SRAM1 AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit clears the SRAM1 interrupt flag.

Bit 0 – SRAM0 Interrupt Flag for SRAM0
This flag is set when an access error is detected by the SRAM0 AHB Client, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.

Writing a '1' to this bit clears the SRAM0 interrupt flag.

24.7.6 Peripheral Interrupt Flag Status - Bridge A

Name: INTFLAGA
Offset: 0x14
Reset: 0x00000000
Property: –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits clears the corresponding INTFLAGx interrupt flag.

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
	TCC2	TCC1	TCC0	TC7	TC6	TC5	TC4	TC3
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TC2	TC1	TC0	SERCOM1	SERCOM0	EIC	FREQM	PAC
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – TCC2 Interrupt Flag for TCC2
This bit is set when a Peripheral Access Error occurs while accessing the TCC2, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 14 – TCC1 Interrupt Flag for TCC1
This bit is set when a Peripheral Access Error occurs while accessing the TCC1, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 13 – TCC0 Interrupt Flag for TCC0
This bit is set when a Peripheral Access Error occurs while accessing the TCC0, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 12 – TC7 Interrupt Flag for TC7
This bit is set when a Peripheral Access Error occurs while accessing the TC7, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 11 – TC6 Interrupt Flag for TC6
This bit is set when a Peripheral Access Error occurs while accessing the TC6, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 10 – TC5 Interrupt Flag for TC5
This bit is set when a Peripheral Access Error occurs while accessing the TC5, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 9 – TC4 Interrupt Flag for TC4
This bit is set when a Peripheral Access Error occurs while accessing the TC4, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 8 – TC3 Interrupt Flag for TC3
This bit is set when a Peripheral Access Error occurs while accessing the TC3, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 7 – TC2 Interrupt Flag for TC2
This bit is set when a Peripheral Access Error occurs while accessing the TC2, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 6 – TC1 Interrupt Flag for TC1
This bit is set when a Peripheral Access Error occurs while accessing the TC1, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 5 – TC0 Interrupt Flag for TC0
This bit is set when a Peripheral Access Error occurs while accessing the TC0, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 4 – SERCOM1 Interrupt Flag for SERCOM1
This bit is set when a Peripheral Access Error occurs while accessing the SERCOM1, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 3 – SERCOM0 Interrupt Flag for SERCOM0
This bit is set when a Peripheral Access Error occurs while accessing the SERCOM0, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 2 – EIC Interrupt Flag for EIC
This bit is set when a Peripheral Access Error occurs while accessing the EIC, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 1 – FREQM Interrupt Flag for FREQM
This bit is set when a Peripheral Access Error occurs while accessing the FREQM, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

Bit 0 – PAC Interrupt Flag for PAC
This bit is set when a Peripheral Access Error occurs while accessing the PAC, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the interrupt flag.

24.7.7 Peripheral Interrupt Flag Status – Bridge B

Name: INTFLAGB
Offset: 0x18
Reset: 0x00000000
Property: –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits clears the corresponding INTFLAGx interrupt flag.

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
				RAMECC	EVSYS	DMAC		DSU
Access				RW	RW	RW		RW
Reset				0	0	0		0

Bit 4 – RAMECC Interrupt Flag for RAMECC
This flag is set when a Peripheral Access Error occurs while accessing the RAMECC, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the RAMECC interrupt flag.

Bit 3 – EVSYS Interrupt Flag for EVSYS
This flag is set when a Peripheral Access Error occurs while accessing the EVSYS, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the EVSYS interrupt flag.

Bit 2 – DMAC Interrupt Flag for DMAC
This flag is set when a Peripheral Access Error occurs while accessing the DMAC, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DMAC interrupt flag.

Bit 0 – DSU Interrupt Flag for DSU
This flag is set when a Peripheral Access Error occurs while accessing the DSU, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DSU interrupt flag.

24.7.8 Peripheral Interrupt Flag Status - Bridge C

Name: INTFLAGC
Offset: 0x1C
Reset: 0x00000000
Property: –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits clears the corresponding INTFLAGx interrupt flag.

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
							HMTX	
Access							RW	
Reset							0	

Bit	7	6	5	4	3	2	1	0
	AC	CCL					SERCOM2	QSPI
Access	RW	RW					RW	RW
Reset	0	0					0	0

Bit 9 – HMTX HMATRIX APB Protection Enable
This flag is set when a Peripheral Access Error occurs while accessing the HMATRIX, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the HMATRIX interrupt flag.

Bit 7 – AC Interrupt Flag for AC
This flag is set when a Peripheral Access Error occurs while the AC, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the AC interrupt flag.

Bit 6 – CCL Interrupt Flag for CCL
This flag is set when a Peripheral Access Error occurs while accessing the CCL, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CCL interrupt flag.

Bit 1 – SERCOM2 Interrupt Flag for SERCOM2
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM2, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the SERCOM2 interrupt flag.

Bit 0 – QSPI Interrupt Flag for QSPI
This flag is set when a Peripheral Access Error occurs while accessing the QSPI, and an interrupt request is generated if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the QSPI interrupt flag.

24.7.9 Peripheral Write Protection Status A

Name: STATUSA
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection

Writing to this register has no effect.

Reading the STATUS register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected
1	Peripheral is write 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
	TCC2	TCC1	TCC0	TC7	TC6	TC5	TC4	TC3
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
	TC2	TC1	TC0	SERCOM1	SERCOM0	EIC	FREQM	PAC
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – TCC2 TCC2 APB Protect Enable

Value	Description
0	TCC2 peripheral is not write protected
1	TCC2 peripheral is write protected

Bit 14 – TCC1 TCC1 APB Protect Enable

Value	Description
0	TCC1 peripheral is not write protected
1	TCC1 peripheral is write protected

Bit 13 – TCC0 TCC0 APB Protect Enable

Value	Description
0	TCC0 peripheral is not write protected
1	TCC0 peripheral is write protected

Bit 12 – TC7 TC7 APB Protect Enable

Value	Description
0	TC7 peripheral is not write protected
1	TC7 peripheral is write protected

Bit 11 – TC6 TC6 APB Protect Enable

Value	Description
0	TC6 peripheral is not write protected
1	TC6 peripheral is write protected

Bit 10 – TC5 TC5 APB Protect Enable

Value	Description
0	TC5 peripheral is not write protected
1	TC5 peripheral is write protected

Bit 9 – TC4 TC4 APB Protect Enable

Value	Description
0	TC4 peripheral is not write protected
1	TC4 peripheral is write protected

Bit 8 – TC3 TC3 APB Protect Enable

Value	Description
0	TC3 peripheral is not write protected
1	TC3 peripheral is write protected

Bit 7 – TC2 TC2 APB Protect Enable

Value	Description
0	TC2 peripheral is not write protected
1	TC2 peripheral is write protected

Bit 6 – TC1 TC1 APB Protect Enable

Value	Description
0	TC1 peripheral is not write protected
1	TC1 peripheral is write protected

Bit 5 – TC0 TC0 APB Protect Enable

Value	Description
0	TC0 peripheral is not write protected
1	TC0 peripheral is write protected

Bit 4 – SERCOM1 SERCOM1 APB Protect Enable

Value	Description
0	SERCOM1 peripheral is not write protected
1	SERCOM1 peripheral is write protected

Bit 3 – SERCOM0 SERCOM0 APB Protect Enable

Value	Description
0	SERCOM0 peripheral is not write protected
1	SERCOM0 peripheral is write protected

Bit 2 – EIC EIC APB Protect Enable

Value	Description
0	EIC peripheral is not write protected
1	EIC peripheral is write protected

Bit 1 – FREQM FREQM APB Protect Enable

Value	Description
0	FREQM peripheral is not write protected
1	FREQM peripheral is write protected

Bit 0 – PAC PAC APB Protect Enable

Value	Description
0	PAC peripheral is not write protected

Value	Description
1	PAC peripheral is write protected

24.7.10 Peripheral Write Protection Status - Bridge B

Name: STATUSB
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection

Writing to this register has no effect.

Reading the STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected
1	Peripheral is write 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								
Reset								

Bit	7	6	5	4	3	2	1	0
				RAMECC	EVSYS	DMAC		DSU
Access				R	R	R		R
Reset				0	0	0		0

Bit 4 – RAMECC RAMECC APB Protect Enable

Value	Description
0	RAMECC peripheral is not write protected
1	RAMECC peripheral is write protected

Bit 3 – EVSYS EVSYS APB Protect Enable

Value	Description
0	EVSYS peripheral is not write protected
1	EVSYS peripheral is write protected

Bit 2 – DMAC DMAC APB Protect Enable

Value	Description
0	DMAC peripheral is not write protected
1	DMAC peripheral is write protected

Bit 0 – DSU DSU APB Protect Enable

Value	Description
0	DSU peripheral is not write protected
1	DSU peripheral is write protected

24.7.11 Peripheral Write Protection Status - Bridge C

Name: STATUSC
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protection

Writing to this register has no effect.

Reading the STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected
1	Peripheral is write 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
							HMTX	
Access							R	
Reset							0	

Bit	7	6	5	4	3	2	1	0
	AC	CCL					SERCOM2	QSPI
Access	R	R					R	R
Reset	0	0					0	0

Bit 9 – HMTX HMATRIX APB Protection Enable

Value	Description
0	HMATRIX APB is not write protected
1	HMATRIX APB is write protected

Bit 7 – AC AC APB Protection Enable

Value	Description
0	AC peripheral is not write protected
1	AC peripheral is write protected

Bit 6 – CCL CCL APB Protection Enable

Value	Description
0	CCL peripheral is not write protected
1	CCL peripheral is write protected

Bit 1 – SERCOM2 SERCOM2 APB Protection Enable

Value	Description
0	SERCOM2 peripheral is not write protected
1	SERCOM2 peripheral is write protected

Bit 0 – QSPI QSPI APB Protection Enable

Value	Description
0	QSPI peripheral is not write protected
1	QSPI peripheral is write protected

25. Real-Time Counter and Calendar (RTCC)

25.1 Overview

The Real-Time Counter (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, overflow wake-up mechanisms or from the wake inputs.

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 overflow 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 30.5 µs, 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.

25.2 Features

- 32-Bit Counter with 10-Bit Prescaler
- Multiple Clock Sources
- 32-Bit or 16-Bit Counter Mode
- Two 32-Bit or Four 16-Bit Compare Values
- 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

25.3 Block Diagram

Figure 25-1. RTCC Block Diagram (Mode 0 — 32-Bit Counter)

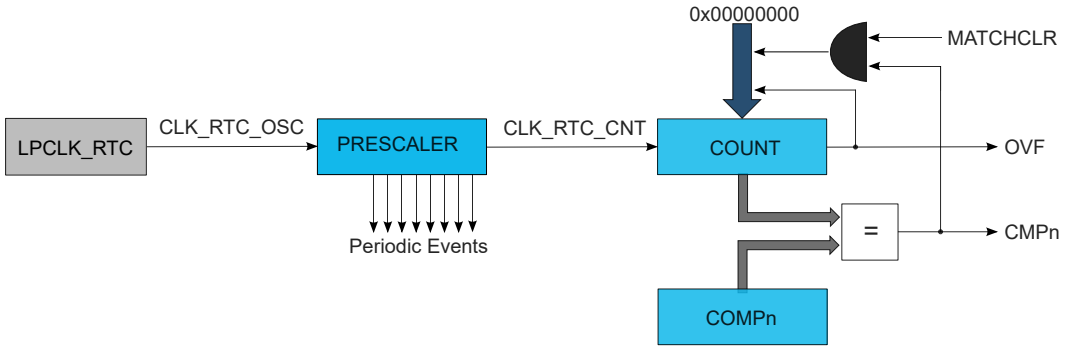


Figure 25-2. RTCC Block Diagram (Mode 1 — 16-Bit Counter)

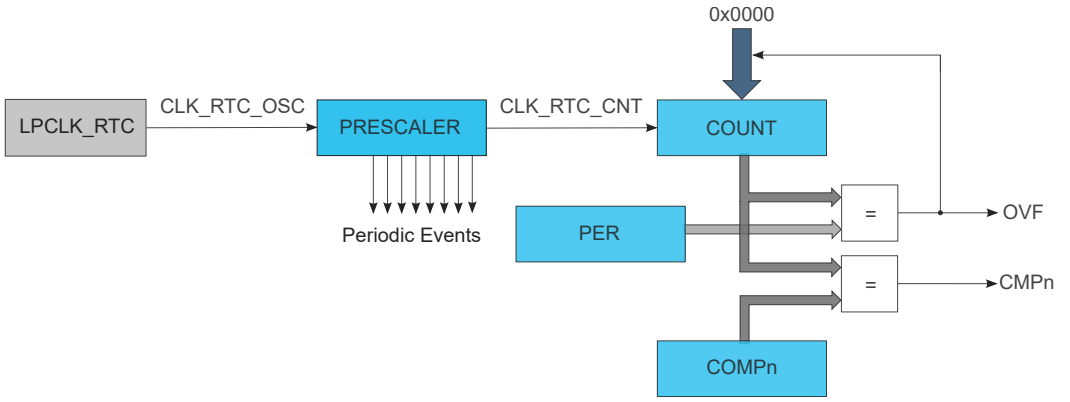


Figure 25-3. RTCC Block Diagram (Mode 2 — Clock/Calendar)

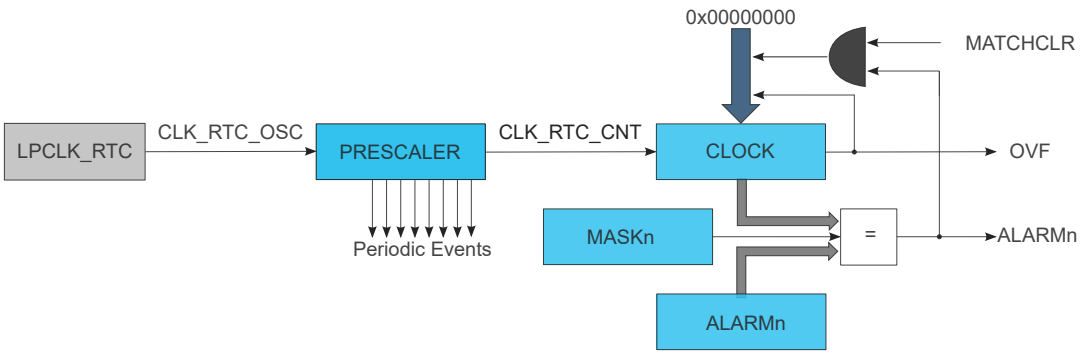
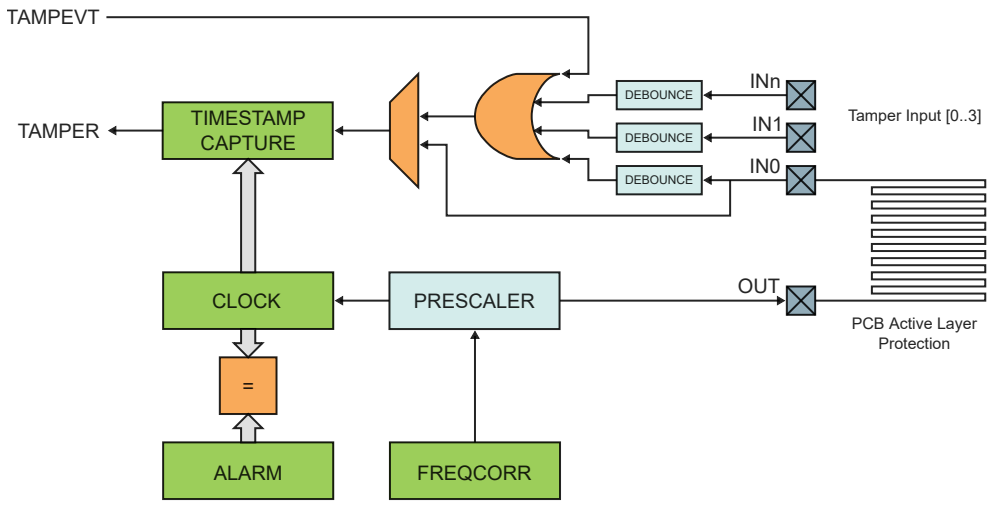


Figure 25-4. RTCC Block Diagram (Tamper Detection)



25.4 Signal Description

Table 25-1. Signal Description

Signal	Description	Type
INn [n=0..3]	Tamper detection input	Digital input
OUT	Tamper detection output	Digital output
RTC_EVENT	RTC event output	Digital output

25.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

25.5.1 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)

25.5.2 Power Management

The RTC continues to operate in any sleep modes (Standby Sleep, Deep Sleep, Idle) 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 the 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

18. Power Management Unit (PMU)

25.5.3 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 CFGCON4.VBKP_32KCSEL bit. The 1 KHz clock source is based on the mux selection controlled by the CFGCON4.VBKP_1KCSEL 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.

25.5.4 DMA

The DMA request lines (or line if only one request) are connected to the DMA Controller (DMAC). Using the RTC DMA requests requires the DMA Controller to be configured first. See *Direct Memory Access Controller (DMAC)* from Related Links.

Related Links

26. Direct Memory Access Controller (DMAC)

25.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

25.5.6 Events

The events are connected to the *Event System*. See *Event System (EVSYS)* from Related Links.

Related Links

30. Event System (EVSYS)

25.5.7 Debug Operation

When the CPU is halted in Debug mode, the RTC halts normal operation. The RTC can be forced to continue operation during debugging. See *DBGCTRL* from Related Links.

Related Links

25.8.7. DBGCTRL

25.5.8 Register Access Protection

All registers with write access are optionally write protected by the PAC, except the Interrupt Flag Status and Clear (INTFLAG) register. Write protection is denoted by the PAC Write-Protection 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

24. Peripheral Access Controller (PAC)

25.6 Functional Description

25.6.1 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

25.6.2 Basic Operation

25.6.2.1 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:

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- Tamper Control register (TAMPCTRL)

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, then, check whether the write synchronization has finished, then, change the desired bit field value. Enable-protected bits in the 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 the Clock/Calendar mode, the prescaler must be configured to provide a 1 Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is derived by the following formula:

$$f_{\mathrm{CLK\_RTC\_CNT}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{\mathrm{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK\_RTC\_OSC}$, and $f_{CLK\_RTC\_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

25.6.2.2 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.

25.6.2.3 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 the *RTC Block Diagram (Mode 0-32-Bit Counter)* figure in the *Block Diagram* from Related Links. When the RTC is enabled, the counter is incremented on every 0-to-1 transition of CLK_RTC_CNT. The counter increments until it reaches the top value of 0xFFFFFFFF, then wraps 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 registers (COMPn, n=0–1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) 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 COMPn 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.CMPn and INTFLAG.OVF will both be set simultaneously on a compare match with COMPn.

Related Links

25.3. Block Diagram

25.6.2.4 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

25.3. Block Diagram

25.6.2.5 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 the *RTC Block Diagram (Mode 2 — Clock/Calendar)* figure in the *Block Diagram* from Related Links. When the RTC is enabled, the counter increments on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1 Hz 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 and so on)
- 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 increments until it reaches the top value of 23:59:59 December 31 of year value 0x3F, then wraps to 00:00:00 January 1 of year value 0x00. This sets 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 registers (ALARMn, n=0–1). When an alarm match occurs, the Alarm n Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARMn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT. For example, for a 1 Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the alarm match occurs.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm n Mask register (MASKn.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 ALARMn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than 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 ALARMn.

Related Links
25.3. Block Diagram
25.6.8.1. Periodic Intervals

25.6.3 DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

25.6.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF) – Indicates that the counter has reached its top value and wrapped to zero
- Tamper (TAMPER) – Indicates detection of valid signal on a tamper input pin or tamper event input
- 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

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

9.2. Nested Vector Interrupt Controller (NVIC)
25.6.8.1. Periodic Intervals

25.6.5 Events

The RTC can generate the following output events and can be used by the EVSYS module:

- Overflow (OVF) – Generated when the counter has reached its top value and wrapped to zero
- Tamper (TAMPER): Generated on detection of a valid signal on a tamper input pin or tamper event input
- 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 a 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 signals can be out at a time in pin-limited variants. The selection between RTC_OUT and RTC_EVENT is configurable through the CFGCON4.RTCEVTYPE bit.

Related Links

25.6.8.1. Periodic Intervals
25.6.8.5. Tamper Detection
30. Event System (EVSYS)

25.6.6 Sleep Mode Operation

The RTC continues to operate in any Sleep modes (Standby Sleep, Deep Sleep) 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 is generated after the wake-up if the NVIC 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

30. Event System (EVSYS)

25.6.7 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
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)

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 General Purpose n registers (GPn)

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'
- The Timestamp Value register (TIMESTAMP)

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.

25.6.8 Additional Features

25.6.8.1 Periodic Intervals

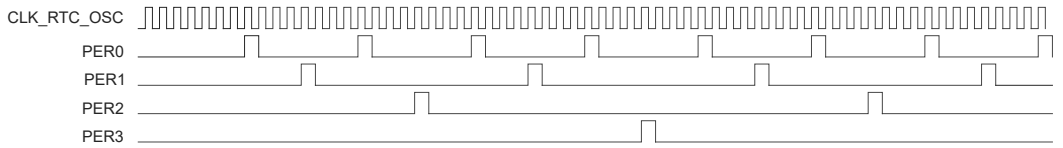
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) 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:

$$f_{\mathrm{PERIODIC(n)}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{n+3}}$$

$f_{CLK\_RTC\_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 generates an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles and so on. This is illustrated in the following figure.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is '0'. Then, no periodic events are generated.

Figure 25-5. Example Periodic Events



Note: This example also applies to interrupts. Just replace EVCTRL.PEREOn with the PERn fields of INTENCLR, INTENSET and INTFLAG. For Modes 0 and 2, n = 0,..7. For Mode 1 n = 2...7.

25.6.8.2 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 1 ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 8192 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 128 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{8192 \cdot 128} \cdot 10^6 \text{ppm}$$

This results in a resolution of 0.95367 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 can also be shortened or lengthened depending on the correction value.

25.6.8.3 Backup Registers

The RTC includes one Backup register (BKUP0). This register maintains its content in the Deep Sleep mode. It is used to store user-defined values.

Use General Purpose registers (GPn) if the stored user-defined data are more than what the Backup register can hold.

25.6.8.4 General Purpose Registers

The RTC includes four General Purpose registers (GPn). These registers are reset only when the RTC is reset or when tamper detection occurs while CTRLA.GPTRST=1 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*n and 2*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 has 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 the COMPARE/ALARM register is written, the write buffer temporarily holds the COMPARE/ALARM value until the synchronization is complete (bit SYNCBUSY.COMPn going to '0'). After the write is completed, the write buffer can be used as an 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 synchronization 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: 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 25-2. 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	—

25.6.8.5 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 is not captured in the TIMESTAMP register.
- Capture – A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn is detected and the tamper interrupt flag (INTFLAG.TAMPER) is set. The RTC value is captured in the TIMESTAMP register.
- Active Layer Protection – A mismatch of an internal RTC signal routed between INn and OUTn pins is detected, and the tamper interrupt flag (INTFLAG.TAMPER) is set. The RTC value is captured in the TIMESTAMP register.

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.

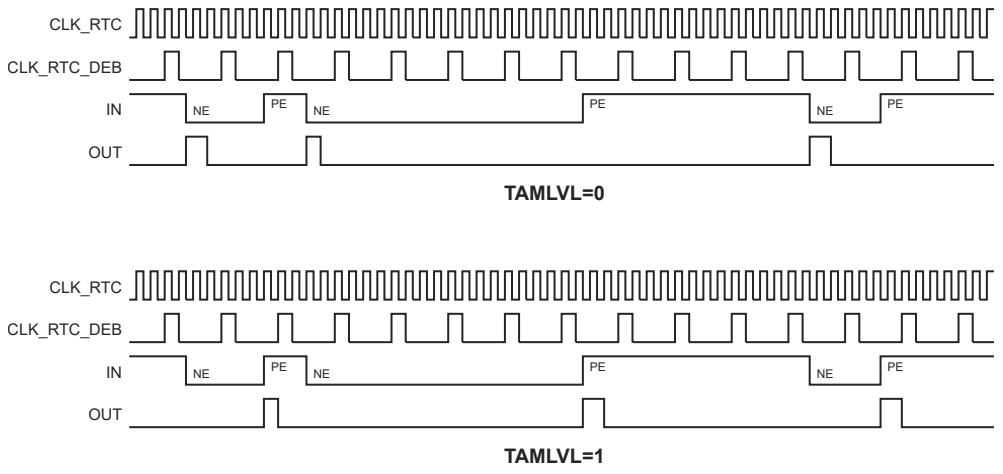
The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVTEI).

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. (In other words, the duration cannot be adjusted separately for each debouncer.)

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. The following figure illustrates an example.

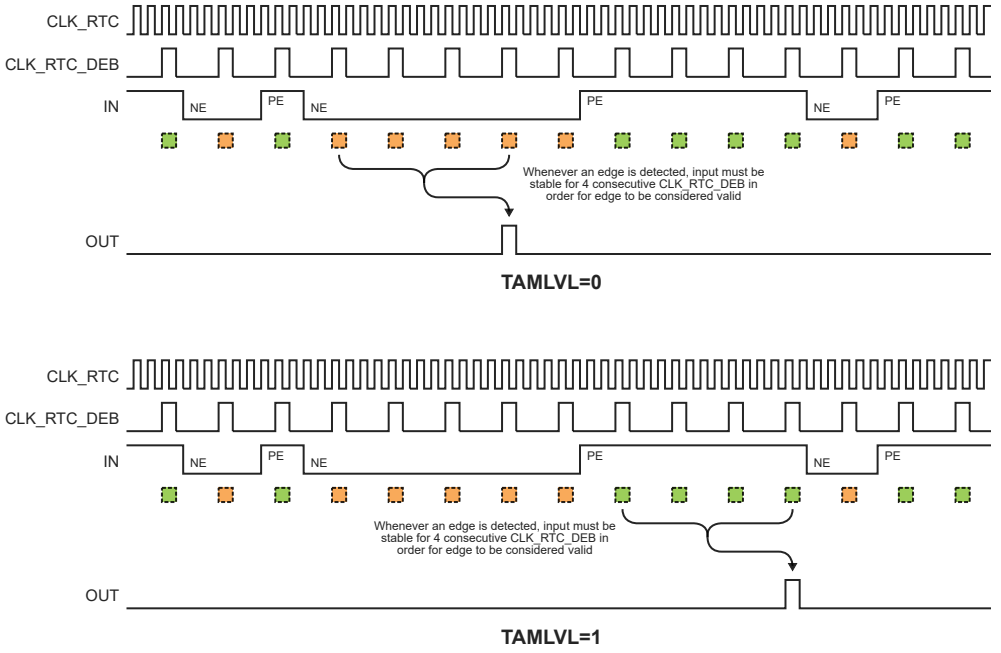
Figure 25-6. Edge Detection with Debouncer Disabled



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 Table 25-3. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

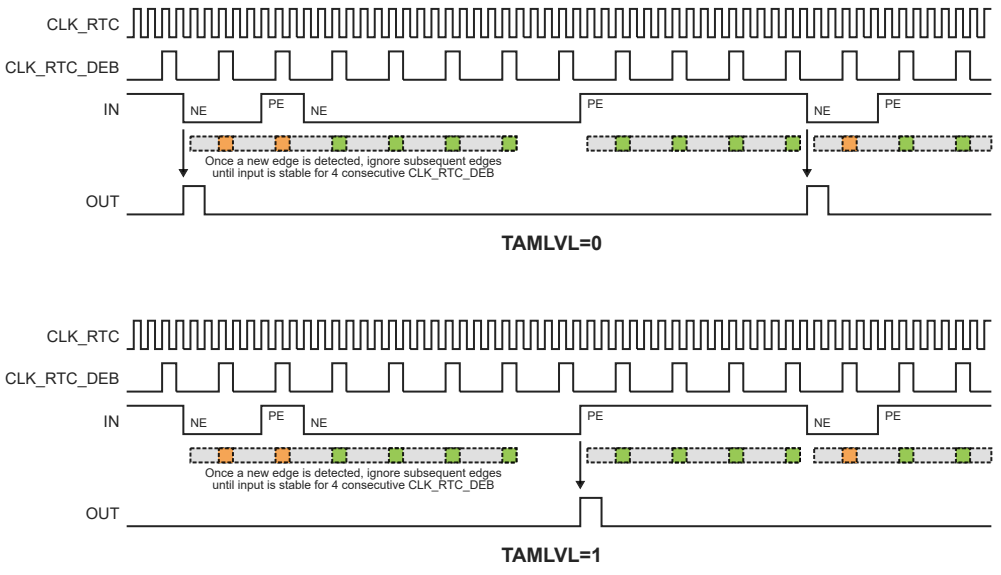
- Synchronous (CTRLB.DEBASYNC = 0): INn is synchronized in two CLK_RTC periods, then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. The following figure illustrates an example.

Figure 25-7. Edge Detection with Synchronous Stability Debouncing



- Asynchronous (CTRLB.DEBASYNC = 1): The first edge on INn is detected. Further detection is blanked until INn remains stable for four CLK_RTC_DEB periods. The following figure illustrates an example.

Figure 25-8. 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. The following figure illustrates an example.

Figure 25-9. Edge Detection with Majority Debouncing

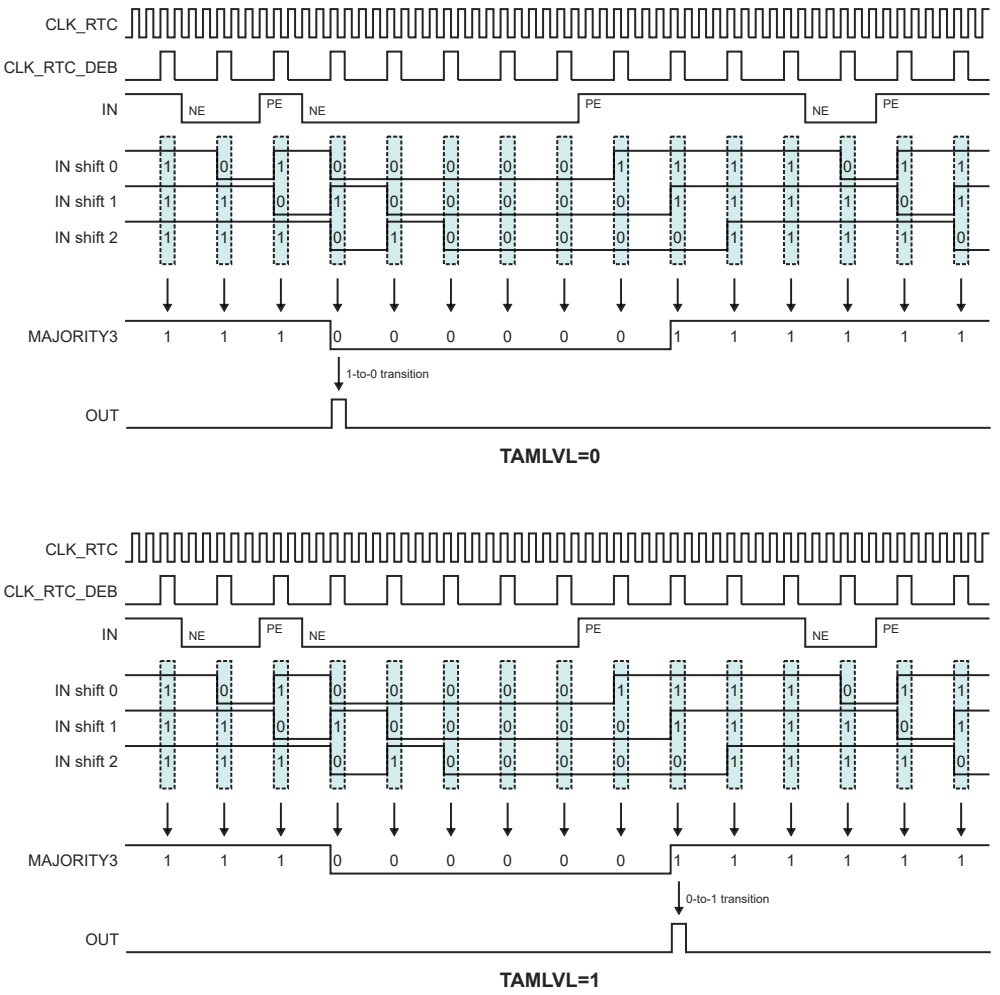


Table 25-3. Debouncer Configuration

TAMPCTRL. DEBNCn	CTRLB. DEBMAJ	CTRLB. DEBASYNC	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 four 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 four consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for three consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = 0 and LHH, HLH, HHL, HHH = 1).

25.6.8.5.1 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 is triggered by the timestamp. 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 is still recorded in TAMPID.

The Tamper Input Event (TAMPEVT) is always performing 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 is still updated.
Note: The TIMESTAMP value must be read once, and INTFLAG.TAMPER must be cleared. The next value must be read only after the INTFLAG.TAMPER is set again.

25.6.8.5.2 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.

25.7 Register Summary - Mode 0 - 32-Bit Counter

See the *RTCC* 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	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ			GP2EN	GP0EN
		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO					CMPEOn[1:0]	
		23:16								TAMPEVEI
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			CMPn[3:0]			
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			CMPn[3:0]			
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			CMPn[3:0]			
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0		COMP1	COMP0		COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16					GP3	GP2	GP1	GP0
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
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 ... 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[0]	IN2ACT[1:0]		IN1ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]
		15:8								IN3ACT[1]
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
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]							

Related Links

8. Product Memory Mapping Overview

25.8 Register Description - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

25.8.1 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	GPTRST	BKTRST		PRESCALER[3:0]			
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	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 prevents 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 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

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

Value	Name	Description
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 writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is 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 is disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation is discarded.
Due to synchronization, there is a delay writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST is cleared when the Reset is complete.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no Reset operation ongoing
1	The Reset operation is ongoing

25.8.2 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	DEBASYNC	DEBMAJ			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

25.8.3 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
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO					CMPEOn[1:0]	
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	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 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled and incoming events will be ignored.
1	Tamper event input is enabled and incoming events will capture the COUNT value.

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 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled and will not be generated.
1	Tamper event output is enabled and will be generated for every tamper input.

Bits 9:8 – CMPEOn[1:0] Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn 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.

25.8.4 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: -

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	TAMPER			CMPn[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
	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 clears the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 11:8 – CMPn[3:0] Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 10:8 – CMPn[2:0] Compare n Interrupt Enable [n = 2..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 8, 9 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bit 8 – CMPn Compare 0 Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare 0 Interrupt Enable bit, which disables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 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 clears 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.

25.8.5 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: -

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	TAMPER			CMPn[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
	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 clears the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 11:8 – CMPn[3:0] Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 10:8 – CMPn[2:0] Compare n Interrupt Enable [n = 2..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 8, 9 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bit 8 – CMPn Compare 0 Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare 0 Interrupt Enable bit, which disables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 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 clears 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.

25.8.6 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			CMPn[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
	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 is 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 is 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 11:8 – CMPn[3:0] Compare n [n = 3..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 10:8 – CMPn[2:0] Compare n [n = 2..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bit 8 – CMPn Compare n [n = 1..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bit 8 – CMP0 Compare 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request is generated if INTENCLR/SET.COMP0 is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare 0 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 is generated if INTENCLR/SET.PERn is '1' .
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.8.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: -

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.

25.8.8 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
					GP3	GP2	GP1	GP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		COMP1	COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access		R	R		R	R	R	R
Reset		0	0		0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

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.

Bits 5, 6 – COMPn Compare n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for COMPx register is complete.
1	Write synchronization for COMPx 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.

25.8.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.FREQCORR must be checked to ensure the FREQCORR register synchronization is complete.

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, in other words., frequency is decreased.
1	The correction value is negative, in other words., frequency is 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.

25.8.10 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Notes:

- This register is read-synchronized when CTRLA.COUNTSYNC=1: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.
- This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

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.

25.8.11 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.

25.8.12 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.

25.8.13 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
25.6.8.4. General Purpose Registers

25.8.14 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
								IN3ACT[1]
Access								
Reset								0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[0]	IN2ACT[1:0]		IN1ACT[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: The 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: The 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 8:7 – IN3ACT[1:0] Tamper Channel 3 Action
These bits determine the action taken by Tamper Channel 3.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 6:5 – IN2ACT[1:0] Tamper Channel 2 Action
These bits determine the action taken by Tamper Channel 2.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 4:3 – IN1ACT[1:0] Tamper Channel 1 Action

These bits determine the action taken by Tamper Channel 1.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

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

25.8.15 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0
Property: -

Bit	31	30	29	28	27	26	25	24
	COUNT[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
	COUNT[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
	COUNT[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
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Count Timestamp Value
The 32-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs

25.8.16 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

25.8.17 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.

25.9 Register Summary - Mode 0 - 32-Bit Counter

See the *RTCC* 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	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ			GP2EN	GP0EN
		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO			CMPEOn[3:0]			
		23:16								TAMPEVEI
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER	CMPn[5:0]					
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0	COMP2	COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							COMP3
		23:16					GP3	GP2	GP1	GP0
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x1A ... 0x1B	Reserved									
0x1C	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1E ... 0x1F	Reserved									
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 ... 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[0]	IN2ACT[1:0]		IN1ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]
		15:8								IN3ACT[1]
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16								
		31:24								
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]							

Related Links

8. Product Memory Mapping Overview

25.10 Register Description - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

25.10.1 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	GPTRST	BKTRST		PRESCALER[3:0]			
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	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 prevents 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 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	GPn registers will not reset when a tamper condition occurs.
1	GPn registers will reset when a tamper condition occurs.

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

Value	Name	Description
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 reads back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is 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 is disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation is discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST is cleared when the Reset is complete.
Note: During a SWRST, access to registers/bits without SWRST is disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no Reset operation ongoing
1	The Reset operation is ongoing

25.10.2 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			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

25.10.3 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
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO			CMPEOn[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
	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 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored
1	Tamper event input is enabled, and incoming events will capture the COUNT value

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 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated.
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 11:8 – CMPEOn[3:0] Compare n Event Output Enable [n = 3..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn 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.

25.10.4 Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: -

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register is also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
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
	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.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9, 10, 11 – CMPn Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n 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.

25.10.5 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: -

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	TAMPER			CMP3	CMP2	CMP1	CMP0
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
	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.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9, 10, 11 – CMPn Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n 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.

25.10.6 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	TAMPER	CMPn[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
	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 is 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
This flag is set after a tamper condition occurs, and an interrupt request is 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 13:8 – CMPn[5:0] Compare n [n = 5..0]
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 is generated if INTENCLR/SET.COMPn is '1' .
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 12:8 – CMPn[4:0] Compare n [n = 4..0]
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 is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 8, 9, 10, 11 – CMPn Compare n [n = 3..0]
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 is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 10:8 – CMPn[2:0] Compare n [n = 2..0]
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 is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 8, 9 – CMPn Compare n [n = 1..0]
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 is generated if INTENCLR/SET.COMPn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bit 8 – CMP0 Compare 0
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 is generated if INTENCLR/SET.COMP0 is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare 0 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 is 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.

25.10.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: -

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.

25.10.8 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
					GP3	GP2	GP1	GP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							COMP3
Access	R							R
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	COMP2	COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

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.

Bits 5, 6, 7, 8 – COMPn Compare n Synchronization Busy Status [n = 3..0]

Value	Description
0	Write synchronization for COMPn register is complete.
1	Write synchronization for COMPn register 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.

Value	Description
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.

25.10.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.FREQCORR must be checked to ensure the FREQCORR register synchronization is complete.

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, in other words., frequency is decreased.
1	The correction value is negative, in other words., frequency is 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.

25.10.10 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Notes:

- This register is read-synchronized when CTRLA.COUNTSYNC=1: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.
- This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

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).

25.10.11 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

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).

25.10.12 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: 0x20 + n*0x02 [n=0..3]
Reset: 0x0000
Property: 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 COMPn 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.CMPn) is set on the next counter cycle.

25.10.13 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
								IN3ACT[1]
Access								
Reset								0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[0]	IN2ACT[1:0]		IN1ACT[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: The 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: The 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 8:7 – IN3ACT[1:0] Tamper Channel 3 Action
These bits determine the action taken by Tamper Channel 3.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 6:5 – IN2ACT[1:0] Tamper Channel 2 Action
These bits determine the action taken by Tamper Channel 2.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 4:3 – IN1ACT[1:0] Tamper Channel 1 Action

These bits determine the action taken by Tamper Channel 1.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

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

25.10.14 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0000
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
	COUNT[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
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Count Timestamp Value
The 16-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs.

25.10.15 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

25.10.16 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.

25.10.17 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
25.6.8.4. General Purpose Registers

25.11 Register Summary - Mode 0 - 32-Bit Counter

See he t*RTCC* 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	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
		15:8	CLOCKSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ			GP2EN	GP0EN
		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO	TAMPEREO					ALARMEO1	ALARMEO0
		23:16								TAMPEVEI
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER					ALARM1	ALARM0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER					ALARM1	ALARM0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF	TAMPER			ALARMn[3:0]			
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0		ALARM1	ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
		15:8	CLOCKSYNC			MASK1	MASK0			
		23:16					GP3	GP2	GP1	GP0
		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]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]							
0x50 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[0]	IN2ACT[1:0]		IN1ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]
		15:8								IN3ACT[1]
		23:16					TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		31:24					DEBNC3	DEBNC2	DEBNC1	DEBNC0
0x64	TIMESTAMP	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]	
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]							

Related Links

8. Product Memory Mapping Overview

25.12 Register Description - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

25.12.1 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	GPTRST	BKTRST		PRESCALER[3:0]			
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	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 prevents 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 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

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

Value	Name	Description
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 match
1	The counter is cleared on a Compare/Alarm 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 reads back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is 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 is disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation is discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST is cleared when the Reset is complete.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no Reset operation ongoing
1	The Reset operation is ongoing

25.12.2 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	DEBASYNC	DEBMAJ			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 – 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 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0 disabled.
1	COMP0 compare function disabled. GP0 enabled.

25.12.3 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
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO					ALARMEO1	ALARMEO0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	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 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored.
1	Tamper event input is enabled, and all incoming events will capture the CLOCK value.

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 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 8, 9 – ALARMEOn Alarm n Event Output Enable [n = 1..0]

Value	Description
0	Alarm n event is disabled and will not be generated.
1	Alarm n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn 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.

25.12.4 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: -

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	TAMPER					ALARM1	ALARM0
Access	R/W	R/W					R/W	R/W
Reset	0	0					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 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.

Bit 14 – TAMPER Tamper Interrupt Enable

Bits 8, 9 – ALARMn Alarm n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm n Interrupt Enable bit, which disables the Alarm n interrupt.

Value	Description
0	The Alarm n interrupt is disabled.
1	The Alarm n 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.

25.12.5 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: -

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	TAMPER					ALARM1	ALARM0
Access	R/W	R/W					R/W	R/W
Reset	0	0					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 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.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt it disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9 – ALARMn Alarm n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm n Interrupt Enable bit, which enables the Alarm n interrupt.

Value	Description
0	The Alarm n interrupt is disabled.
1	The Alarm n 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.

25.12.6 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	TAMPER			ALARMn[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
	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 is 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
This flag is set after a tamper condition occurs, and an interrupt request is 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 11:8 – ALARMn[3:0] Alarm n [n = 3..0]
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 is generated if INTENCLR/SET.ALARMn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm n interrupt flag.

Bits 10:8 – ALARMn[2:0] Alarm n [n = 2..0]
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 is generated if INTENCLR/SET.ALARMn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm n interrupt flag.

Bits 8, 9 – ALARMn Alarm n [n = 1..0]
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 is generated if INTENCLR/SET.ALARMn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm n interrupt flag.

Bit 8 – ALARM0 Alarm 0
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 is generated if INTENCLR/SET.ALARM0 is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm 0 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 is 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.

25.12.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: -

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.

25.12.8 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
					GP3	GP2	GP1	GP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC			MASK1	MASK0			
Access	R			R	R			
Reset	0			0	0			

Bit	7	6	5	4	3	2	1	0
		ALARM1	ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access		R	R		R	R	R	R
Reset		0	0		0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

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.

Bits 11, 12 – MASKn Mask n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for MASKx register is complete.
1	Write synchronization for MASKx register is ongoing.

Bits 5, 6 – ALARMn Alarm n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for ALARMx register is complete.
1	Write synchronization for ALARMx register 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.

25.12.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.FREQCORR must be checked to ensure the FREQCORR register synchronization is complete.

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, in other words., frequency is decreased.
1	The correction value is negative, in other words., frequency is 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.

25.12.10 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Notes:

- This register is read-synchronized and write-synchronized: SYNCBUSY.CLOCK must be checked to ensure the CLOCK register synchronization is complete.
- This register must be written with 32-bit accesses only.

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

25.12.11 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.

25.12.12 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

25.12.13 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
25.6.8.4. General Purpose Registers

25.12.14 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
								IN3ACT[1]
Access								
Reset								0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[0]	IN2ACT[1:0]		IN1ACT[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: The 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: The 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 8:7 – IN3ACT[1:0] Tamper Channel 3 Action
These bits determine the action taken by Tamper Channel 3.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 6:5 – IN2ACT[1:0] Tamper Channel 2 Action
These bits determine the action taken by Tamper Channel 2.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 4:3 – IN1ACT[1:0] Tamper Channel 1 Action

These bits determine the action taken by Tamper Channel 1.

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 OUTn pins or inside the TrustRAM. When a mismatch occurs, capture timestamp and set Tamper flag

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

25.12.15 Timestamp Value

Name: TIMESTAMP
Offset: 0x64
Reset: 0
Property: -

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
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
	MONTH[1:0]		DAY[4:0]					HOUR[4]
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
	HOUR[3:0]				MINUTE[5:2]			
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
	MINUTE[1:0]		SECOND[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The year value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 25:22 – MONTH[3:0] Month
The month value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 21:17 – DAY[4:0] Day
The day value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 16:12 – HOUR[4:0] Hour
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 11:6 – MINUTE[5:0] Minute
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 5:0 – SECOND[5:0] Second
The second value is captured by the TIMESTAMP when a tamper condition occurs.

25.12.16 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

25.12.17 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.

26. Direct Memory Access Controller (DMAC)

26.1 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 that can all 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 grants one DMA channel at a time to act as the active channel. When an active channel is 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 executes the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC writes back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grants the higher prioritized channel to start the transfer as the new active channel. When 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 the 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.

Related Links

26.3. Block Diagram

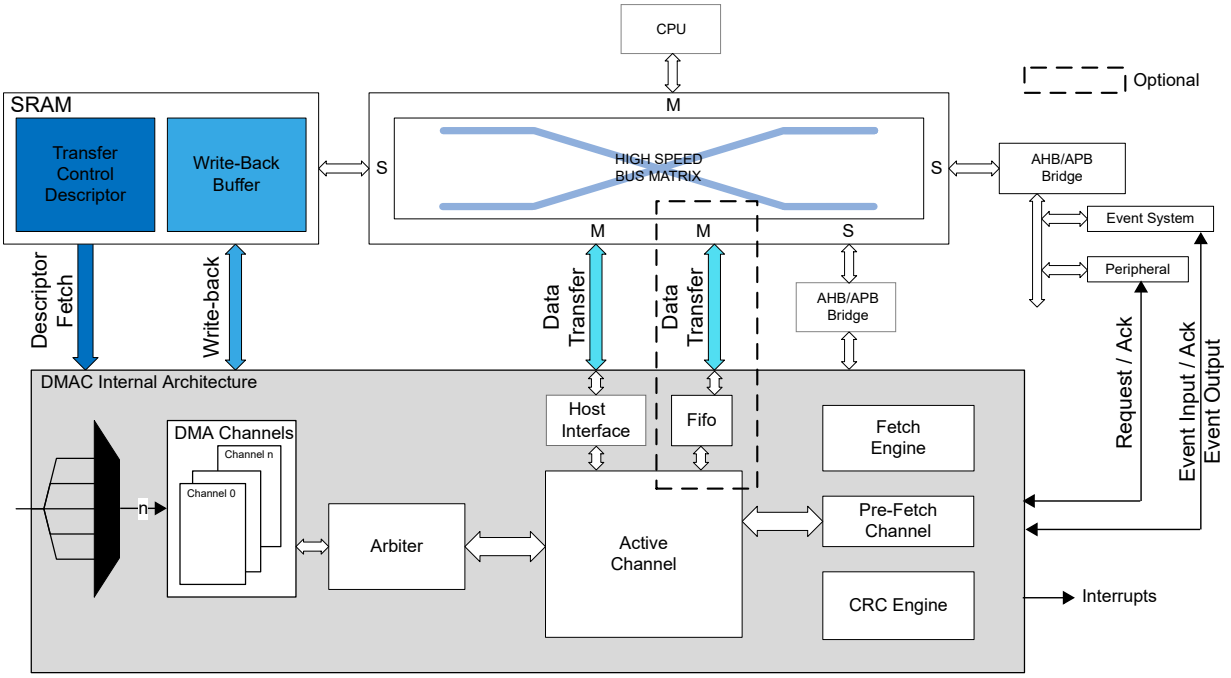
26.2 Features

- 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
  - Four configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256 KB 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
- Eight Event Inputs
  - One event input for each of the eight 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
- Four Event Outputs
  - One output event for each of the four 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)

26.3 Block Diagram

Figure 26-1. DMAC Block Diagram



26.4 Signal Description

Not applicable.

26.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

26.5.1 I/O Lines

Not applicable.

26.5.2 Power Management

The DMAC continues to operate in any sleep modes (Idle, Standby Sleep) where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from the sleep modes. Events connected to the event system can trigger other operations in the system without exiting the sleep modes. On a hardware or software Reset, all registers are set to their Reset value.

26.5.3 Clocks

An AHB clock (SYS_CLK) is required to clock the DMAC. The PB2_CLK is used when writing and reading the I/O registers of the DMAC.

26.5.4 DMA

Not applicable.

26.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

26.5.6 Events

The events are connected to the event system. See *Event System (EVSYS)* from Related Links.

Related Links

30. Event System (EVSYS)

26.5.7 Debug Operation

When the CPU is halted in Debug mode, the DMAC halts normal operation. The DMAC can be forced to continue operation during debugging. See *DBGCTRL* from Related Links.

Related Links

26.8.6. DBGCTRL

26.5.8 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 Pending register (INTPEND)
- 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.

26.5.9 Analog Connections

Not applicable.

26.6 Functional Description

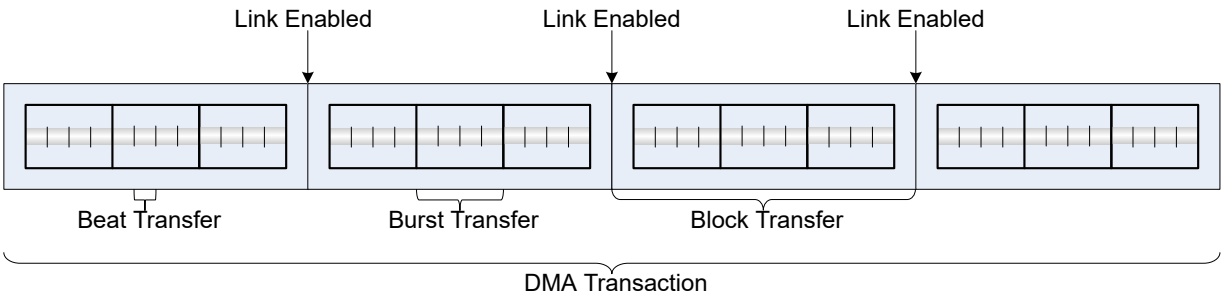
26.6.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

26.6.1.1 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 illustrates the relationship between the different transfer sizes:

Figure 26-2. 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 Figure 26-2. 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 26-2 illustrates 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 results 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 resumes the block transfer when the related 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

26.6.2.3. Transfer Descriptors
26.6.3.1. Linked Descriptors

26.6.1.2 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. Refer to 26.6.3.8. CRC Operation for details.

26.6.2 Basic Operation

26.6.2.1 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 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.

26.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control (CTRL.DMAENABLE) register to '1'. The DMAC is disabled by writing a '0' to the CTRL.DMAENABLE register.

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 CHCTRLAn.ENABLE.

The CRC is enabled by writing a value to the CRC Source bits in the Control register (CRCCTRL.CRCSRC). The CRC is disabled by writing a '0' to CRCCTRL.CRCSRC.

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 the corresponding channel is configured. 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.

26.6.2.3 Transfer Descriptors

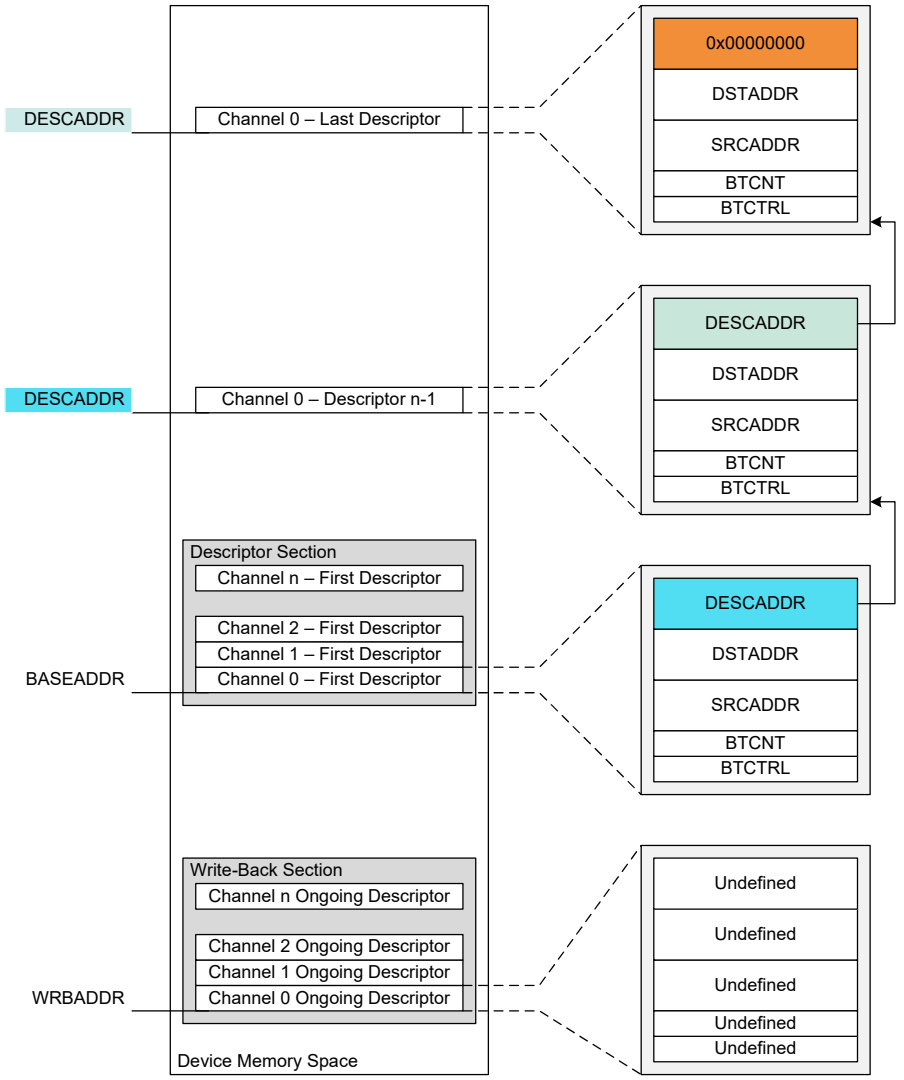
The transfer descriptors, together with the channel configurations, decide how a block transfer must be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one) and receives a transfer trigger, its first transfer descriptor must be initialized and valid (BTCTRL.VALID). The first transfer 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 26-3. 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. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM.

Related Links

26.6.3.1. Linked Descriptors

26.6.2.4 Arbitration

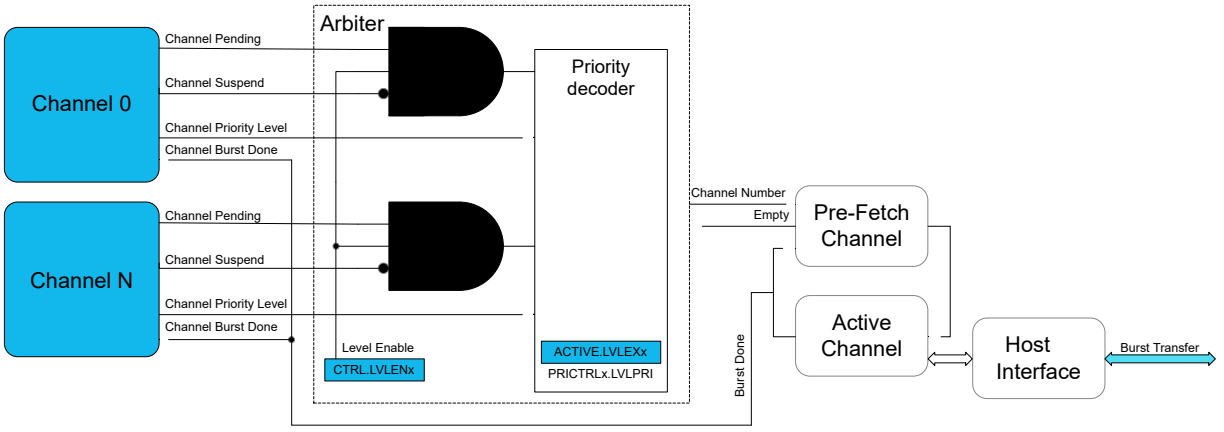
If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x 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 26-4. 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).

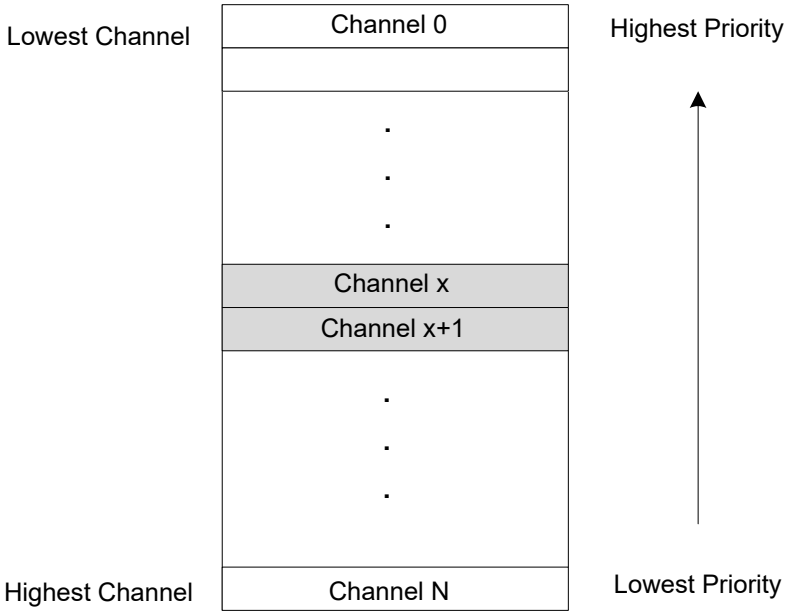
Each DMA channel supports up to 4-level priority scheme.

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.

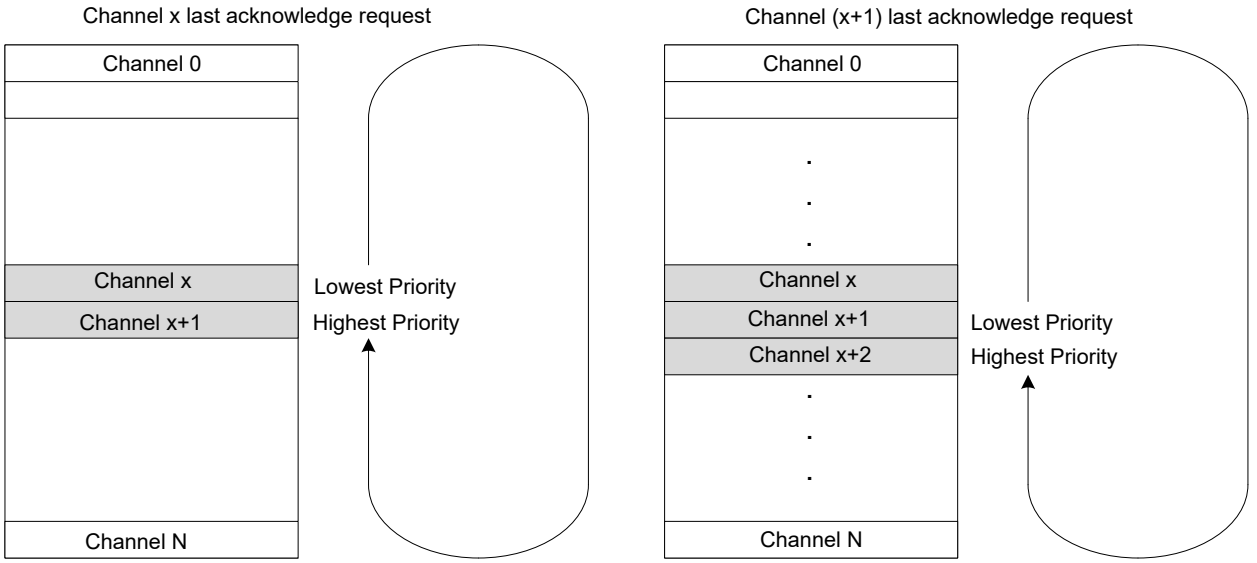
Within each priority level, the DMAC's arbiter can be configured to prioritize statically or dynamically. For the arbiter to perform static arbitration within a priority level, the Level X Round-Robin Scheduling Enable bit in the Priority Control x register (PRICTRL0.RRLVLENx) has to be written to '0'. 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 26-5. 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 26-6. Dynamic (Round-Robin) Priority Scheduling



26.6.2.5 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

26.3. Block Diagram
26.6.2.7. Addressing

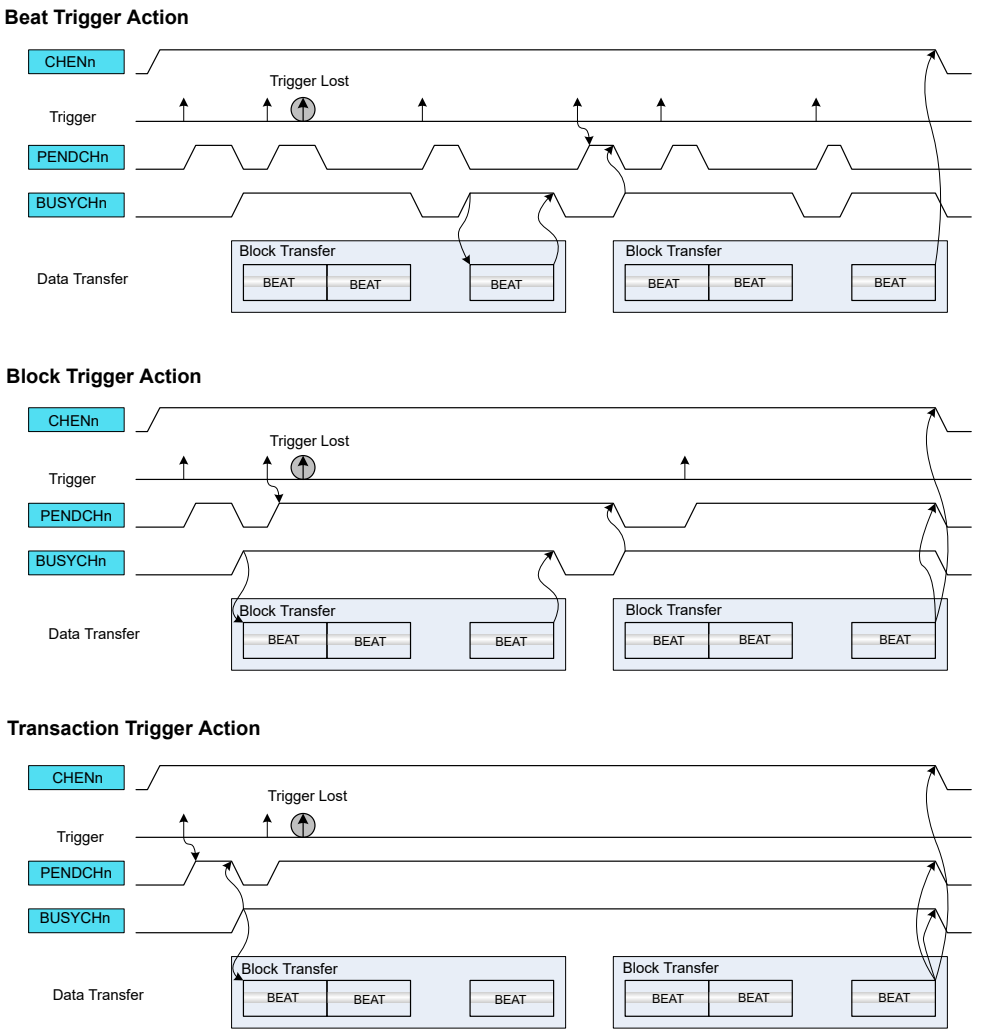
26.6.2.6 Transfer Triggers and Actions

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 26-7. Trigger Action and Transfers



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.

26.6.2.7 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, and 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:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

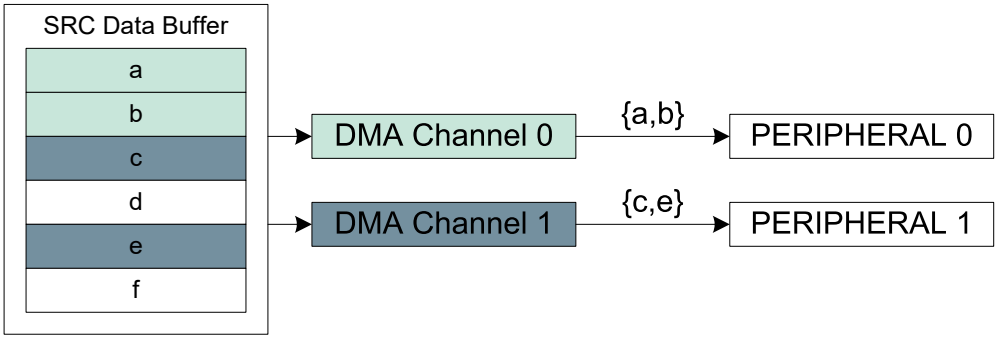
If BTCTRL.STEPSEL=0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ 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=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 26-8. 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.

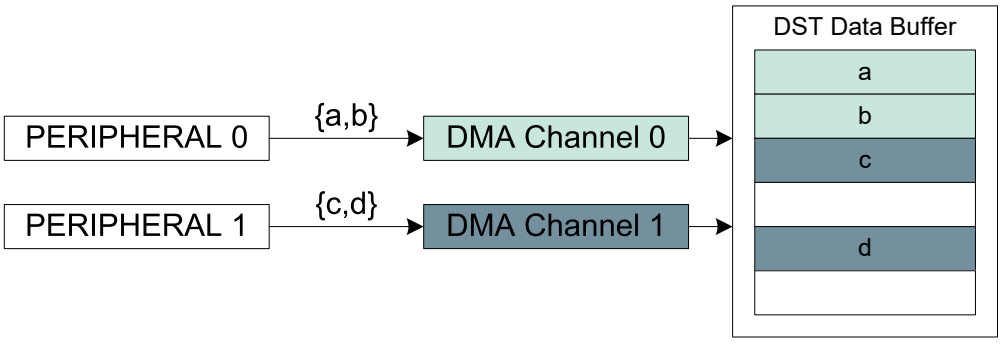
When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	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 26-9. Destination Address Increment



26.6.2.8 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.

26.6.2.9 Error Handling

If a bus error is received from an AHB client 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.

26.6.3 Additional Features

26.6.3.1 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

26.6.2.5. Data Transmission
26.6.2.3. Transfer Descriptors

26.6.3.1.1 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.

26.6.3.1.2 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.

26.6.3.1.3 Adding a Descriptor Between Existing Descriptors

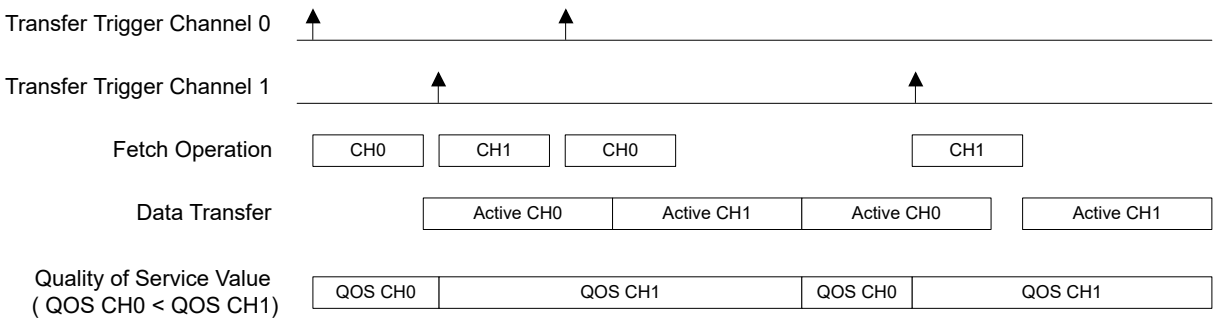
To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

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.
   c. Set the DESCADDR value of descriptor C to point to descriptor B.
   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.

26.6.3.2 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 26-10. 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.

26.6.3.3 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).

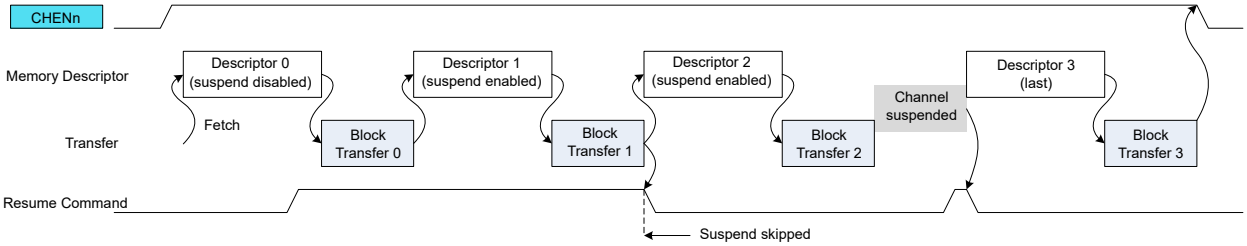
Related Links

26.6.2.3. Transfer Descriptors

26.6.3.4 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 26-11. Channel Suspend/Resume Operation



26.6.3.5 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 26-1. 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	

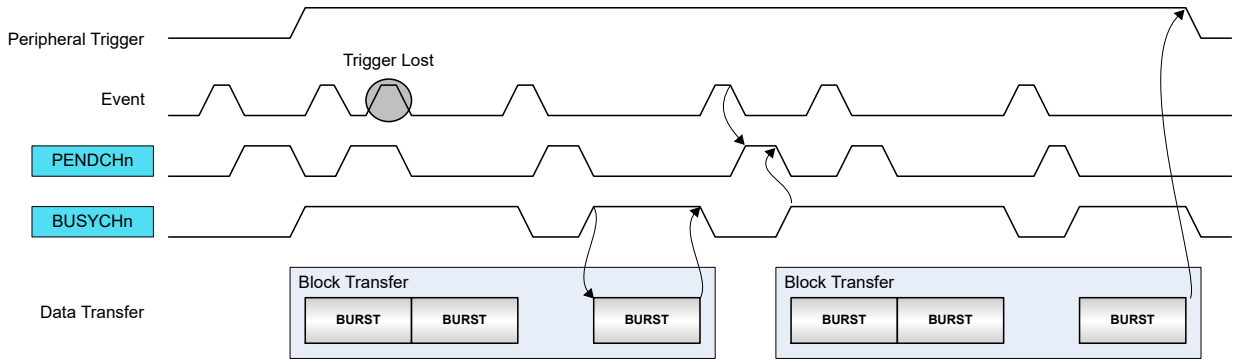
Normal Transfer

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 26-12. Burst Event Trigger Action



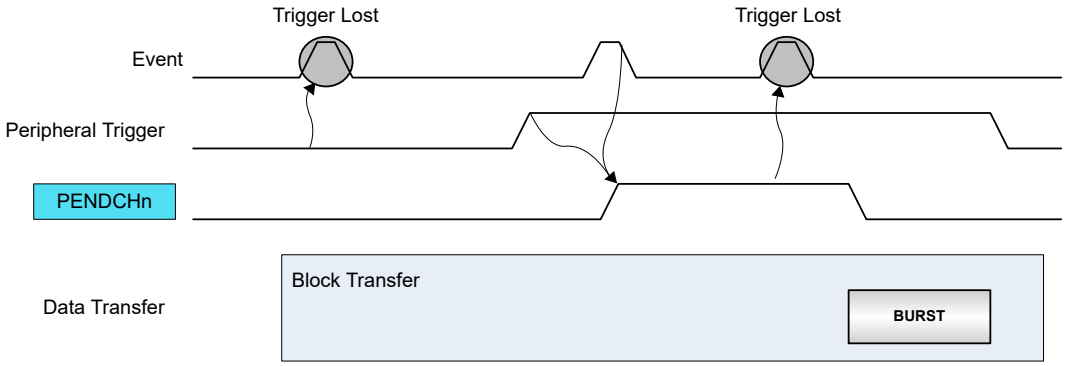
Conditional Transfer on Strobe

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 26-13. Periodic Event with Burst Peripheral Triggers



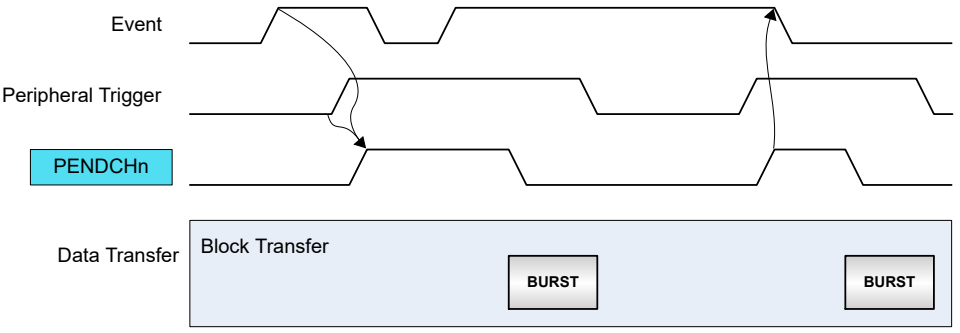
Conditional Transfer

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 26-14. Conditional Event with Burst Peripheral Triggers



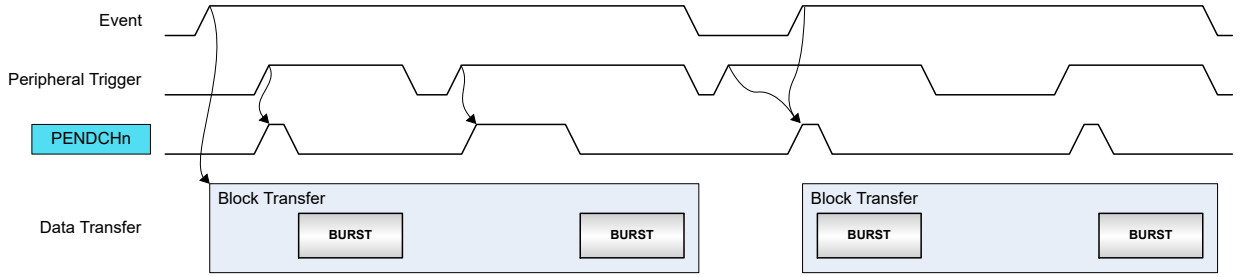
Conditional Block Transfer

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 26-15. 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

26.6.3.3. Channel Suspend
30. Event System (EVSYS)

26.6.3.6 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.

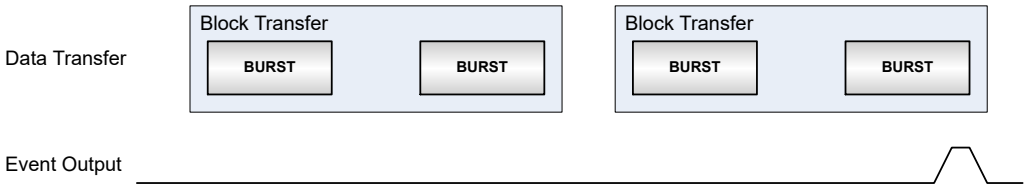
The transfer events (CHEVCTRL.EVOMODE = DEFAULT) are strobe events and their duration is one CLK_DMAC_AHB clock period. The transfer event type selection is available in each Descriptor Block Control location (BTCTRL.EVOSEL). Block or burst event output generation is supported.

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.

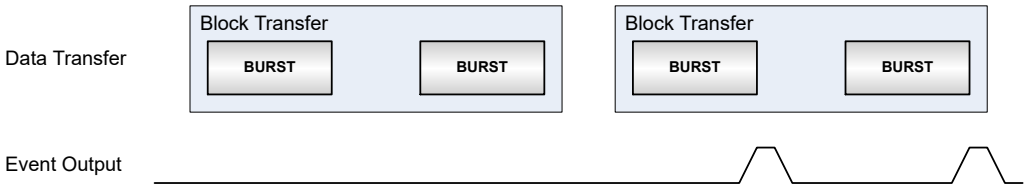
Figure 26-16. Block Event Output Generation



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.

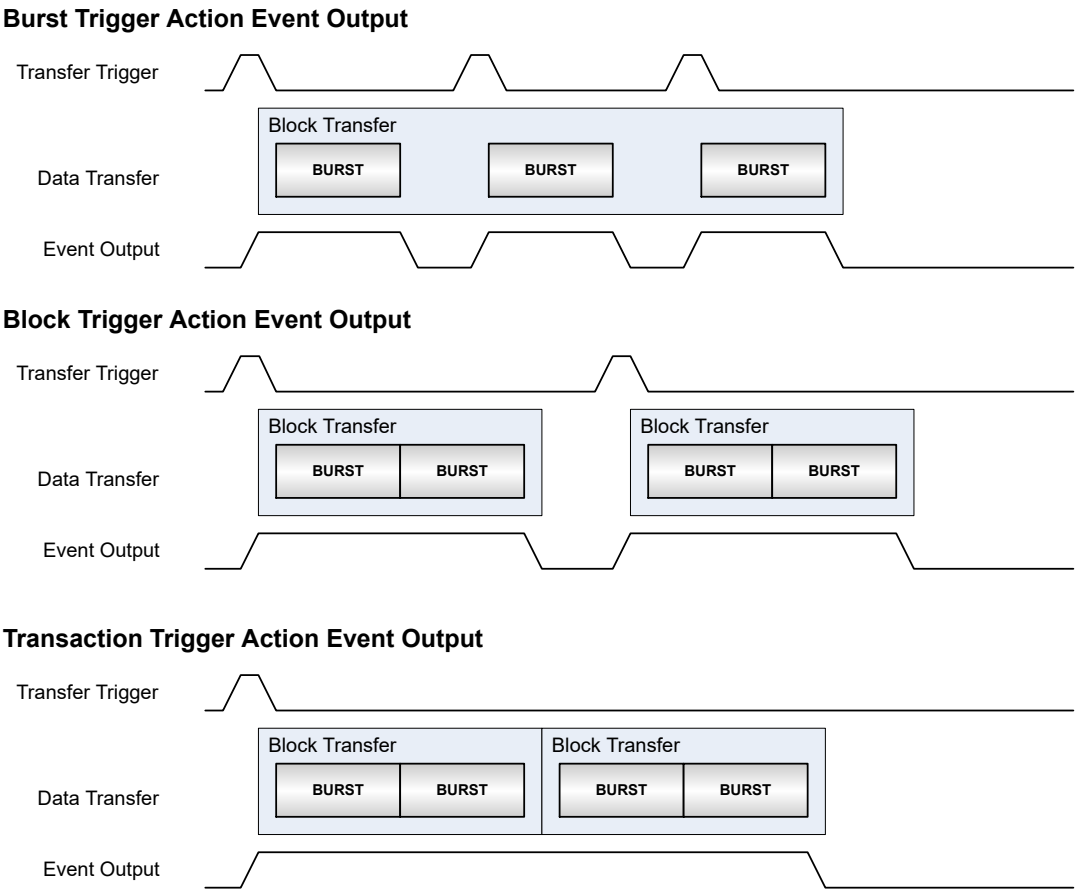
Figure 26-17. Burst Event Output Generation



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 26-18. Trigger Action Event Output Generation



26.6.3.7 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.

26.6.3.8 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 using the DSU's CRC engine. 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.

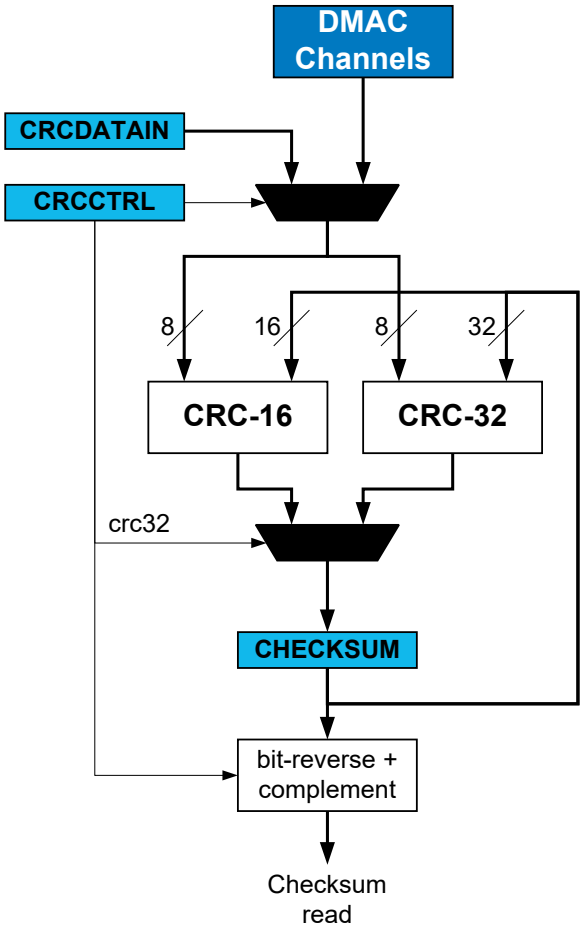
The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer 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 CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in *CRC Generator Block Diagram*.

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 data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of 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 26-19. 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 CRCBUSY flag is not set.

26.6.3.9 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.

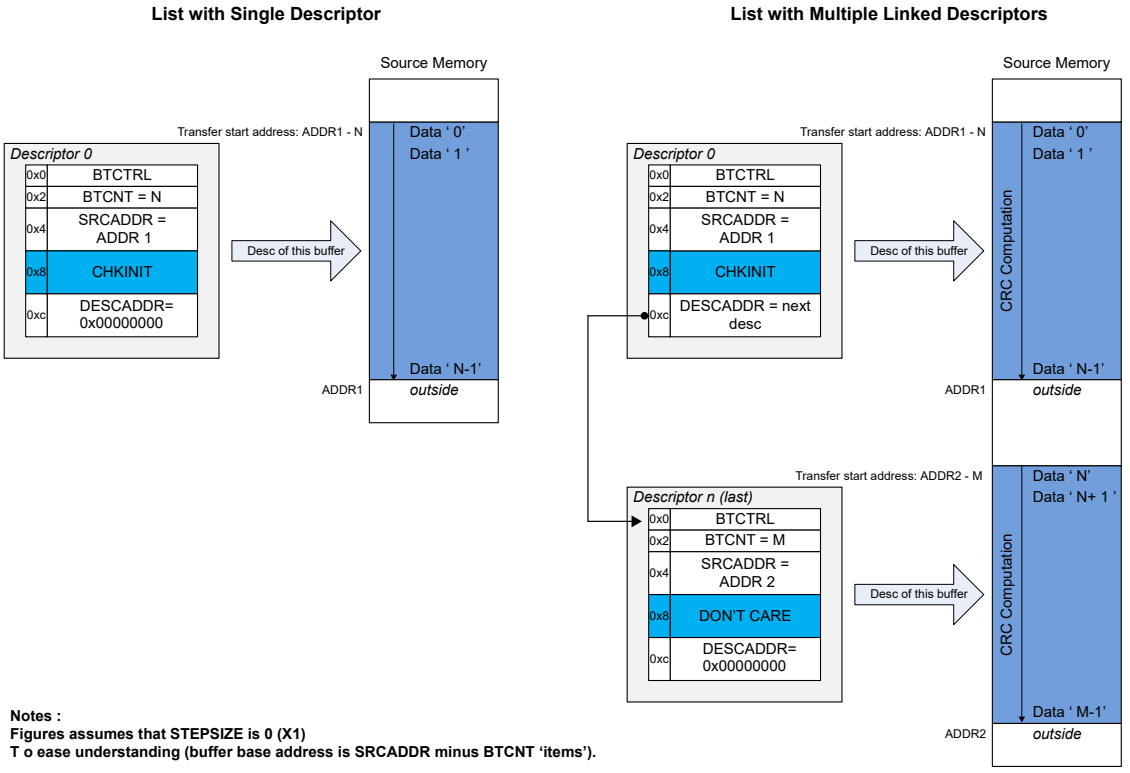
If linked descriptor is in the list (DESCADDR !=0), the DMA will fetch the next descriptor and CRC calculation continues as described above. When the last list descriptor is executed, the channel is automatically disabled.

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 descriptior 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 (CHCTRLAn.ENABLE)

The figure below shows the CRC computation slots and descriptor configuration when single or linked-descriptors transfers are enabled.

Figure 26-20. CRC Computation with Single Linked Transfers



26.6.3.10 Memory CRC Monitor

When enabled, it is possible to continuously check a a memory block data integrity by calculating and checking the CRC checksum. The following table shows the expected CRC checksum value that must be located in the last memory block location.

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 Expected CRC[7:0] Expected CRC[15:8]	
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, and the calculated CRC value from the CRC Checksum register is compared to zero or the 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.TCMPL) is set, optional interrupt is generated and the DMA performs the following actions depending on the descriptor list settings:

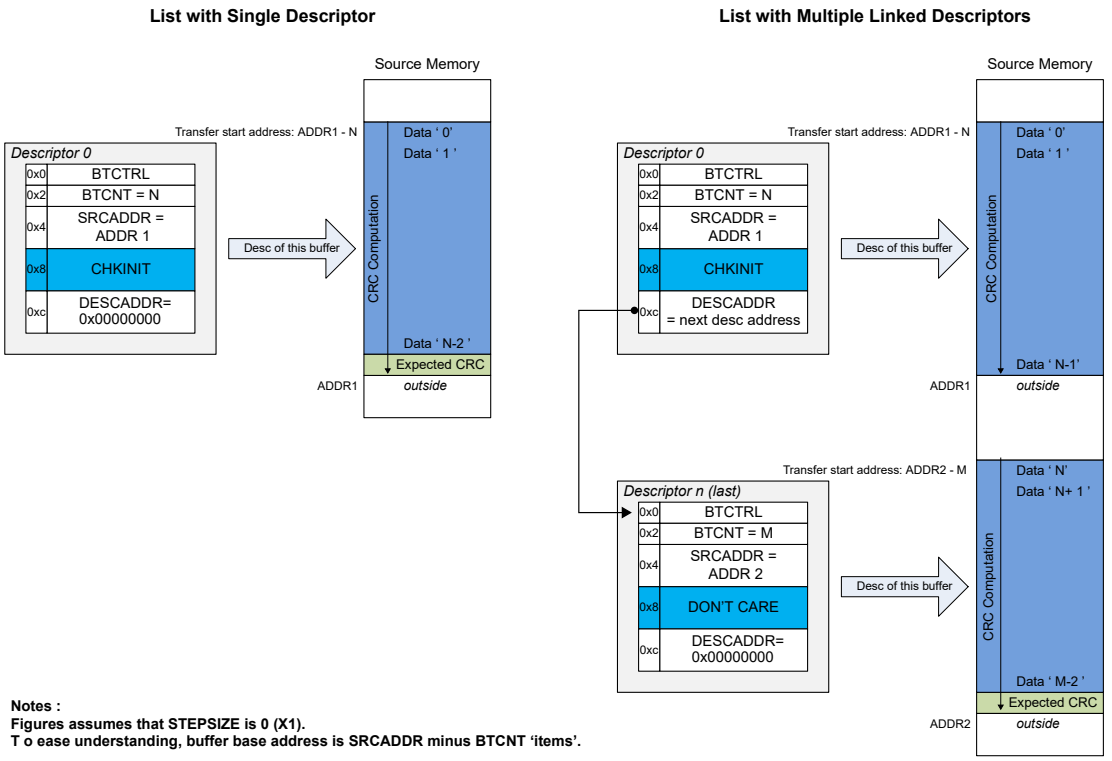
- If the list has only one descriptor, the DMA re-fetches the descriptor.
- If the current descriptor is the last descriptor from the list, the DMA fetches the first descriptor from the list.

When the fetch is completed, the DMA restarts the operations described above when new triggers are detected.

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 26-21. CRC Computation and Check with Single or Linked Transfers



Notes :
Figures assumes that STEPSIZE is 0 (X1).
T o ease understanding, buffer base address is SRCADDR minus BTCNT 'items'.

26.6.4 DMA Operation

Not applicable.

26.6.5 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 was fetched. See *Error Handling* from Related Links.
- Channel Suspend (SUSP) – Indicates that the corresponding channel was 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 the *CHINTFLAG* register from Related Links 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

9.2. Nested Vector Interrupt Controller (NVIC)
26.6.3.3. Channel Suspend
26.6.2.9. Error Handling
26.6.2.5. Data Transmission
26.8.22. CHINTFLAG

26.6.6 Events

The DMAC can generate the following output events:

- Channel (CH) – Generated when a block transfer for a given channel completes or when a beat transfer within a block transfer for a given channel completes. 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 is 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

26.2. Features
26.6.3.5. Event Input Actions
26.6.3.6. Event Output Selection

26.6.7 Sleep Mode Operation

DMAC behaves as in Dream mode. See *Dream Mode* from Related Links.

Related Links

18.3.1.4. Dream Mode

26.6.8 Synchronization

Not applicable.

26.7 Register Summary

Se the the *DMAC* 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	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]				LVLPRI0[4:0]		
		15:8	RRLVLEN1	QOS01[1:0]				LVLPRI1[4:0]		
		23:16	RRLVLEN2	QOS02[1:0]				LVLPRI2[4:0]		
		31:24	RRLVLEN3	QOS03[1:0]				LVLPRI3[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	CHINTn[7:0]							
		15:8	CHINTn[15:8]							
		23:16								
		31:24								
0x28	BUSYCH	7:0	BUSYCHn[7:0]							
		15:8	BUSYCHn[15:8]							
		23:16								
		31:24								
0x2C	PENDCH	7:0	PENDCHn[7:0]							
		15:8	PENDCHn[15:8]							
		23:16								
		31:24								
0x30	ACTIVE	7:0					LVLEXx3	LVLEXx2	LVLEXx1	LVLEXx0
		15:8	ABUSY			ID[4:0]				
		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	CHPRILVL0	7:0							PRILVL[1:0]	
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	CHPRILVL1	7:0							PRILVL[1:0]	
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	CHPRILVL2	7:0							PRILVL[1:0]	
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	CHPRILVL3	7:0							PRILVL[1:0]	
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							PRILVL[1:0]	
0x86	CHEVCTRL4	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x87 ... 0x8B	Reserved									
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							PRILVL[1:0]	
0x96	CHEVCTRL5	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x97 ... 0x9B	Reserved									
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							PRILVL[1:0]	
0xA6	CHEVCTRL6	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xA7 ... 0xAB	Reserved									
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							PRILVL[1:0]	
0xB6	CHEVCTRL7	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xB7 ... 0xBB	Reserved									
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							PRILVL[1:0]	
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							PRILVL[1:0]	
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							PRILVL[1:0]	
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							PRILVL[1:0]	
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							PRILVL[1:0]	
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							PRILVL[1:0]	
0x0116	CHEVCTRL13	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0117 ... 0x011B	Reserved									
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							PRILVL[1:0]	
0x0126	CHEVCTRL14	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0127 ... 0x012B	Reserved									
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							PRILVL[1:0]	
0x0136	CHEVCTRL15	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0137 ... 0x013B	Reserved									
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

Related Links

8. Product Memory Mapping Overview

26.8 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 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

26.5.8. Register Access Protection

26.8.1 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
26.6.2.4. Arbitration

26.8.2 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	NOACT	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

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	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3	—	Reserved

26.8.3 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.

26.8.4 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.

26.8.5 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

26.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write Protection

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 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.

26.8.7 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.
This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.
Writing a '0' to this bit clears the bit.
Writing a '1' to this bit generates a DMA software trigger on channel n, if CHSTATUS.PEND = 0 for channel n. CHSTATUS.PEND will be set and SWTRIGn remains cleared.

26.8.8 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
26.6.2.4. Arbitration

26.8.9 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 reads '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy
This bit reads '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error
This bit reads '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 12 – CRCERR CRC Error
This bit reads '1' when the channel selected by the 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 clears it. It also clears 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 reads '1' when the channel selected by the Channel ID field (ID) has a pending Suspend interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears 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 reads '1' when the channel selected by Channel ID field (ID) has a pending Transfer Complete interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears 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 reads '1' when the channel selected by the Channel ID field (ID) has a pending Transfer Error interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears 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 a 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 channel interrupts are available, these bits always return a zero value when read. When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

26.8.10 Interrupt Status

Name: INTSTATUS
Offset: 0x24
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
	CHINTn[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
	CHINTn[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CHINTn[15:0] Channel n Pending Interrupt [n=15..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.
This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

26.8.11 Busy Channels

Name: BUSYCH
Offset: 0x28
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
	BUSYCHn[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
	BUSYCHn[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BUSYCHn[15:0] Busy Channel n [n=15..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.
This bit is set when DMA channel n starts a DMA transfer.

26.8.12 Pending Channels

Name: PENDCH
Offset: 0x2C
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
	PENDCHn[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
	PENDCHn[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PENDCHn[15:0] Pending Channel n [n=0..15]
This bit is cleared when a trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on DMA channel n.

26.8.13 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			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEXx3	LVLEXx2	LVLEXx1	LVLEXx0
Access					R	R	R	R
Reset					0	0	0	0

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.

Bits 12:8 – ID[4:0] Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 0, 1, 2, 3 – LVLEXxx Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.
This bit is cleared when no request is pending or being executed.

26.8.14 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.

26.8.15 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.

26.8.16 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 where 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 addess 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 26-2. Triggers Map

Number	Name
0x00	DISABLE; Only software/event triggers
1	RTC_DMAC_ID_TIMESTAMP
2	DSU_DMAC_ID_DCC0
3	DSU_DMAC_ID_DCC1
4	SERCOM0_DMAC_ID_RX
5	SERCOM0_DMAC_ID_TX
6	SERCOM1_DMAC_ID_RX
7	SERCOM1_DMAC_ID_TX
8	SERCOM2_DMAC_ID_RX
9	SERCOM2_DMAC_ID_TX
10	TCC0_DMAC_ID_OVF
11	TCC0_DMAC_ID_MC_0
12	TCC0_DMAC_ID_MC_1
13	TCC0_DMAC_ID_MC_2
14	TCC0_DMAC_ID_MC_3
15	TCC0_DMAC_ID_MC_4
16	TCC0_DMAC_ID_MC_5
17	TCC1_DMAC_ID_OVF
18	TCC1_DMAC_ID_MC_0
19	TCC1_DMAC_ID_MC_1
20	TCC1_DMAC_ID_MC_2
21	TCC1_DMAC_ID_MC_3
22	TCC1_DMAC_ID_MC_4
23	TCC1_DMAC_ID_MC_5
24	TCC2_DMAC_ID_OVF
25	TCC2_DMAC_ID_MC_0
26	TCC2_DMAC_ID_MC_1
27	TC0_DMAC_ID_OVF
28	TC0_DMAC_ID_MC_0
29	TC0_DMAC_ID_MC_1
30	TC1_DMAC_ID_OVF
31	TC1_DMAC_ID_MC_0
32	TC1_DMAC_ID_MC_1
33	TC2_DMAC_ID_OVF
34	TC2_DMAC_ID_MC_0
35	TC2_DMAC_ID_MC_1

.....continued	
Number	Name
36	TC3_DMAC_ID_OVF
37	TC3_DMAC_ID_MC_0
38	TC3_DMAC_ID_MC_1
39	TC4_DMAC_ID_OVF
40	TC4_DMAC_ID_MC_0
41	TC4_DMAC_ID_MC_1
42	TC5_DMAC_ID_OVF
43	TC5_DMAC_ID_MC_0
44	TC5_DMAC_ID_MC_1
45	TC6_DMAC_ID_OVF
46	TC6_DMAC_ID_MC_0
47	TC6_DMAC_ID_MC_1
48	TC7_DMAC_ID_OVF
49	TC7_DMAC_ID_MC_0
50	TC7_DMAC_ID_MC_1
51	QSPI_DMAC_ID_RX
52	QSPI_DMAC_ID_TX

Bit 6 – RUNSTDBY Channel run in standby

This bit is used to keep the DMAC channel running in Standby Sleep 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

When writing a '0' to this bit during an ongoing transfer, the bit must not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer is empty when the ongoing burst transfer is completed.
Writing a '1' to this bit enables 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 is 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.

26.8.17 Channel Control B

Name: CHCTRLB
Offset: 0x44 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

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
26.6.3.4. Channel Resume and Next Suspend Skip
26.6.3.3. Channel Suspend

26.8.18 Channel Priority Level

Name: CHPRILVL
Offset: 0x45 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

Bits 1:0 – PRILVL[1:0] Channel Priority Level
These bits define the priority level used for the DMA channel. The available levels are shown below, where a high level has priority over a low level. These bits are not enable-protected.

Value	Name	Description
0x0	LVL0	Channel Priority Level 0 (Lowest Level)
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3 (Highest Level)

26.8.19 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 is 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
26.6.3.5. Event Input Actions
26.6.3.6. Event Output Selection

26.8.20 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.

26.8.21 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.

26.8.22 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
26.7. Register Summary
26.9. DMAC Register Summary (SRAM)

26.8.23 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
26.7. Register Summary
26.9. DMAC Register Summary (SRAM)

26.9 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]							

26.10 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 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

26.5.8. Register Access Protection

26.10.1 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 applies to the source or destination address, depending on the 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 the 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 disables the destination address incrementation. The address is kept fixed during the data transfer.
Writing a '1' to this bit enables 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 disables the source address incrementation. The address is kept fixed during the data transfer.
Writing a '1' to this bit enables 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 applies to both the read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	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 completes.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel is disabled if it is the last block transfer in the transaction
0x1	INT	Channel is 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 the 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

26.10.2 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.

26.10.3 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:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL= 0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ 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

26.10.4 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:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$$

If BTCTRL.STEPSEL = 0:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$$

- $DSTADDR_{START}$ 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

26.10.5 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.

27. External Interrupt Controller (EIC)

27.1 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 to wake-up the device from Sleep modes where all clocks were disabled. External pins can generate an event.

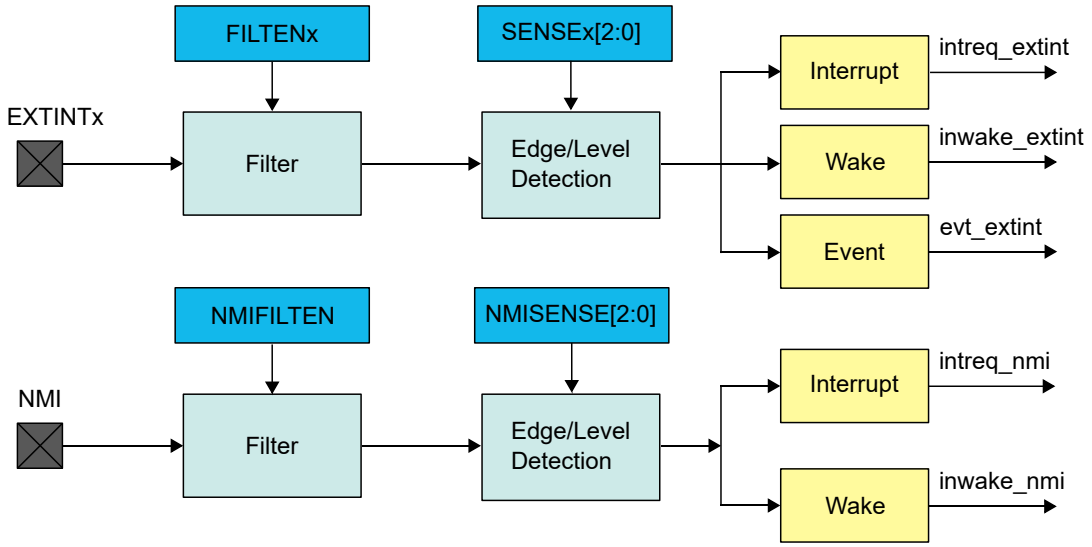
A separate Non-Maskable Interrupt (NMI) is also supported. It has properties similar to the other external interrupts but is connected to the NMI request of the NVIC.

27.2 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
- Synchronous or Asynchronous Edge Detection Mode
- Interrupt Pin Debouncing
- Interrupt on High or Low Levels
- Asynchronous Interrupts for Sleep Modes without Clock
- Filtering of External Pins
- Event Generation from EXTINTx

27.3 Block Diagram

Figure 27-1. EIC Block Diagram



27.4 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.

27.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

27.5.1 I/O Lines

To use EIC's I/O lines, configure the I/O pins using the I/O Peripheral Pin Select (PPS).

27.5.2 Power Management

The EIC continues to operate in any sleep modes (Standby Sleep, Idle) 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 the sleep modes.

27.5.3 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 32KHz_LPCLK, for the 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. For more details, see *Clock and Reset Unit (CRU)* from Related Links.

32KHz_LPCLK is provided by the various internal and external low power clock sources. For more details on configuration and selection of the clock, see *Clock and Reset Unit (CRU)* from Related Links.

Both GCLK_EIC and 32KHz_LPCLK are asynchronous to the user interface clock (PB1_CLK). Due to this asynchronicity, writes to certain registers require synchronization between the clock domains.

Related Links

17. Clock and Reset Unit (CRU)

27.5.4 DMA

Not applicable.

27.5.5 Interrupts

There are four external interrupts (EXTINT) and one Non-Maskable Interrupt (NMI).

All the EXTINT interrupt request lines are connected to a single interrupt in the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is connected to the non-maskable interrupt of the interrupt controller but does not require the interrupt to be configured.

27.5.6 Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

30. Event System (EVSYS)

27.5.7 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.

27.5.8 Register Access Protection

All registers with write access can be write protected optionally by the 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 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.

27.5.9 Analog Connections

Not applicable.

27.6 Functional Description

27.6.1 Principle of Operation

The EIC detects the 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 32KHz_LPCLK.

Related Links

27.6.3. External Pin Processing

27.6.2 Basic Operation

27.6.2.1 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 32KHz_LPCLK when one of the following configurations is selected:
   - The NMI uses edge detection or filtering
   - One EXTINT uses filtering
   - One EXTINT uses synchronous edge detection
   - One EXTINT uses debouncing

   GCLK_EIC is used when a frequency higher than 32 KHz is required for filtering.

   32KHz_LPCLK is recommended when power consumption is the priority. For 32KHz_LPCLK, 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. Optionally, enable the Asynchronous mode.
5. Optionally, enable the Debouncer mode.
6. Enable the EIC by writing a '1' to CTRLA.ENABLE.

The following bits are enable-protected, meaning that they 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 the *NMICTRL*, *CTRLA*, *CONFIG*, *ASYNCH*, *DEBOUNCEN*, *DPRESCALER*, *EVCTRL* registers in the *EIC Register Summary* from Related Links.

Related Links

27.7. Register Summary

27.6.2.2 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.

27.6.3 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 32KHz_LPCLK. 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 32KHz_LPCLK and outputs the value when two or more samples are equal.

Table 27-1. Majority Vote Filter

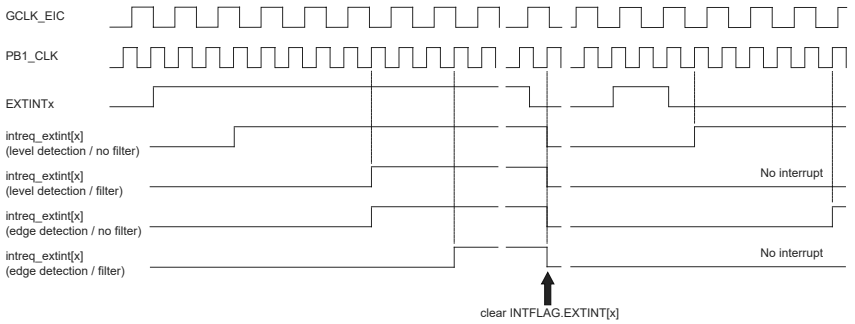
Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection does not require GCLK_EIC or 32KHz_LPCLK, but interrupt and events can still be generated.

If filtering or synchronous edge detection or debouncing is enabled, the EIC automatically requests GCLK_EIC or 32KHz_LPCLK 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 27-2. Interrupt Detection Latency by Modes (Rising Edge)



The detection latency depends on the detection mode.

Table 27-2. Detection Latency

Detection Mode	Latency (Worst Case)
Level without filter	Five PB1_CLK periods
Level with filter	Four GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods
Edge without filter	Four GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods
Edge with filter	Six GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods

27.6.4 Additional Features

27.6.4.1 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 32KHz_LPCLK.

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.

27.6.4.2 Asynchronous Edge Detection Mode (No Debouncing)

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 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.

Note: 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 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.

27.6.4.3 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 timeframe is set with the debouncer prescaler DPRESCALER.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) cannot be selected.

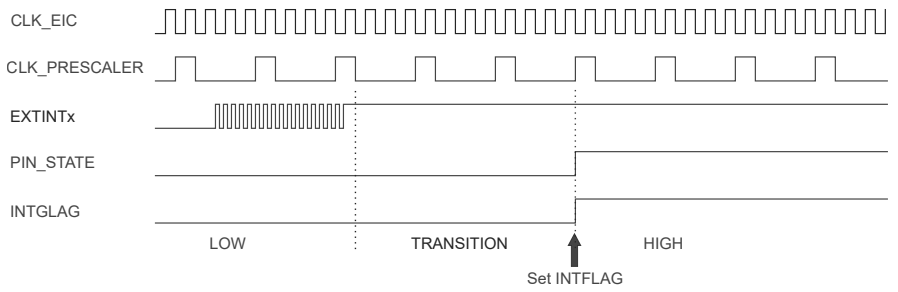
The debouncer manages an internal valid pin state that depends on the external interrupt (EXTINT) pin transitions, the Debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continuously on the EIC clock. The sampled value is evaluated on each low frequency clock tick to detect a transitional edge when the sampled value is different from the current valid pin state. The sampled value is evaluated on each EIC clock when DPRESCALER.TICKON=0 or on each low frequency clock tick when DPRESCALER.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 DPRESCALER.STATESn bit field. In the Synchronous mode, the threshold is 4 when DPRESCALER.STATESn=0 or 8 when DPRESCALER.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 either the Synchronous or Asynchronous Debouncing mode.

Synchronous Edge Detection mode – In this mode, the external interrupt (EXTINT) pin is sampled continuously on the EIC clock.

1. A pin edge transition is validated when the sampled value is consistently different from the current valid pin state for 4 (or 8 depending on bit DPRESCALER.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 increments the transition counter.
3. Any pin sample at the EIC clock rate (when DPRESCALER.TICKON=0) or the low frequency clock tick (when DPRESCALER.TICKON=1) with a value identical to the current valid pin state returns 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 27-3. EXTINT Pin Synchronous Debouncing (Rising Edge)

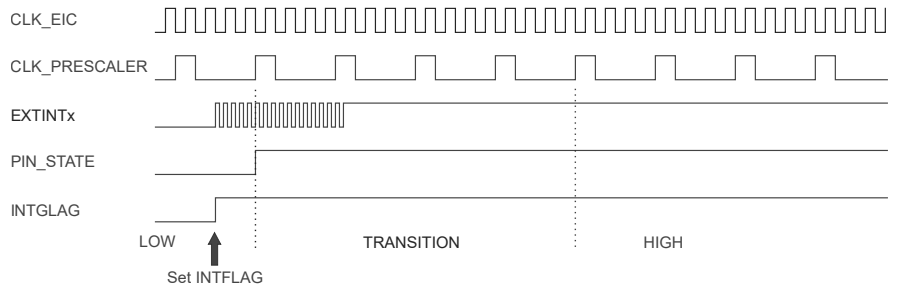


In the Synchronous Edge Detection mode, the EIC clock is required. The Synchronous Edge Detection mode can be used in Idle and Standby Sleep modes.

Asynchronous Edge Detection mode – 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 immediately sets 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 four consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the Idle state returns the transition counter to 0.
5. After four consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

Figure 27-4. 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.

27.6.5 DMA Operation

Not applicable.

27.6.6 Interrupts

The EIC has the following interrupt sources:

- External interrupt (EXTINTx) pins. See *Basic Operation* from Related Links.
- Non-maskable interrupt (NMI) pin. See *Additional Features* 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. For details on how to clear Interrupt flags, see the *INTFLAG* register from Related Links. The EIC has one interrupt request line for all external interrupts (EXTINTx) and one line for NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which Interrupt condition is present.

Notes:

- Interrupts must be globally enabled for interrupt requests to be generated.
- If an external interrupt (EXTINT) is common on two or more I/O pins, only one will be active (the first one programmed).TPUBSAMD-367

Related Links

27.6.2. Basic Operation
27.6.4. Additional Features
27.8.8. INTFLAG

27.6.7 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

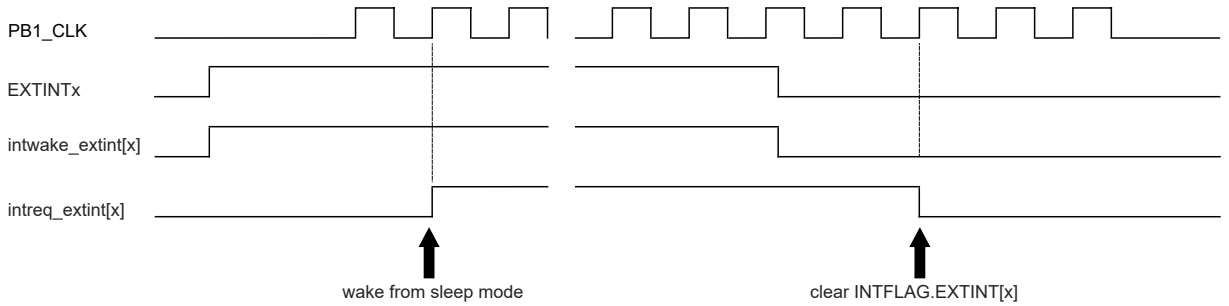
30. Event System (EVSYS)

27.6.8 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'.

Note: As soon as the EIC module is enabled and SENSEx is configured with different settings than no detection, the INTFLAGx bit records the activity on the EXTINTx pin, whether or not the Interrupt Enable bit is set.

Figure 27-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



27.6.9 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.

27.7 Register Summary

See the *EIC* module in the *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
0x03	Reserved									
0x04	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x08	EVCTRL	7:0					EXTINTEO[3:0]			
		15:8								
		23:16								
		31:24								
0x0C	INTENCLR	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x10	INTENSET	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x14	INTFLAG	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x18	ASYNCH	7:0					ASYNCH[3:0]			
		15:8								
		23:16								
		31:24								
0x1C	CONFIG	7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		23:16								
		31:24								
0x20 ... 0x2F	Reserved									
0x30	DEBOUNCEN	7:0					DEBOUNCEN[3:0]			
		15:8								
		23:16								
		31:24								
0x34	DPRESCALER	7:0					STATES0	PRESCALER0[2:0]		
		15:8								
		23:16								TICKON
		31:24								
0x38	PINSTATE	7:0					PINSTATE[3:0]			
		15:8								
		23:16								
		31:24								

Related Links

8. Product Memory Mapping Overview

27.8 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 require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

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

27.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW			RW	W
Reset				0			0	0

Bit 4 – CKSEL Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32.768 KHz is required for filtering) or by 32KHz_LPCLK (when power consumption is the priority).
This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by 32KHz_LPCLK.

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay between writing to 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 is set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	The EIC is disabled.
1	The EIC 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 EIC to their initial state, and the EIC is disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

27.8.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – NMIASYNCH Non-Maskable Interrupt Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.
In Synchronous Edge Detection mode, the NMI pin is sampled using the EIC clock as defined by the bit CTRLA.CKSEL. The Non-Maskable Interrupt flag (NMIFLAG) is set when the pin and the pin sampler have a different value. In this mode, the EIC clock is required. The Synchronous Edge Detection mode can be used in all Sleep modes except STANDBY.
In Asynchronous Edge Detection mode, the NMI pins directly drive the set of the Non-Maskable Interrupt flag (NMIFLAG). In this mode, the EIC clock is not requested. The Asynchronous Edge Detection Mode can be used in all sleep modes.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0] Non-Maskable Interrupt Sense Configuration
These bits define which edge or level the NMI triggers on.
Note: The NMI cannot be triggered based on level, but it is always based on edge.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	—	Reserved

27.8.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x2
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
								NMI
Access								RW
Reset								0

Bit 0 – NMI Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.
This flag is set when the NMI pin matches the NMI sense configuration and generates an interrupt request.
Writing a '0' to this bit has no effect.

27.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
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								
Reset								

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

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

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

27.8.5 Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, 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								
Reset								

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

Bits 3:0 – EXTINTEO[3:0] External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

27.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
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 register (INTENSET).

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
					EXTINT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EXTINT[3:0] External Interrupt Enable
The bit x of EXTINT disables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
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 Interrupt Enable Clear (INTENCLR) register.

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
					EXTINT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EXTINT[3:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
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
					EXTINT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EXTINT[3:0] External Interrupt
The flag bit x is cleared by writing a '1' to it.
This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the External Interrupt x flag.

27.8.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, 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								
Reset								

Bit	7	6	5	4	3	2	1	0
					ASYNCH[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 3:0 – ASYNCH[3:0] Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection Mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge detection is synchronously operated.
1	The EXTINT x edge detection is asynchronously operated.

27.8.10 External Interrupt Sense Configuration

Name: CONFIG
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, 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
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 3, 7, 11, 15 – FILTENx Filter Enable x [x=3..0]
Note: The filter must be disabled if the asynchronous detection is enabled.

Value	Description
0	Filter is disabled for EXTINT[x] input.
1	Filter is enabled for EXTINT[x] input.

Bits 0:2, 4:6, 8:10, 12:14 – SENSEx Input Sense Configuration x [x=3..0]
These bits define which edge or level the interrupt or event for EXTINT[x] is generated on.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	—	Reserved

27.8.11 Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, 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								
Reset								

Bit	7	6	5	4	3	2	1	0
					DEBOUNCEN[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 3:0 – DEBOUNCEN[3:0] Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge input is not debounced.
1	The EXTINT x edge input is debounced.

27.8.12 Debouncer Prescaler

Name: DPRESCALER
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TICKON
Access								RW
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					STATES0	PRESCALER0[2:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit 16 – TICKON Pin Sampler frequency selection
This bit selects the clock used for the sampling of bounce during transition detection.

Value	Description
0	The bounce sampler is using GCLK_EIC.
1	The bounce sampler is using the low frequency clock.

Bit 3 – STATES0 Debouncer number of states
This bit selects the number of samples by the debouncer low frequency clock needed to validate a transition from the current pin state to the next pin state in synchronous debouncing mode for pins EXTINT[3:0].

Value	Description
0	The number of low frequency samples is 3.
1	The number of low frequency samples is 7.

Bits 2:0 – PRESCALER0[2:0] Debouncer Prescaler
These bits select the debouncer low frequency clock for pins EXTINT[3:0].

Value	Name	Description
0x0	F/2	EIC clock divided by 2
0x1	F/4	EIC clock divided by 4
0x2	F/8	EIC clock divided by 8
0x3	F/16	EIC clock divided by 16
0x4	F/32	EIC clock divided by 32
0x5	F/64	EIC clock divided by 64
0x6	F/128	EIC clock divided by 128
0x7	F/256	EIC clock divided by 256

27.8.13 Pin State

Name: PINSTATE
Offset: 0x38
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								
Reset								

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

Bits 3:0 – PINSTATE[3:0] Pin State
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.

28. Configurable Custom Logic (CCL)

28.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral that can be connected to the device pins, to events or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each Look-up Table (LUT) consists of three inputs:

- Truth table
- Optional synchronizer/filter
- Optional edge detector

Each LUT can generate an output as a user-programmable logic expression with three inputs. Inputs can be individually masked.

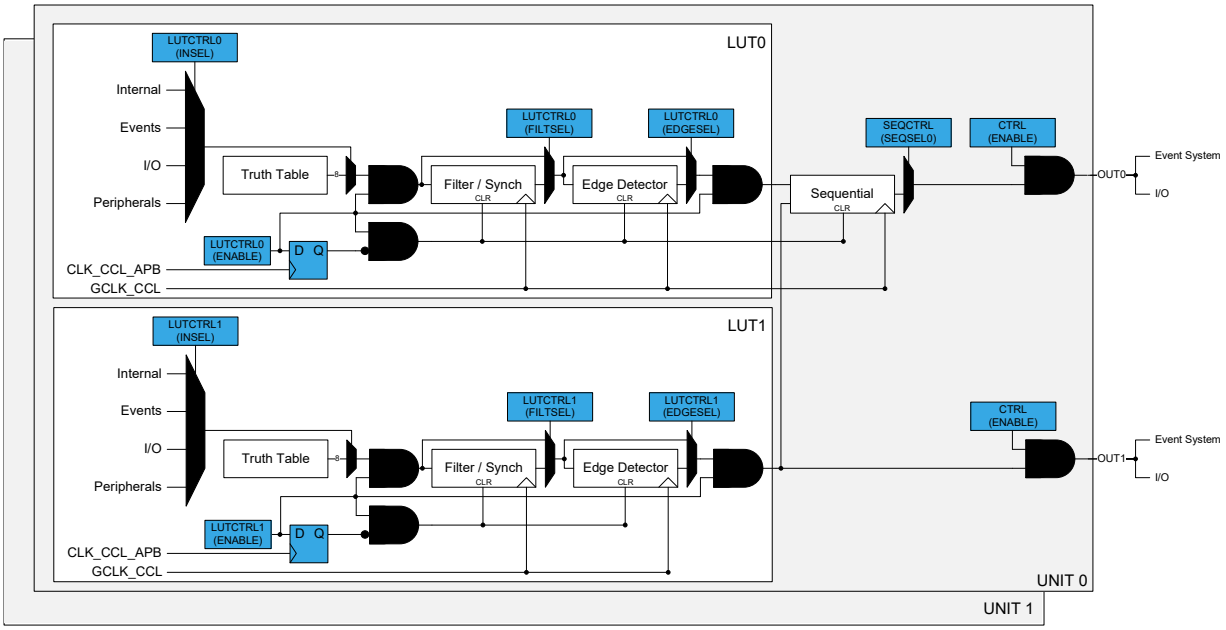
The output can be combinatorially generated from the inputs and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1) outputs, enabling complex waveform generation.

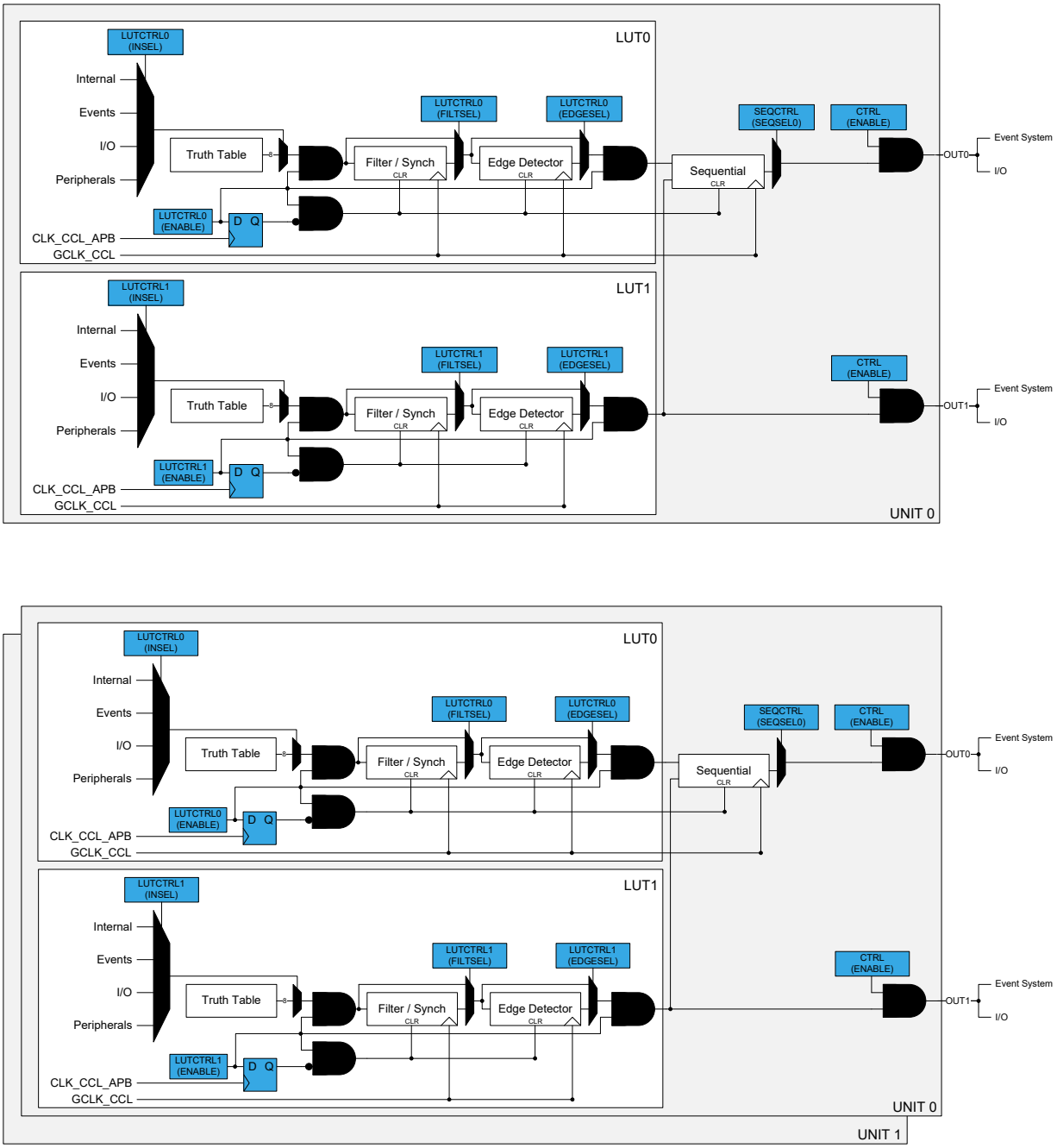
28.2 Features

- Glue Logic for General Purpose PCB Design
- Two Programmable Look-up Tables (LUTs)
- Combinatorial Logic Functions:
  - AND
  - NAND
  - OR
  - NOR
  - XOR
  - XNOR
  - NOT
- Sequential Logic Functions:
  - Gated D Flip-Flop
  - JK Flip-Flop
  - Gated D latch
  - RS latch
- Flexible LUT Inputs Selection:
  - I/Os
  - Events
  - Internal peripherals
  - Subsequent LUT output
- Output Can be Connected to the I/O Pins or the Event System
- Optional Synchronizer, Filter or Edge Detector Available on Each LUT Output

28.3 Block Diagram

Figure 28-1. Configurable Custom Logic





28.4 Signal Description

Pin Name	Type	Description
OUT[n:0]	Digital output	Output from look-up table
IN[3n+2:0]	Digital input	Input to look-up table

1. n (n=1) is the number of CCL groups.

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

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

28.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

28.5.1 I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look-up Table (LUT) using PPS configuration.

28.5.2 Power Management

This peripheral can continue to operate in any sleep modes (Standby Sleep, Idle) where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

28.5.3 Clocks

The CCL bus clock PB2_CLK (CLK_CCL_APB) can be enabled and disabled in the CRU.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector or a sequential sub-module is enabled.

This generic clock is asynchronous to the user interface clock.

See *Clock and Reset Unit (CRU)* from Related Links.

Related Links

17. Clock and Reset Unit (CRU)

28.5.4 DMA

Not applicable.

28.5.5 Interrupts

Not applicable.

28.5.6 Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the events have to be configured properly. See *Event System (EVSYS)* from Related Links.

Related Links

30. Event System (EVSYS)

28.5.7 Debug Operation

When the CPU is halted in Debug mode the CCL continues normal operation. However, the CCL cannot be halted when the CPU is halted in Debug mode. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

28.5.8 Register Access Protection

All registers with write access can be write protected optionally by the PAC. See *Peripheral Access Controller (PAC)* from Related Links.

Optional write protection by the 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.

Related Links
24. Peripheral Access Controller (PAC)

28.5.9 Analog Connections

Not applicable.

28.6 Functional Description

28.6.1 Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

28.6.2 Operation

28.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1' but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable protection is denoted by the Enable-Protected property in the register description.

28.6.2.2 Enabling, Disabling and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is Reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL are reset to their initial state, and the CCL is disabled.

28.6.2.3 Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 28-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 28-2. Truth Table Output Value Selection

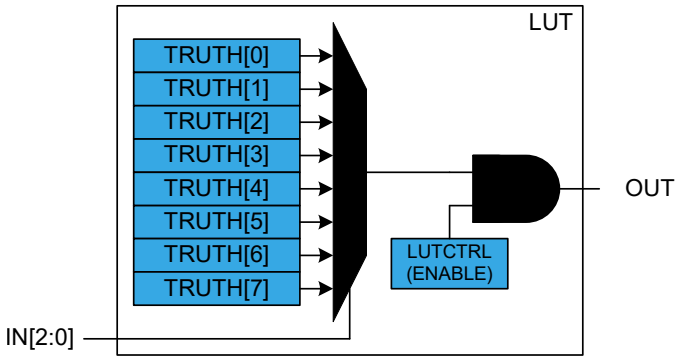


Table 28-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

28.6.2.4 Truth Table Inputs Selection

Input Overview

The inputs can be individually:

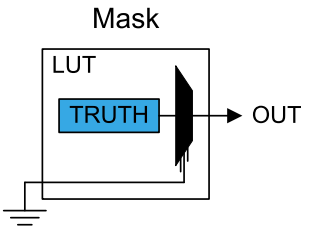
- Masked
- Driven by peripherals:
  - Analog Comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input 'y' of LUT x is configured by writing the Input 'y' Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELy = MASK), the corresponding TRUTH input (IN) is internally tied to zero, as illustrated in this figure.

Figure 28-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)

When selected (LUTCTRLx.INSELy = FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as an input source for the LUT. See Figure 28-4 for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

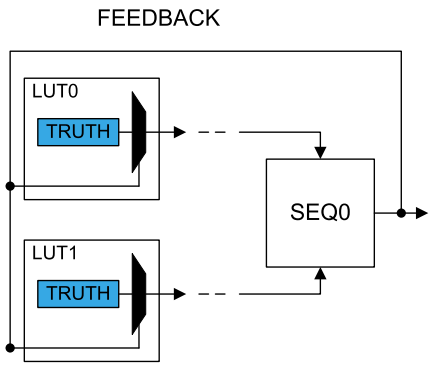
$IN[2N][i] = \mathrm{SEQ}[N]$

$IN[2N+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i* = [0,1,2] representing the LUT input index.

For additional information, see *Sequential Logic* from Related Links.

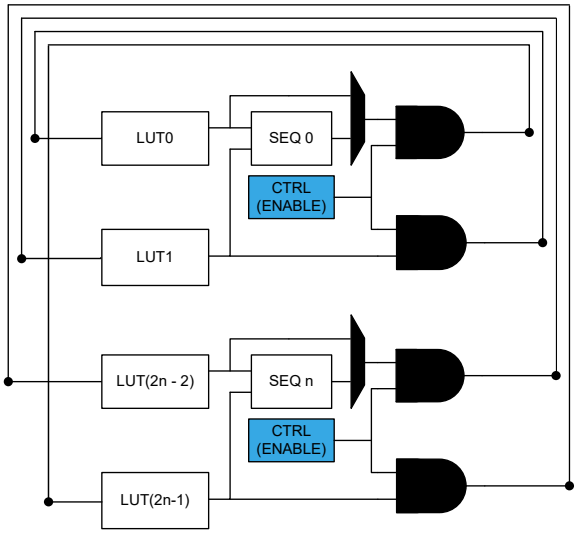
Figure 28-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (for example, LUT1 is the input for LUT0), as illustrated in the following figure.

Figure 28-5. Linked LUT Input Selection



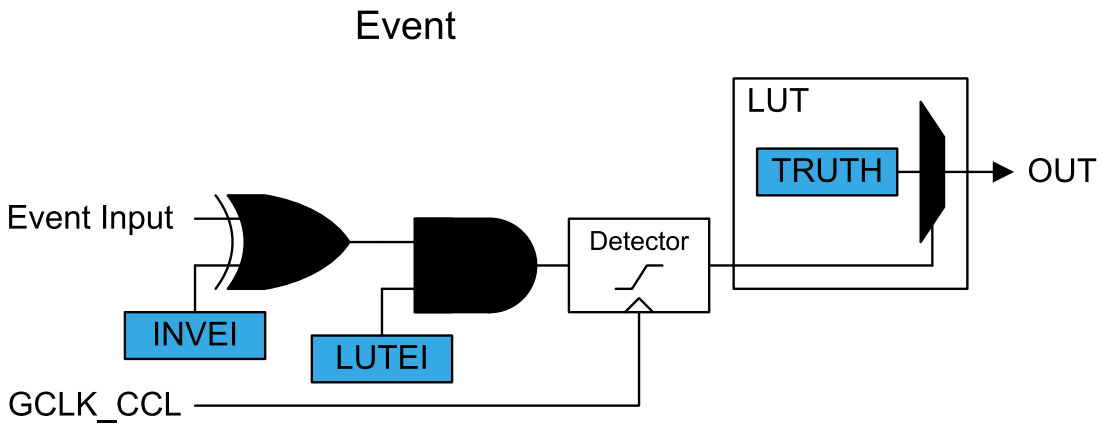
Internal Events Inputs Selection (EVENT)

Asynchronous events from the Event System can be used as an input selection, as illustrated in the following figure. For each LUT, one event input line is available and can be selected on each

LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default, CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle.

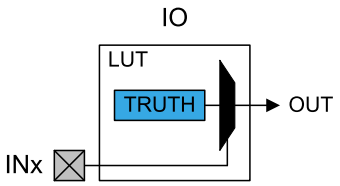
Figure 28-6. Event Input Selection



I/O Pin Inputs (IO)

When the I/O pin is selected as the LUT input (LUTCTRLx.INSELy = IO), the corresponding LUT input is connected to the pin, as illustrated in the following figure.

Figure 28-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as an input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

$$IN[N]\,[i] \;=\; AC[N\,\%\;ComparatorOutput\_Number]$$

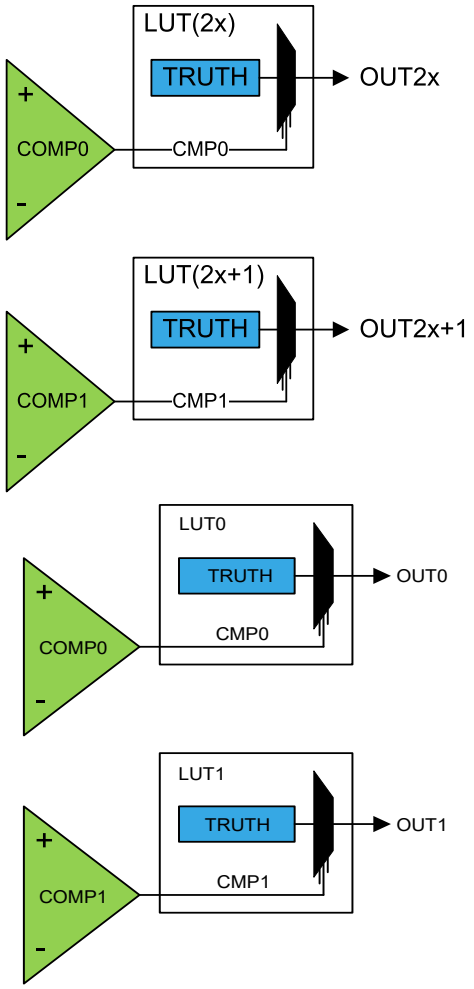
With *N* representing the LUT number and *i* = [0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1), as shown in the figure below.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1), as illustrated in the following figure.

Figure 28-8. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as an input source for the LUT (LUTCTRLx.INSELy = TC). Only consecutive instances of the TC, that is, TCx and the subsequent TC(x+1), are available as default and alternative TC selections (for example, TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1). For an example for LUT0, see Figure 28-9. More generally, the Timer/Counter selection for each LUT follows the formula:

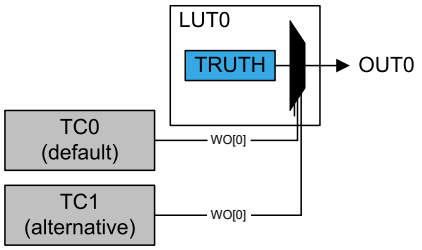
$$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC\_Instance\_Number}]$$

$$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC\_Instance\_Number}]$$

Where *N* represents the LUT number and *i* represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TC must be configured first.

Figure 28-9. TC Input Selection



Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as an input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (that is, IN0 is connected to WO0, IN1 to WO1 and IN2 to WO2), as illustrated in the following figure.

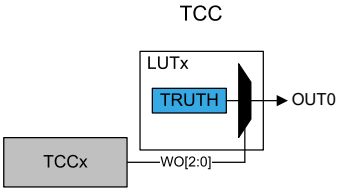
The TCC selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = TCC[N\ \%\ \mathrm{TCC\_Instance\_Number}].\mathrm{WO}[\mathrm{i}]$$

Where N represents the LUT number and i represents the LUT input index (i = [0,1,2]).

Before selecting the waveform outputs, the TCC must be configured first.

Figure 28-10. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

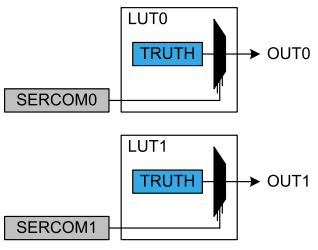
The serial engine transmitter output from the Serial Communication Interface (SERCOM TX, TXD for USART, MOSI for SPI) can be used as an input source for the LUT. The following figure illustrates an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = SERCOM[N\ \%\ \mathrm{SERCOM\_Instance\_Number}]$$

With *N* representing the LUT number and *i* = [0,1,2] representing the LUT input index.

Before selecting the SERCOM as an input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as the input pad to the CCL.

Figure 28-11. SERCOM Input Selection



Related Links

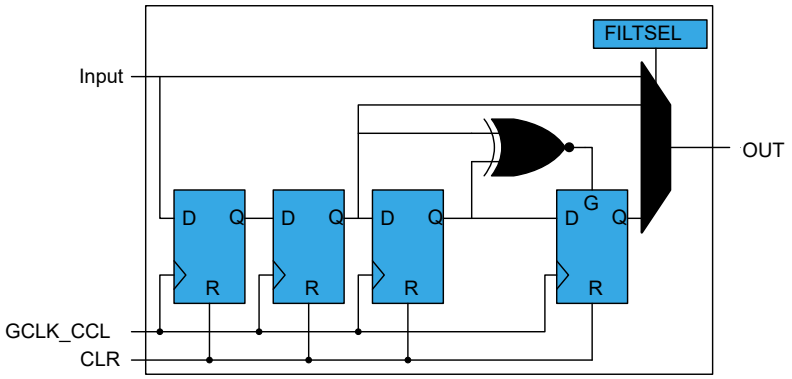
28.6.2.7. Sequential Logic

28.6.2.5 Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.
Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 28-12. Filter



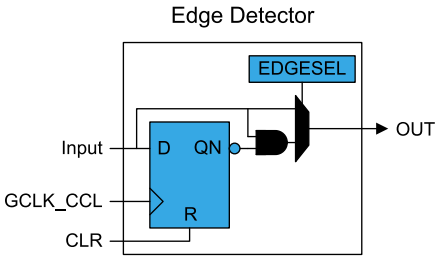
28.6.2.6 Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table must be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in the LUT Control register (LUTCTRLx.EDGESEL). To avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 28-13. Edge Detector



28.6.2.7 Sequential Logic

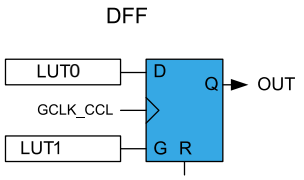
Each LUT pair can be connected to the internal sequential logic, which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

Note: While configuring the sequential logic, the even LUT must be disabled. When configured, the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output LUT0, and the G-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-14. D Flip Flop



When the even LUT is disabled LUTCTRL0.ENABLE=0, the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

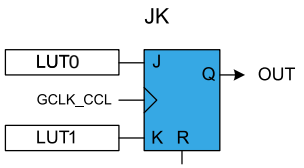
Table 28-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output LUT0, and the K-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-15. JK Flip Flop



When the even LUT is disabled LUTCTRL0.ENABLE=0, the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

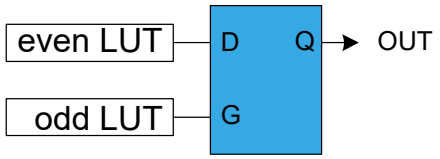
Table 28-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output LUT0, and the G-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-16. D-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0, the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

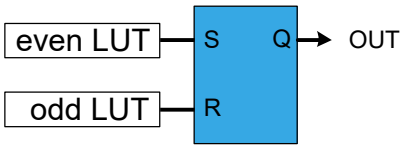
Table 28-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output LUT0, and the R-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-17. RS-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0, the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

Table 28-5. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

28.6.3 Events

The CCL can generate the following output events:

- LUTn where n=0-1 – Look-up Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The CCL can take the following actions on an input event:

- INSELx where x=0-2 – The event is used as input for the TRUTH table. See *Event System (EVSYS)* from Related Links.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

Related Links

30. Event System (EVSYS)

28.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' allows GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL is disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic is enabled, the LUT output is forced to zero in Standby mode. In all other cases, the TRUTH table decoder continues operation and the LUT output is refreshed accordingly.

28.7 Register Summary

See the *CCL* 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	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRLX	7:0					SEQSEL[3:0]			
0x05 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							

Related Links

8. Product Memory Mapping Overview

28.8 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 PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description. For details, 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

28.5.8. Register Access Protection

28.8.1 Control

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

Note: CTRL register (except the bits ENABLE & SWRST) is Enable Protected when CCL.CTRL.ENABLE = 1.

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in Standby Sleep mode. The setting is ignored for configurations where the generic clock is not required. For details, see *Sleep Mode Operation* from Related Links.

Important: This bit must be written before enabling the CCL.

Value	Description
0	Generic clock is not required in Standby Sleep mode.
1	Generic clock is required in Standby Sleep mode.

Bit 1 – ENABLE Enable

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 CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Related Links
28.6.4. Sleep Mode Operation

28.8.2 Sequential Control X

Name: SEQCTRLX
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-protected

Note: SEQCTRLX register is Enable-protected when CCL.LUTCTRL0.ENABLE = 1.

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

Bits 3:0 – SEQSEL[3:0] Sequential Selection
These bits select the sequential configuration:
Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 - 0xF	—	Reserved

28.8.3 LUT Control n

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Note: The LUTCTRLn register is Enable Protected when CCL.LUTCTRLn.ENABLE = 1.

Bit	31	30	29	28	27	26	25	24
	TRUTH[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	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSEL1[3:0]				INSEL0[3:0]			
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
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	R/W		R/W	R/W			R/W	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0] Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bits 8:11, 12:15, 16:19 – INSELx LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source

Value	Name	Description
0x5	AC	AC input source: CMP[0] (LUT0) / CMP[1] (LUT1)
0x6	TC	TC input source: TC0 WO[0] (LUT0) / TC1 WO[0] (LUT1)
0x7	ALTTC	Alternative TC input source: TC1 WO[0] (LUT0) / TC2 WO[0] (LUT1)
0x8	TCC	TCC input source: TCC0 (LUT0) / TCC1 (LUT1)
0x9	SERCOM	SERCOM input source: SERCOM0 PAD0 (LUT0) / SERCOM1 PAD0 (LUT1)
0xA-0xF	Reserved	Reserved

Bit 7 – EDGESEL Edge Selection

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0] Filter Selection

These bits select the LUT output filter options:
Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	—	Reserved

Bit 1 – ENABLE LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

29. Frequency Meter (FREQM)

29.1 Overview

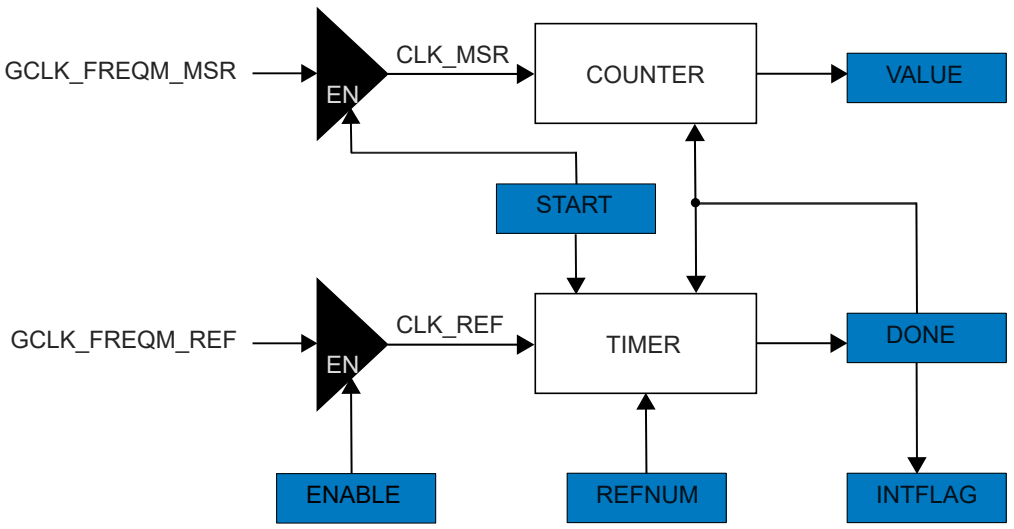
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

29.2 Features

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock with respect to a reference clock
- Reference clock can be selected from the available GCLK_FREQM_REF sources
- Measured clock can be selected from the available GCLK_FREQM_MSR sources

29.3 Block Diagram

Figure 29-1. FREQM Block Diagram



29.4 Signal Description

Not applicable.

29.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

29.5.1 I/O Lines

Other than the internal GCLK sources, the clock provided on the REFI line is the external clock input source for GCLK_FREQM_REF/GCLK_FREQM_MSR, which can be used for measurement or reference clock sources. This requires the I/O pins to be configured using PPS configuration. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

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

29.5.2 Power Management

The FREQM continues to operate in the Idle mode, where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from the Idle mode. See *Power Management Unit (PMU)* from Related Links for more details on the different sleep modes.

Related Links

18. Power Management Unit (PMU)

29.5.3 Clocks

The clock for the FREQM bus interface (PB1_CLK) is enabled and disabled by the CRU generator.

Two generic clocks are used by the FREQM:

- Reference Clock (GCLK_FREQM_REF – GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.
- Measurement Clock (GCLK_FREQM_MSR – GCLK_FREQM_MSR is required to clock a ripple counter for the frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

See *Clock and Reset Unit (CRU)* from Related Links.

Related Links

17. Clock and Reset Unit (CRU)

29.5.4 DMA

Not applicable.

29.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

29.5.6 Events

Not applicable

29.5.7 Debug Operation

When the CPU is halted in Debug mode, the FREQM continues its normal operation. The FREQM cannot be halted when the CPU is halted in Debug mode. If the FREQM is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

29.5.8 Register Access Protection

All registers with write access can be write protected optionally by the PAC, except the following registers:

- Control B register (CTRLB)
- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write protection by the PAC is denoted by the PAC Write-Protection property in each individual register description.

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

29.6 Functional Description

29.6.1 Principle of Operation

FREQM counts the number of periods of the measured clock (GCLK_FREQM_MSR) with respect to the reference clock (GCLK_FREQM_REF). The measurement is done for a period of REFNUM/f_{CLK_REF} and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM).

The frequency of the measured clock, f_{CLK_MSR}, is calculated by

$f_{CLK\_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK\_REF}$. The error can be maximum two measured clock cycles.

29.6.2 Basic Operation

29.6.2.1 Initialization

Before enabling FREQM, the device and peripheral must be configured:

- Each of the generic clocks (GCLK_FREQM_REF and GCLK_FREQM_MSR) must be configured and enabled.

> **Important:** The reference clock must be slower than the measurement clock.

- Write the number of Reference clock cycles the measurement needs to be done in to the Configuration A register (CFGA.REFNUM). This must be a non-zero number.

The following register is enable-protected, meaning that it can only be written when the FREQM is disabled (CTRLA.ENABLE=0):

- Configuration A register (CFGA)

Enable protection is denoted by the Enable-Protected property in the register description.

29.6.2.2 Enabling, Disabling and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE=0.

The FREQM is Reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). On software Reset, all registers in the FREQM reset to their initial state, and the FREQM is disabled.

Then ENABLE and SWRST bits are write-synchronized.

29.6.2.3 Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.
Note: The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{CLK\_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK\_REF}$$

Notes:

1. In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) must be checked.
2. Due to asynchronous operations, the VALUE Error measurement can be up to two samples.

If an overflow condition occurred, indicated by the overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM) or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then, another measurement can be started by writing a '1' to CTRLB.START.

Note: See *CFGA*, *CTRLB*, *STATUS*, *INTENSET*, *INTFLAG*, *VALUE* registers in the *Register Summary - FREQM* from Related Links.

Related Links

29.7. Register Summary

29.6.3 DMA Operation

Not applicable.

29.6.4 Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be 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. 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 FREQM is reset. See the *INTFLAG* register Related Links 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. The user must read the INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

29.8.6. INTFLAG

29.6.5 Events

Not applicable.

29.6.6 Sleep Mode Operation

The FREQM continues to operate in Idle mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from Idle mode.

For lowest chip power consumption in sleep modes, FREQM must be disabled before entering a Standby Sleep mode.

29.6.7 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 and registers are write-synchronized:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

Required write synchronization is denoted by the Write-Synchronized property in the register description.

29.7 Register Summary

See the *FREQM* 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	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	7:0	REFNUM[7:0]							
		15:8								
0x04 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DONE
0x09	INTENSET	7:0								DONE
0x0A	INTFLAG	7:0								DONE
0x0B	STATUS	7:0							OVF	BUSY
0x0C	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x10	VALUE	7:0	VALUE[7:0]							
		15:8	VALUE[15:8]							
		23:16	VALUE[23:16]							
		31:24								

Related Links

8. Product Memory Mapping Overview

29.8 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 require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

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

Some registers are optionally write-protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

29.8.1 CTRLA - Control A Register

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled or disabled. The value written to CTRLA.ENABLE reads back immediately, and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete. 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 resets all registers in the FREQM to their initial state, and the FREQM is disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST are cleared when the Reset is complete. This bit is not enable-protected.

Value	Description
0	There is no ongoing Reset operation.
1	The Reset operation is ongoing.

29.8.2 CTRLB - Control B Register

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

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

Bit 0 – START Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a measurement.

29.8.3 CFGA - Configuration A Register

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REFNUM[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 7:0 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in the number of GCLK_FREQM_REF cycles. This must be a non-zero value, in other words, 0x01 (one cycle) to 0xFF (255 cycles).

29.8.4 INTENCLR - Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

29.8.5 INTENSET - Interrupt Enable Set Register

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

29.8.6 INTFLAG - Interrupt Flag Status and Clear Register

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE Mesurement Done
This flag is cleared by writing a '1' to it.
This flag is set when the STATUS.BUSY bit has a one-to-zero transition.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DONE Interrupt flag.

29.8.7 STATUS - Status Register

Name: STATUS
Offset: 0x0B
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							OVF	BUSY
Access							R/W	R
Reset							0	0

Bit 1 – OVF Sticky Count Value Overflow
This bit is cleared by writing a '1' to it.
This bit is set when an overflow condition occurs to the value counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the OVF status.
Clearing the CTRLA.ENABLE register bit will clear the OVF bit.

Bit 0 – BUSY FREQM Status

Value	Description
0	No ongoing frequency measurement.
1	Frequency measurement is ongoing.

Note: If the measurement clock stalls (or is very slow) during a measurement slot, the STATUS.BUSY will never de-assert and the DONE interrupt will not be raised.

29.8.8 SYNCBUSY - Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x0C
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
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable
This bit is cleared when the synchronization of CTRLA.ENABLE is complete.
This bit is set when the synchronization of CTRLA.ENABLE is started.

Bit 0 – SWRST Synchronization Busy
This bit is cleared when the synchronization of CTRLA.SWRST is complete.
This bit is set when the synchronization of CTRLA.SWRST is started.

29.8.9 VALUE - Value Register

Name: VALUE
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
	VALUE[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
	VALUE[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
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – VALUE[23:0] Measurement Value
Result from measurement.

30. Event System (EVSYS)

30.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

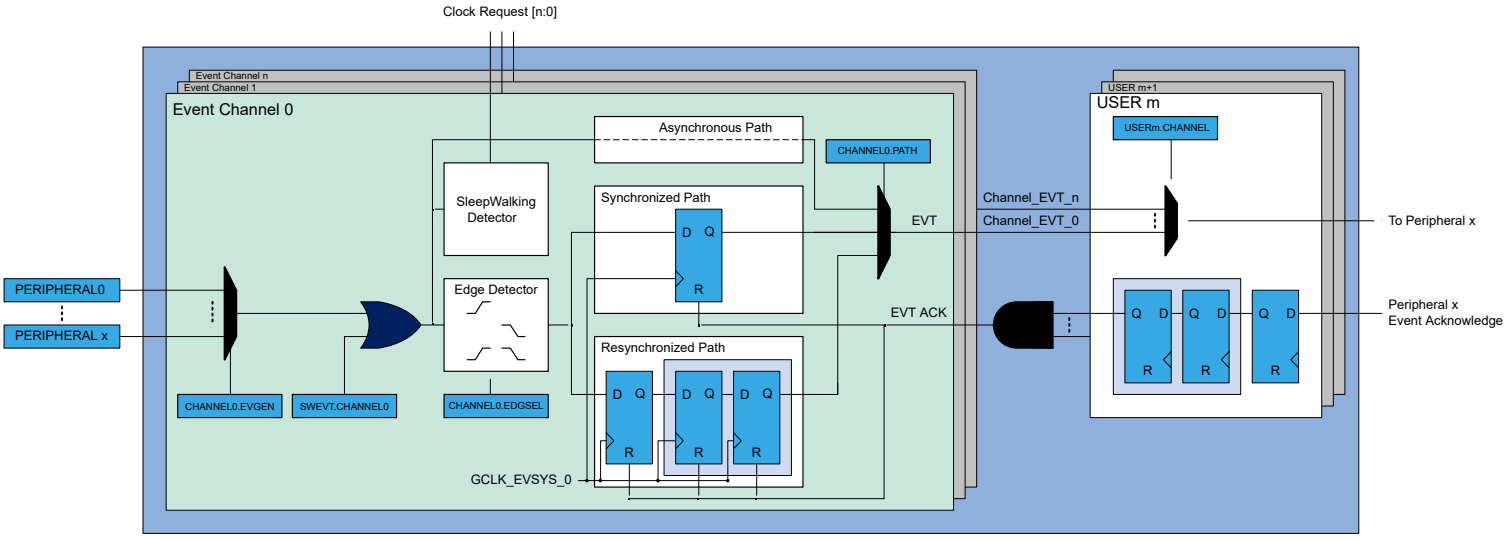
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

30.2 Features

- 32 Configurable Event Channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - Twelve channels provide a resynchronized or synchronous path
- 81 Event Generators
- 57 Event Users
- Configurable Edge Detector
- Peripherals Can be Event Generators, Event Users or Both
- Sleep-Walking and Interrupt for Operation in Sleep Modes
- Software Event Generation
- Each Event User Can Choose which Channel to Respond to
- Optional Static or Round-Robin Interrupt Priority Arbitration

30.3 Block Diagram

Figure 30-1. Event System Block Diagram



30.4 Signal Description

Not applicable.

30.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

30.5.1 I/O Lines

Not applicable.

30.5.2 Power Management

The EVSYS can be used to wake up the CPU from the Idle and Standby Sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. See *Power Management Unit (PMU)* from Related Links for details on the different sleep modes.

In sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CH_n) is restarted, if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Related Links

18. Power Management Unit (PMU)

30.5.3 Clocks

The EVSYS bus clock (PB2_CLK) can be enabled and disabled in the CRU. Each EVSYS channel, which can be configured as synchronous or resynchronized, has a dedicated generic clock (GCLK_EVSYS_CH_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock generator before using the EVSYS. See *Peripheral Clock Generation (GCLK)* from Related Links for more details on clock generation.

Important: Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

Related Links
17.3.7. Peripheral Clock Generation (GCLK)

30.5.4 DMA
Not applicable.

30.5.5 Interrupts
The interrupt request line is connected to the interrupt controller. Using the EVSYS interrupts requires the interrupt controller to be configured first (see *Nested Vector Interrupt Controller (NVIC)* from Related Links).

Related Links
9.2. Nested Vector Interrupt Controller (NVIC)

30.5.6 Events
Not applicable.

30.5.7 Debug Operation
When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

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

- Channel Pending Interrupt (INTPEND)
- Channel n Interrupt Flag Status and Clear (CHINTFLAGn)

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

When the CPU is halted in Debug mode, all write protection is automatically disabled. Write protection does not apply for accesses through an external debugger.

30.5.9 Analog Connections
Not applicable.

30.6 Functional Description

30.6.1 Principle of Operation
The Event System consists of channels which route the internal events from peripherals (generators) to other internal peripherals. Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

See *Channel Path* from Related Links.

Related Links
30.6.2.6. Channel Path

30.6.2 Basic Operation

30.6.2.1 Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event must be configured. Follow these steps to configure the event:

1. In the peripheral generating the event, enable the output of the event by writing a '1' to the respective Event Output Enable bit (EO) in the peripheral's Event Control register, for example, AC.EVCTRL.WINEO0 or RTC.EVCTRL.OVFEO.
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register; see *User Multiplexer Setup* from Related Links.
   b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register; see *Event System Channel* from Related Links.
3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register, for example, TC.EVCTRL.EVACT.
   Note: This step is not applicable for all the peripherals.
4. In the event it is a user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register, for example, AC.EVCTRL.IVEI0.
   Note: The ADC, CVD and ZB modules do not have the EVCTRL register; therefore, the EVCTRL register configuration is not applicable for these modules.

Related Links
30.6.2.3. User Multiplexer Setup
30.6.2.4. Event System Channel

30.6.2.2 Enabling, Disabling and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS reset to their initial state and all ongoing events are canceled.

See *CTRLA* from Related Links.

Related Links
30.8.1. CTRLA

30.6.2.3 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all available event users is found in the User (USERm) register description.

30.6.2.4 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in *Block Diagram* section.

30.6.2.5 Event Generators

Each event channel can receive the events from all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (in other words, CHANNELn.EVGEN=0).

30.6.2.6 Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CH_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel x Status register (CHSTATUSx) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Synchronous Path

The synchronous path must be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user.

When using the synchronous path, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

30.6.2.7 Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

30.6.2.8 Event Latency

The latency from the event generator to the event user depends on the channel's configuration:

- Asynchronous Path – The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path – The maximum routing latency of an external event is one GCLK_EVSYS_CH_n clock cycle.
- Resynchronized Path – The maximum routing latency of an external event is three GCLK_EVSYS_CH_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

30.6.2.9 The Overrun Channel n Interrupt

The Overrun Channel n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAGn.OVR) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAGn.OVR is always read as zero.

30.6.2.10 The Event Detected Channel n Interrupt

The Event Detected Channel n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAGn.EVD) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a synchronous or resynchronized path. In the case of an asynchronous path, the INTFLAGn.EVD is always zero.

30.6.2.11 Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUSn.BUSYCH bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUSn.RDYUSR bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

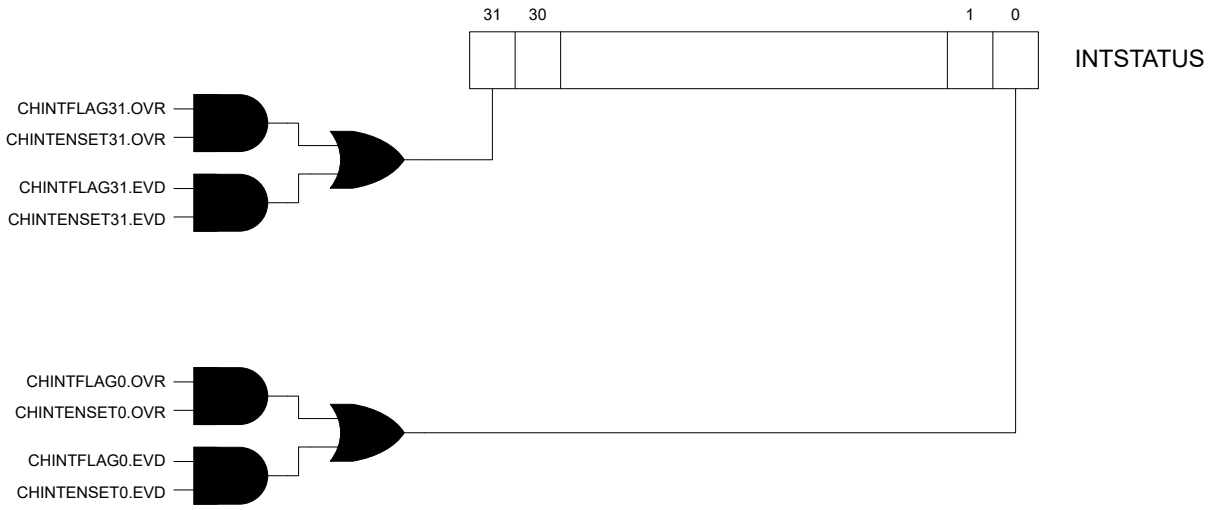
30.6.2.12 Software Event

A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (CHANNELm.SWEVT). Then, the software event can be serviced as any event generator; in other words, when the bit is set to '1', an event is generated on the respective channel.

30.6.2.13 Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as illustrated in the following figure.

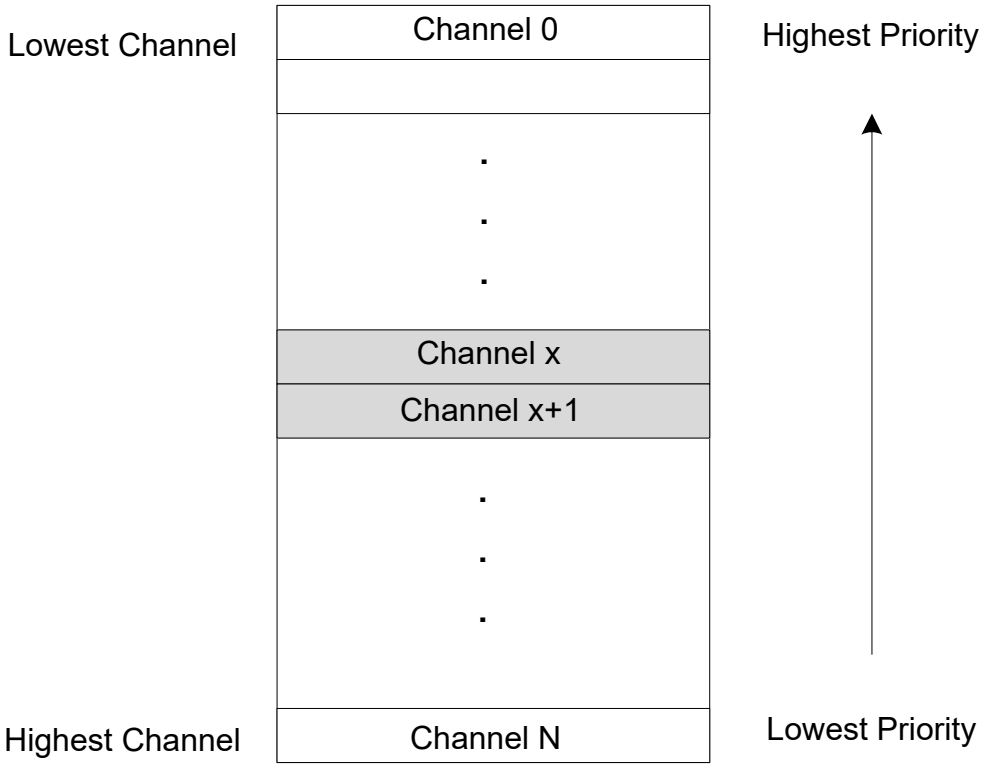
Figure 30-2. Interrupt Status Register



The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically the incoming events. The priority is evaluated each time a new channel has an interrupt pending or an interrupt is cleared. The Channel Pending Interrupt register (INTPEND) provides the channel number with the highest interrupt priority, the corresponding channel interrupt flags and status bits.

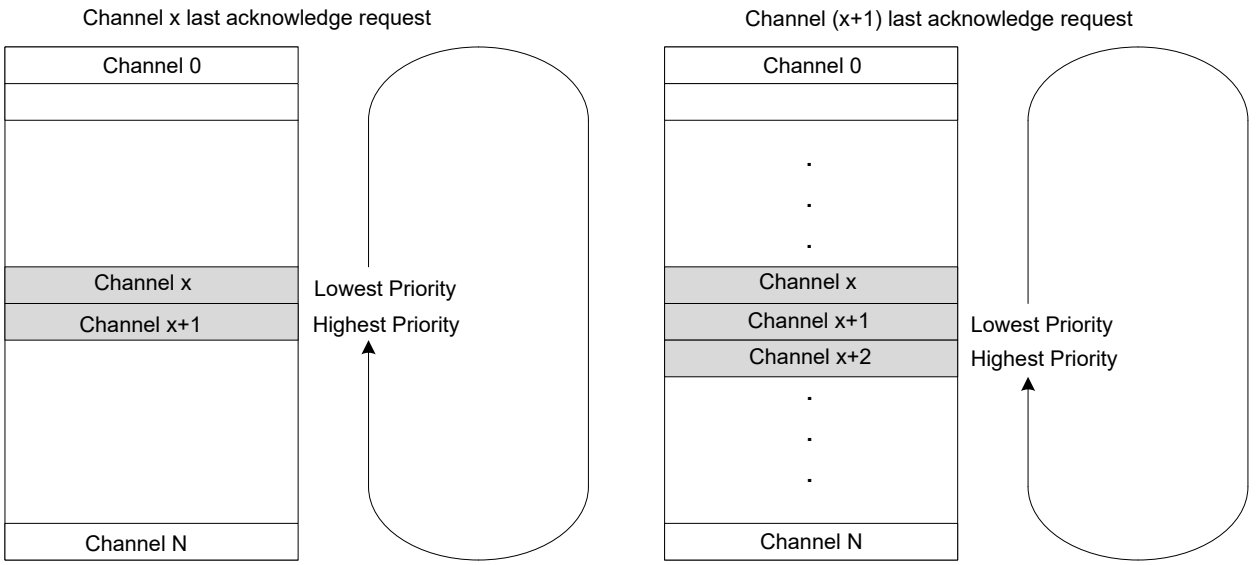
By default, static arbitration is enabled (PRICTRL.RREN is '0'), the arbiter will prioritize a low channel number over a high channel number as illustrated in Figure 30-3. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

Figure 30-3. Static Priority



The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing PRICTRL.RREN to one. 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, as illustrated in the following figure. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (PRICTRL.PRI).

Figure 30-4. Round-Robin Scheduling



The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, clears the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

30.6.3 Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel n interrupt (OVR)
- Event Detected Channel n interrupt (EVD)

These interrupts events are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the corresponding Channel n Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs.

Note: Interrupts must be globally enabled to allow the generation of interrupt requests.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Set (CHINTENSET) register and disabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Clear (CHINTENCLR) 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 interrupt flag is cleared, the interrupt is disabled or the Event System is reset. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel n 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 highest priority channel with the pending interrupt and the respective interrupt flags.

30.6.4 Sleep Mode Operation

The Event System can generate interrupts to wake up the device from the Idle or Standby Sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in the Channel register (CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path increases by two GCLK_EVSYS_CHANNEL_n clock (in other words, up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes depending on whether using CHANNELn.RUNSTDBY or CHANNELn.ONDEMAND:

Table 30-1. Event Channel Sleep Behavior

CHANNELn.PATH	CHANNELn. ONDEMAND	CHANNELn. RUNSTDBY	Sleep Behavior
ASYNC	0	0	Only run in Idle modes if an event must be propagated. Disabled in Standby Sleep mode.
SYNC/RESYNC	0	1	Run in both Idle and Standby Sleep modes.
SYNC/RESYNC	1	0	Only run in Idle modes if an event must be propagated. Disabled in Standby Sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.
SYNC/RESYNC	1	1	Run in both Idle and Standby Sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

30.7 Register Summary

See the *EVSYS* 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	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
		15:8	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		23:16	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
		31:24	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
0x08	PRICTRL	7:0	RREN			PRI[4:0]				
0x09 ... 0x0F	Reserved									
0x10	INTPEND	7:0					ID[3:0]			
		15:8	BUSY	READY					EVD	OVR
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
		15:8					CHINT11	CHINT10	CHINT9	CHINT8
		23:16								
		31:24								
0x18	BUSYCH	7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
		15:8					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
		23:16								
		31:24								
0x1C	READYUSR	7:0	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
		15:8					READYUSR11	READYUSR10	READYUSR9	READYUSR8
		23:16								
		31:24								
0x20	CHANNEL0	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x24	CHINTENCLR0	7:0							EVD	OVR
0x25	CHINTENSET0	7:0							EVD	OVR
0x26	CHINTFLAG0	7:0							EVD	OVR
0x27	CHSTATUSn0	7:0							BUSYCH	RDYUSR
0x28	CHANNEL1	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x2C	CHINTENCLR1	7:0							EVD	OVR
0x2D	CHINTENSET1	7:0							EVD	OVR
0x2E	CHINTFLAG1	7:0							EVD	OVR
0x2F	CHSTATUSn1	7:0							BUSYCH	RDYUSR
0x30	CHANNEL2	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x34	CHINTENCLR2	7:0							EVD	OVR
0x35	CHINTENSET2	7:0							EVD	OVR
0x36	CHINTFLAG2	7:0							EVD	OVR
0x37	CHSTATUSn2	7:0							BUSYCH	RDYUSR
0x38	CHANNEL3	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	CHINTENCLR3	7:0							EVD	OVR
0x3D	CHINTENSET3	7:0							EVD	OVR
0x3E	CHINTFLAG3	7:0							EVD	OVR
0x3F	CHSTATUSn3	7:0							BUSYCH	RDYUSR
0x40	CHANNEL4	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x44	CHINTENCLR4	7:0							EVD	OVR
0x45	CHINTENSET4	7:0							EVD	OVR
0x46	CHINTFLAG4	7:0							EVD	OVR
0x47	CHSTATUSn4	7:0							BUSYCH	RDYUSR
0x48	CHANNEL5	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x4C	CHINTENCLR5	7:0							EVD	OVR
0x4D	CHINTENSET5	7:0							EVD	OVR
0x4E	CHINTFLAG5	7:0							EVD	OVR
0x4F	CHSTATUSn5	7:0							BUSYCH	RDYUSR
0x50	CHANNEL6	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x54	CHINTENCLR6	7:0							EVD	OVR
0x55	CHINTENSET6	7:0							EVD	OVR
0x56	CHINTFLAG6	7:0							EVD	OVR
0x57	CHSTATUSn6	7:0							BUSYCH	RDYUSR
0x58	CHANNEL7	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x5C	CHINTENCLR7	7:0							EVD	OVR
0x5D	CHINTENSET7	7:0							EVD	OVR
0x5E	CHINTFLAG7	7:0							EVD	OVR
0x5F	CHSTATUSn7	7:0							BUSYCH	RDYUSR
0x60	CHANNEL8	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x64	CHINTENCLR8	7:0							EVD	OVR
0x65	CHINTENSET8	7:0							EVD	OVR
0x66	CHINTFLAG8	7:0							EVD	OVR
0x67	CHSTATUSn8	7:0							BUSYCH	RDYUSR
0x68	CHANNEL9	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x6C	CHINTENCLR9	7:0							EVD	OVR
0x6D	CHINTENSET9	7:0							EVD	OVR
0x6E	CHINTFLAG9	7:0							EVD	OVR
0x6F	CHSTATUSn9	7:0							BUSYCH	RDYUSR
0x70	CHANNEL10	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x74	CHINTENCLR10	7:0							EVD	OVR
0x75	CHINTENSET10	7:0							EVD	OVR
0x76	CHINTFLAG10	7:0							EVD	OVR
0x77	CHSTATUSn10	7:0							BUSYCH	RDYUSR

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	CHANNEL11	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x7C	CHINTENCLR11	7:0							EVD	OVR
0x7D	CHINTENSET11	7:0							EVD	OVR
0x7E	CHINTFLAG11	7:0							EVD	OVR
0x7F	CHSTATUSn11	7:0							BUSYCH	RDYUSR
0x80	CHANNEL12	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x84 ... 0x87	Reserved									
0x88	CHANNEL13	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x8C ... 0x8F	Reserved									
0x90	CHANNEL14	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x94 ... 0x97	Reserved									
0x98	CHANNEL15	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x9C ... 0x9F	Reserved									
0xA0	CHANNEL16	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xA4 ... 0xA7	Reserved									
0xA8	CHANNEL17	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xAC ... 0xAF	Reserved									
0xB0	CHANNEL18	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xB4 ... 0xB7	Reserved									
0xB8	CHANNEL19	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xBC ... 0xBF	Reserved									
0xC0	CHANNEL20	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xC4 ... 0xC7	Reserved									
0xC8	CHANNEL21	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xCC ... 0xCF	Reserved									
0xD0	CHANNEL22	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xD4 ... 0xD7	Reserved									
0xD8	CHANNEL23	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xDC ... 0xDF	Reserved									
0xE0	CHANNEL24	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xE4 ... 0xE7	Reserved									
0xE8	CHANNEL25	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xEC ... 0xEF	Reserved									
0xF0	CHANNEL26	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xF4 ... 0xF7	Reserved									
0xF8	CHANNEL27	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xFC ... 0xFF	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100	CHANNEL28	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x0104 ... 0x0107	Reserved									
0x0108	CHANNEL29	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x010C ... 0x010F	Reserved									
0x0110	CHANNEL30	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x0114 ... 0x0117	Reserved									
0x0118	CHANNEL31	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x011C ... 0x011F	Reserved									
0x0120	USER0	7:0			CHANNEL[5:0]					
...										
0x0158	USER56	7:0			CHANNEL[5:0]					

Related Links

8. Product Memory Mapping Overview

30.8 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.

Optional write protection by the PAC is denoted by the PAC Write-Protection property in each individual register description.

For more details, see *Register Access Protection* and *Peripheral Access Controller (PAC)* from Related Links.

Related Links

24. Peripheral Access Controller (PAC)

30.5.8. Register Access Protection

30.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state.
Note: Before applying a Software Reset, disabling the event generators is recommended.

30.8.2 Software Event

Name: SWEVT
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
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
	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
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
	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	W	W	W	W	W	W	W	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 – CHANNELx Channel x Software Selection [x=0..31]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit triggers a software event for channel x.
These bits always return '0' when read.

30.8.3 Priority Control

Name: PRICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RREN			PRI[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – RREN Round-Robin Scheduling Enable
For details on scheduling schemes, see *Interrupt Status and Interrupts Arbitration* from Related Links.

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 4:0 – PRI[4:0] Channel Priority Number
When round-robin arbitration is enabled (PRICTRL.RREN=1) for the priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.
When static arbitration is enabled (PRICTRL.RREN=0) for the priority level and the value of this bit group is nonzero, it does not affect the static priority scheme.
This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

Related Links
30.6.2.13. Interrupt Status and Interrupts Arbitration

30.8.4 Channel Pending Interrupt

Name: INTPEND
Offset: 0x10
Reset: 0x4000

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
	BUSY	READY					EVD	OVR
Access	R	R					RW	RW
Reset	0	1					0	0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit 15 – BUSY Busy
This bit is read as '1' when the event on a channel selected by the Channel ID field (ID) was not handled by all the event users connected to this channel.

Bit 14 – READY Ready
This bit is read as '1' when all event users connected to the channel selected by the Channel ID field (ID) are ready to handle incoming events on this channel.

Bit 9 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request is generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD bit will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bit 8 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request is generated if CHINTENCLR/SET.OVRx is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on the channel selected by the Channel ID field (ID) are not ready when a new event occurs
- An event happens when all the event users have not yet handled the previous event on the channel selected by the Channel ID field (ID)

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bits 3:0 – ID[3:0] Channel ID
These bits store the channel number of the highest priority.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

30.8.5 Interrupt Status

Name: INTSTATUS
Offset: 0x14
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
					CHINT11	CHINT10	CHINT9	CHINT8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHINTx Channel x Pending Interrupt
This bit is set when Channel x has a pending interrupt.
This bit is cleared when the corresponding Channel x interrupts are disabled, or the source interrupt sources are cleared.

30.8.6 Busy Channels

Name: BUSYCH
Offset: 0x18
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
					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – BUSYCH Busy Channel x
This bit is set if an event occurs on channel x that was not handled by all event users connected to channel x.
This bit is cleared when channel x is idle.
When the event channel x path is asynchronous, this bit is always read as '0'.

30.8.7 Ready Users

Name: READYUSR
Offset: 0x1C
Reset: 0x00000FFF

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
					READYUSR11	READYUSR10	READYUSR9	READYUSR8
Access					R	R	R	R
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – READYUSRn Ready User for Channel n
This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.
This bit is cleared when at least one of the event users connected to the channel is not ready.
When the event channel n path is asynchronous, this bit is always read zero.

30.8.8 Channel n Control

Name: CHANNEL
Offset: 0x20 + n*0x08 [n=0..31]
Reset: 0x00008000
Property: PAC Write-Protection

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

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
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	RW	RW			RW	RW	RW	RW
Reset	1	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
		EVGEN[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bit 15 – ONDEMAND Generic Clock On Demand
This bit is used to determine whether the generic clock is requested.

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in Standby
This bit is used to define the behavior during Standby Sleep mode.

Value	Description
0	The channel is disabled in Standby Sleep mode.
1	The channel is not stopped in Standby Sleep mode and depends on the CHANNEL.ONDEMAND bit.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection
These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0] Path Selection
These bits are used to choose which path is used by the selected channel.
Note: The path choice can be limited by the channel source; see *USERm* from Related Links.

Important: Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

Value	Name	Description
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
Other	—	Reserved

Bits 6:0 – EVGEN[6:0] Event Generator Selection
These bits are used to choose the event generator to connect to the selected channel.

Table 30-2. Event Generator Selection

Value	Name	Description
0x00	None	No event generator selected
0x01 - 0x08	RTC_PERx	RTC period x=0..7
0x09 - 0x0C	RTC_CMPx	RTC comparison x=0..3
0x0D	RTC_TAMPER	RTC tamper detection
0x0E	RTC_OVF	RTC overflow
0x0F - 0x12	EIC_EXTINTx	EIC external interrupt x=0..3
0x13 - 0x16	DMAC_CHx	DMA channel x=0..3
0x17	PAC_ACCERR	PAC Acc. error
0x18	TCC0_OVF	TCC0 overflow
0x19	TCC0_TRG	TCC0 trigger event
0x1A	TCC0_CNT	TCC0 counter
0x1B-0x20	TCC0_MCx	TCC0 match/compare x=0..5
0x21	TCC1_OVF	TCC1 overflow
0x22	TCC1_TRG	TCC1 trigger event
0x23	TCC1_CNT	TCC1 counter
0x24 - 0x29	TCC1_MCx	TCC1 match/compare x=0..5
0x2A	TCC2_OVF	TCC2 overflow
0x2B	TCC2_TRG	TCC2 trigger event
0x2C	TCC2_CNT	TCC2 counter
0x2D - 0x2E	TCC2_MCx	TCC2 match/compare x=0..1
0x2F	TC0_OVF	TC0 overflow
0x30-0x31	TC0_MCx	TC0 match/compare x=0..1
0x32	TC1_OVF	TC1 overflow
0x33 - 0x34	TC1_MCx	TC1 match/compare x=0..1
0x35	TC2_OVF	TC2 overflow
0x36 - 0x37	TC2_MCx	TC2 match/compare x=0..1
0x38	TC3_OVF	TC3 overflow
0x39 - 0x3A	TC3_MCx	TC3 match/compare x=0..1
0x3B	TC4_OVF	TC4 overflow
0x3C - 0x3D	TC4_MCx	TC4 match/compare x=0..1
0x3E	TC5_OVF	TC5 overflow
0x3F - 0x40	TC5_MCx	TC5 match/compare x=0..1
0x41	TC6_OVF	TC6 overflow
0x42 - 0x43	TC6_MCx	TC6 match/compare x=0..1
0x44	TC7_OVF	TC7 overflow
0x45 - 0x46	TC7_MCx	TC7 match/compare x=0..1
0x47	ADC_RESRDY	ADC end-of-scan ready interrupt
0x48 - 0x49	Not used	—
0x4A - 0x4B	AC_COMPx	AC comparator, x=0..1
0x4C	AC_WIN_0	AC0 window

.....continued		
Value	Name	Description
0x4D	Not used	—
0x4E - 0x4F	CCL_LUTOUTx	CCL LUTOUT x-0..1
0x50	ZB_TX_TS_ACTIVE	ZB transmit packet active time
0x51	ZB_RX_TS_ACTIVE	ZB receive packet active time

Related Links

30.8.13. USERm

30.8.9 Channel n Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x24 + n*0x08 [n=0..11]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

30.8.10 Channel n Interrupt Enable Set

Name: CHINTENSET
Offset: 0x25 + n*0x08 [n=0..11]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Event Detected Channel Interrupt Enable bit, which enables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Channel Interrupt Enable bit, which enables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

30.8.11 Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x26 + n*0x08 [n=0..11]
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

Bit 0 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVR is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel interrupt flag.

30.8.12 Channel n Status

Name: CHSTATUSn
Offset: 0x27 + n*0x08 [n=0..11]
Reset: 0x01

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

Bit 1 – BUSYCH Busy Channel
This bit is cleared when the channel is idle.
This bit is set if an event on channel was not handled by all event users connected to the channel.
When the event channel path is asynchronous, this bit is always read as '0'.

Bit 0 – RDYUSR Ready User
This bit is cleared when at least one of the event users connected to the channel is not ready.
This bit is set when all event users connected to the channel are ready to handle incoming events on the channel.
When the event channel path is asynchronous, this bit is always read as '0'.

30.8.13 Event User m

Name: USERm
Offset: 0x0120 + m*0x01 [m=0..56]
Reset: 0x0
Property: PAC Write-Protection

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

Bits 5:0 – CHANNEL[5:0] Channel Event Selection
These bits select channel n to connect to the event user m.
Note: A value x of this bit field selects channel n = x-1.

Table 30-3. User Multiplexer Number m

USER*m*	User Multiplexer	Description	Path Type(1)
m = 0	RTC_TAMPER	RTC Tamper	A, S, R
m = 1..8	DMAC_CH0..7	Channel 0..7	S, R
m = 9	CM4_TRACE_START	CM4 trace start	A, S, R
m = 10	CM4_TRACE_STOP	CM4 trace stop	A, S, R
m = 11	CM4_TRACE_TRIG	CM4 trace trigger	A, S, R
m = 12..13	TCC0 EV0..1	TCC0 EVx	A, S, R
m = 14..19	TCC0 MC0..5	TCC0 MCx	A, S, R
m = 20..21	TCC1 EV0..1	TCC1 EVx	A, S, R
m = 22..27	TCC1 MC0..5	TCC1 MCx	A, S, R
m = 28..29	TCC2 EV0..1	TCC2 EVx	A, S, R
m = 30..31	TCC2 MC0..1	TCC2 MCx	A, S, R
m = 32	TC0 EVU	TC0 EVU	A, S, R
m = 33	TC1 EVU	TC1 EVU	A, S, R
m = 34	TC2 EVU	TC2 EVU	A, S, R
m = 35	TC3 EVU	TC3 EVU	A, S, R
m = 36	TC4 EVU	TC4 EVU	A, S, R
m = 37	TC5 EVU	TC5 EVU	A, S, R
m = 38	TC6 EVU	TC6 EVU	A, S, R
m = 39	TC7 EVU	TC7 EVU	A, S, R
m = 40..51	ADC_TRIGGER5..16	ADC_TRIGGERx	A
m = 52..53	AC_SOC0..1	AC_SOCx	A, S, R
m = 54..55	CCL_LUTIN0..1	CCL_LUTINx	A, S, R
m = 56	CVD_TRIGGER	CVD_TRIGGER	A

1) A = Asynchronous path, S = Synchronous path, R = Resynchronized path

31. Serial Communication Interface (SERCOM)

31.1 Overview

The device supports up to three SERCOM modules. Two SERCOM's (SERCOM0/1) can be configured to support a number of modes: I^2C, SPI and USART. One of the SERCOM (SERCOM2) has only I^2C functionality. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance is dedicated to the selected mode.

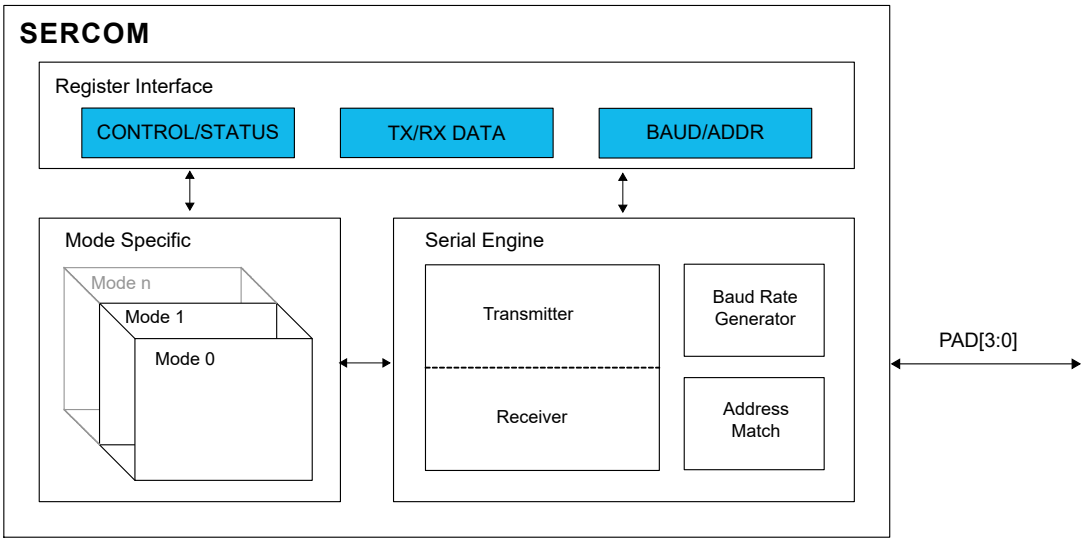
The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address-matching functionality and mode-specific transmitter and receiver logic. It can use the internal generic clock or an external clock.

31.2 Features

- Interface for Configuring into One of the Following:
  - I^2C two-wire serial interface
  - SPI
  - USART
  - SMBus™ compatible
- Single Transmit Buffer and Double Receive Buffer
- Baud-Rate Generator
- Address Match/Mask Logic
- Operational in Idle and Standby Sleep Mode with an External Clock Source
- Can be Used with DMA

31.3 Block Diagram

Figure 31-1. SERCOM Block Diagram



31.4 Signal Description

See the respective SERCOM mode chapters for details.

31.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

31.5.1 I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using the System Configuration registers or PPS.

The SERCOM has four internal pads, PAD[3:0], and the signals from I^2C, SPI and USART are routed through these SERCOM pads through a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes.

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

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

31.5.2 Power Management

The SERCOM can operate in any Sleep mode (Idle, Standby Sleep) provided the selected clock source is running. SERCOM interrupts can be configured to wake the device from sleep modes.

31.5.3 Clocks

The SERCOM uses two generic clocks:

- GCLK_SERCOMx_CORE – The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a host.
- GCLK_SERCOMx_SLOW – The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM. See *Clock and Reset Unit (CRU)* from Related Links.

The generic clocks are asynchronous to the bus clock (PBx_CLK). Therefore, writing to certain registers requireS synchronization between the clock domains.

Related Links

17. Clock and Reset Unit (CRU)

31.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used. See *Direct Memory Access Controller (DMAC)* from Related Links.

Related Links

26. Direct Memory Access Controller (DMAC)

31.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

31.5.6 Events

Not applicable.

31.5.7 Debug Operation

When the CPU is halted in Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging; see *DBGCTRL* register from Related Links.

Related Links

33.8.13. DBGCTRL

31.5.8 Register Access Protection

Registers with write access can be write protected optionally by the PAC.

Optional write protection by the 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.

31.5.9 Analog Connections

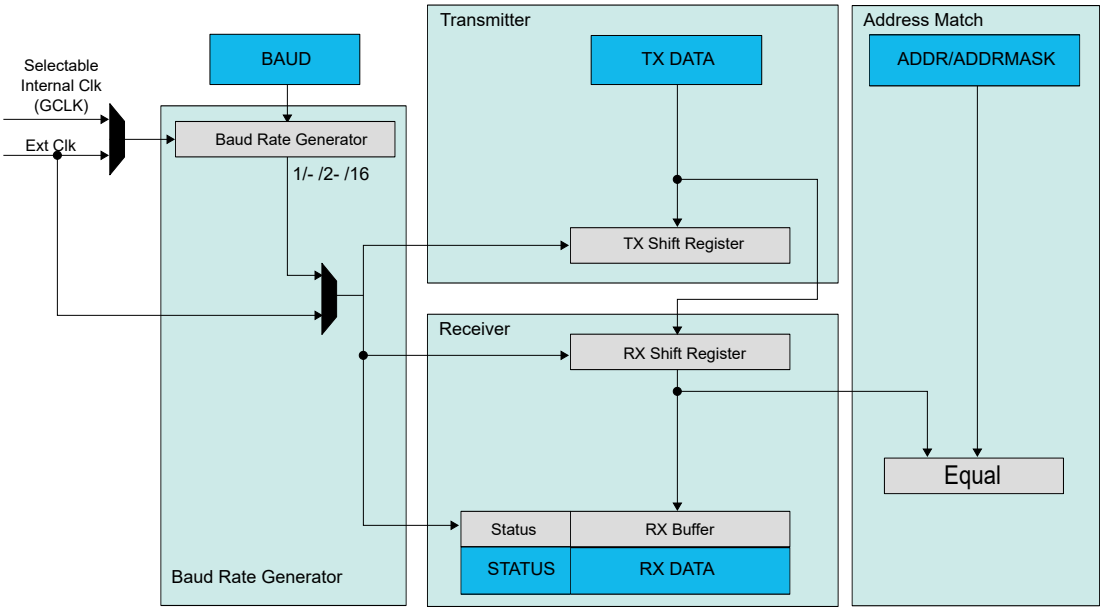
Not applicable.

31.6 Functional Description

31.6.1 Principle of Operation

The basic structure of the SERCOM serial engine is shown in *SERCOM Serial Engine*. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

Figure 31-2. SERCOM Serial Engine



The transmitter consists of a single write buffer and a Shift register.

The receiver consists of a one-level (I^2C), or two-level (USART, SPI) receive buffer and a Shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

31.6.2 Basic Operation

31.6.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE) as shown in the table below.

Table 31-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in client operation
0x3	SPI in host operation
0x4	I^2C client operation
0x5	I^2C host operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

31.6.2.2 Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

See the *CTRLA* register from Related Links.

Related Links

33.8.1. CTRLA

31.6.2.3 Clock Generation – Baud-Rate Generator

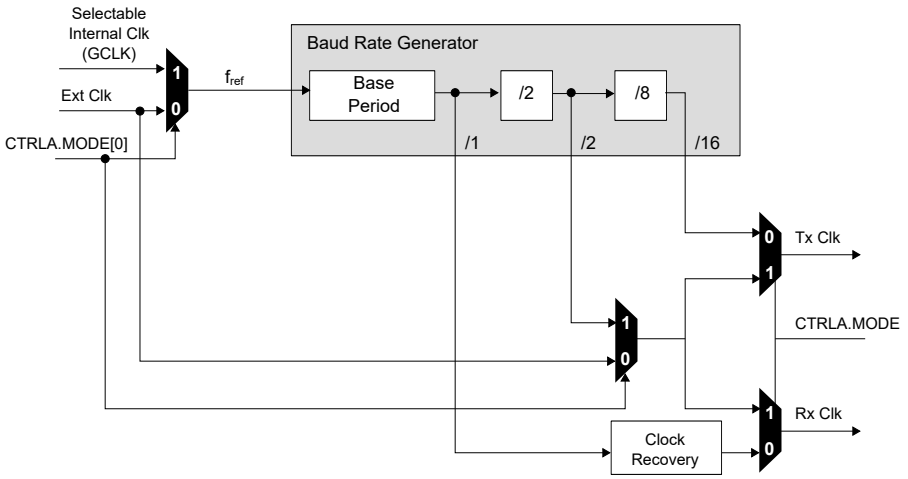
The baud-rate generator, as illustrated in the following figure, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas, the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 31-3. Baud Rate Generator



The following table contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there are two modes:

- Arithmetic mode – The BAUD register value is 16 bits (0 to 65,535)
- Fractional mode – The BAUD register value is 13 bits, while the fractional adjustment is 3 bits. In this mode, the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 31-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit, which can be 16, 8 or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

31.6.3 Additional Features

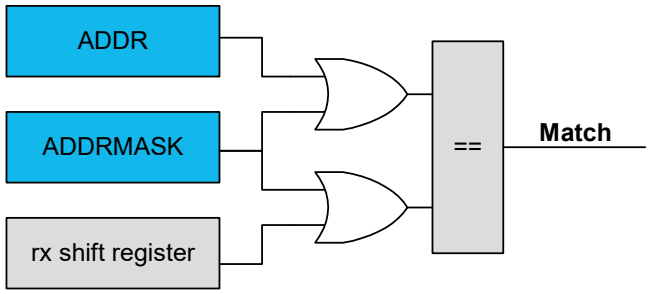
31.6.3.1 Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

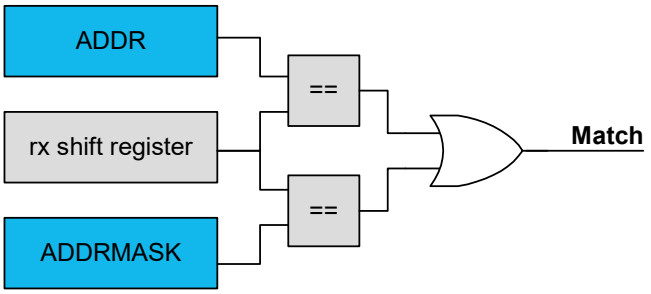
Figure 31-4. Address With Mask



Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Figure 31-5. Two Unique Addresses



Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

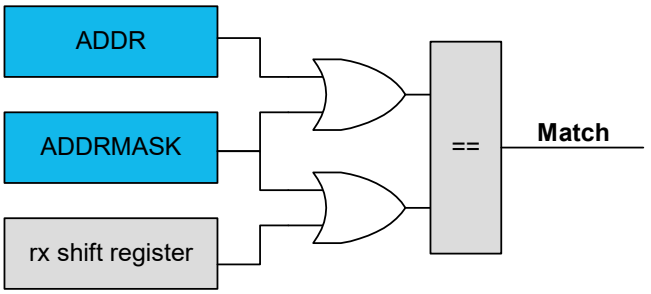
Figure 31-6. Address Range

31.6.3.1.1 Address Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match.

Note: Writing the ADDR.ADDRMASK to “all zeros” will match a single unique address, while writing ADDR.ADDRMASK to “all ones” will result in all addresses being accepted.

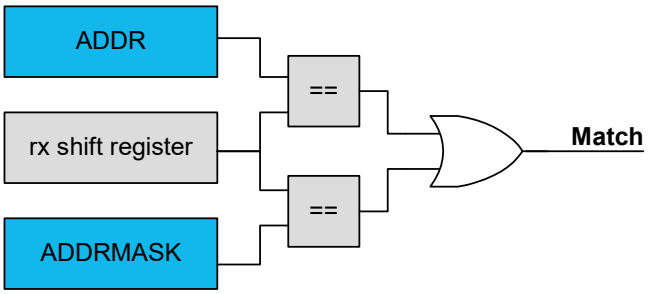
Figure 31-7. Address With Mask



31.6.3.1.2 Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Figure 31-8. Two Unique Addresses



31.6.3.1.3 Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 31-9. Address Range

31.6.4 DMA Operation

The available DMA interrupts depend on the operation mode of the SERCOM peripheral. Refer to the Functional Description sections of the respective SERCOM mode.

31.6.5 Interrupts

Interrupt sources are mode specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own Interrupt flag.

The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing Interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which Interrupt condition occurred. 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.

31.6.6 Events

Not applicable.

31.6.7 Sleep Mode Operation

The peripheral can operate in any sleep mode (Idle, Standby Sleep) where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

31.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

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.

32. SERCOM Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART)

32.1 Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver (see 32.3. Block Diagram). Labels in uppercase letters are synchronous to PB1_CLK and accessible for the CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a Shift register and control logic for different frame formats. The write buffer supports data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a Shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

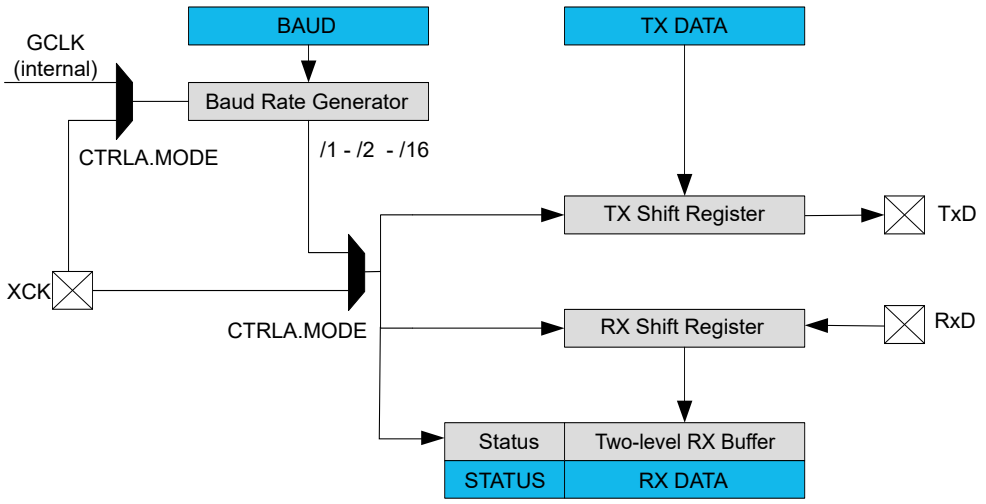
32.2 USART Features

- Full-Duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock Source for Asynchronous and Synchronous Operation
- Baud-Rate Generator
- Supports Serial Frames with 5, 6, 7, 8 or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check
- Selectable LSB- or MSB- First Data Transfer
- Buffer Overflow and Frame Error Detection
- Noise Filtering, Including False Start-Bit Detection and Digital Low-Pass Filter
- Collision Detection
- Can Operate in All Sleep Modes
- Operation at Speeds Up to Half the System Clock for Internally-Generated Clocks
- Operation at Speeds Up to the System Clock for Externally-Generated Clocks
- RTS and CTS Flow Control
- IrDA Modulation and Demodulation Up to 115.2 kbps
- LIN Host Support
- LIN Client Support
  - Auto-baud and break character detection
- ISO 7816 T = 0 or T = 1 Protocols for Smart Card Interfacing
  • RS485 Support
- Start-of-Frame Detection
- Two-Level Receive Buffer
- Can Work with DMA
- 32-Bit Extension for Better System Bus Utilization

Note: SERCOM2 does not have USART functionality.

32.3 Block Diagram

Figure 32-1. USART Block Diagram



32.4 Signal Description

Table 32-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Note: For details on pin mapping, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

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

32.5 Product Dependencies

The following sections describe how the other parts of the system must be configured to correctly use this peripheral.

32.5.1 I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the System Configuration registers (See *System Configuration and Register Locking (CFG)* from Related Links.) (SCOM_HSEN[1:0] of CFGCON1/DEVCFG1 register). If SERCOM pins are selected through PPS, the PPS registers have to be configured. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

If SCOMx_HSEN = 1, SERCOM uses dedicated pins.

If SCOMx_HSEN = 0, SERCOM uses the PPS path and I/O pins are multiplexed to pin groups defined in the PPS section.

When the SERCOM is used in the USART mode, the SERCOM controls the direction and value of the I/O pins according to the following table. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 32-2. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input

.....continued	
Pin	Pin Configuration
XCK	Output or input

The configuration of the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) defines the physical position of the USART signals in the above table.

Related Links

6. I/O Ports and Peripheral Pin Select (PPS)
22. System Configuration and Register Locking (CFG)

32.5.2 Power Management

This peripheral can continue to operate in any sleep mode (Idle, Standby Sleep) where its source clock is running. The interrupts can wake up the device from sleep modes.

32.5.3 Clocks

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the CRU registers before using it for SERCOMx_CORE. See *Clock and Reset Unit (CRU)* and *Overview* from Related Links.

This generic clock is asynchronous to the bus clock (PB1_CLK). Therefore, writing to certain registers requires synchronization to the clock domains.

Related Links

17. Clock and Reset Unit (CRU)
32.1. Overview

32.5.4 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 (DMAC)* from Related Links).

Related Links

26. Direct Memory Access Controller (DMAC)

32.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the NVIC must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

32.5.6 Events

Not applicable.

32.5.7 Debug Operation

When the CPU is halted in the Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See *DBGCTRL* register from Related Links.

Related Links

32.8.14. DBGCTRL

32.5.8 Register Access Protection

Registers with write access can be write-protected optionally by the PAC.

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

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

32.5.9 Analog Connections

Not applicable.

32.6 Functional Description

32.6.1 Principle of Operation

The USART uses the following lines for data transfer:

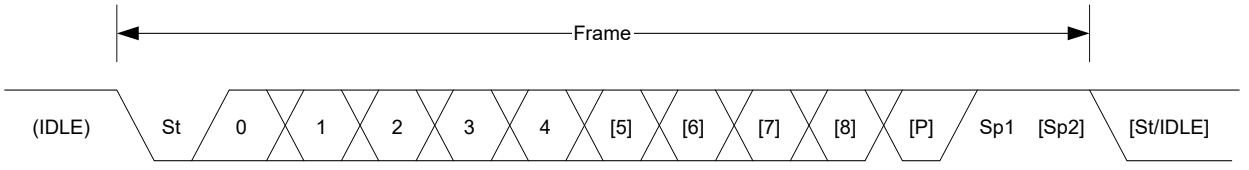
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the Start bit followed by one character of Data bits. If enabled, the parity bit is inserted after the Data bits and before the first Stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately or the communication line can return to the Idle (high) state. The following figure illustrates the possible frame formats. Values inside brackets ([x]) denote optional bits.

Figure 32-2. Frame Formats



St	Start bit. Signal is always low.
n, [n]	Data bits. 0 to [4..8]
[P]	Parity bit. Either odd or even.
Sp, [Sp]	Stop bit. Signal is always high.
IDLE	No frame is transferred on the communication line. Signal is always high in this state.

32.6.2 Basic Operation

32.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits
- Baud register (BAUD)

Any writes to these registers when the USART is enabled or is being enabled (CTRL.ENABLE is one) are discarded. Writes to these registers while the peripheral is being disabled is completed after the disabling is complete.

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers is discarded. If the peripheral is being disabled, writing to these registers is executed after disabling is completed. Enable protection is denoted by the Enable-Protection property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either the external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either Asynchronous (0) or Synchronous (1) Communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select the pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for the character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use Parity mode:
   a. Enable Parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
   b. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

32.6.2.2 Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

See the *CTRLA* register from Related Links.

Related Links

32.8.1. CTRLA

32.6.2.3 Clock Generation and Selection

For both Synchronous and Asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The Synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the Asynchronous mode is selected by writing '0' to CTRLA.CMODE.

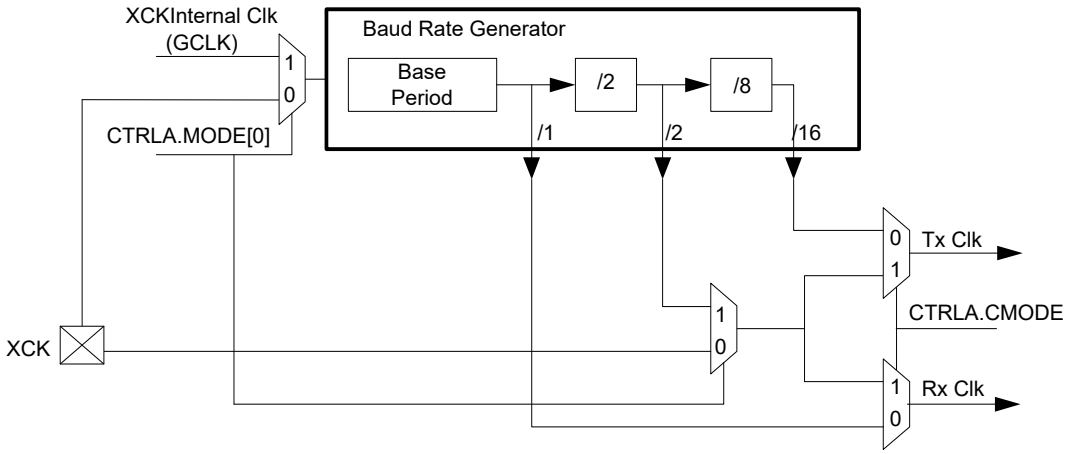
The internal clock source is selected by writing '1' to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing '0' to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the following figure.

In Asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In Synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. For more details on configuring the baud rate (see *Clock Generation – Baud-Rate Generator* from Related Links).

Figure 32-3. Clock Generation



Related Links

31.6.2.3. Clock Generation – Baud-Rate Generator

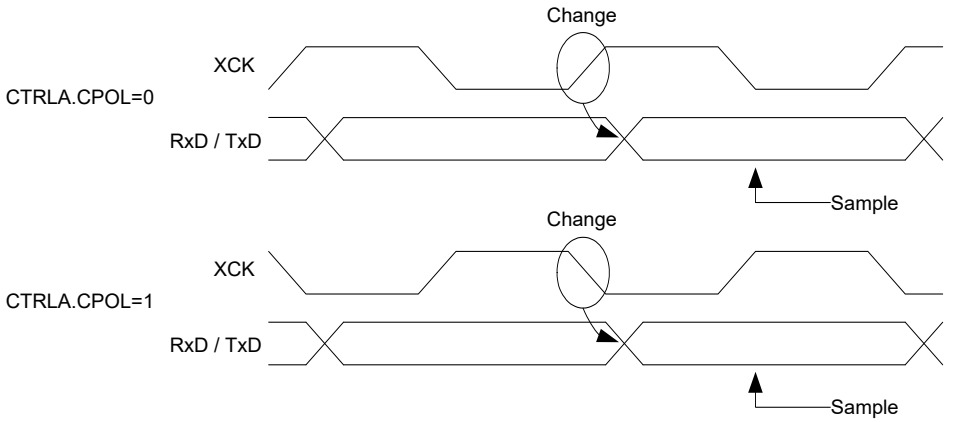
32.6.2.3.1 Synchronous Clock Operation

In Synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling and which is used for TxD change:

- When CTRLA.CPOL is '0', the data is changed on the rising edge of XCK and sampled on the falling edge of XCK.
- When CTRLA.CPOL is '1', the data is changed on the falling edge of XCK and sampled on the rising edge of XCK.

Figure 32-4. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE = 0x0), the Shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

32.6.2.4 Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

32.6.2.5 Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit has been transmitted (both the Tx buffer and the shift register are empty) and no new data was written to DATA register, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

32.6.2.5.1 Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter completes only after any ongoing and pending transmissions are completed; in other words, there are no data in the Transmit Shift register and TxDATA to transmit.

32.6.2.6 Data Reception

The receiver accepts data when a valid Start bit is detected. Each bit following the Start bit is sampled according to the baud rate or XCK clock and shifted into the receive Shift register until the first Stop bit of a frame is received. The second Stop bit is ignored by the receiver.

When the first Stop bit is received and a complete serial frame is present in the Receive Shift register, the contents of the Shift register are moved into the two-level receive buffer. Then, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set and the optional interrupt is generated.

The received data can be read from the DATA register when the Receive Complete Interrupt flag is set.

32.6.2.6.1 Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) disables the receiver, flushes the two-level receive buffer and data from ongoing receptions are lost.

32.6.2.6.2 Error Bits

The USART receiver has three error bits in the Status (STATUS) register:

- Frame Error (FERR)
- Buffer Overflow (BUFOVF)
- Parity Error (PERR)

When an error happens, the corresponding error bit is set until it is cleared by writing '1' to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

- When CTRLA.IBON = 1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can, then, empty the receive FIFO by reading RxDATA until the Receiver Complete Interrupt flag (INTFLAG.RXC) is cleared.
- When CTRLA.IBON = 0, the Buffer Overflow condition is attending data through the receive FIFO, which, then, sets the INTFLAG.ERROR bit. After the received data are read, STATUS.BUFOVF (and INTFLAG.ERROR) is set along with INTFLAG.RXC.

32.6.2.6.3 Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

32.6.2.6.4 Asynchronous Operational Range

The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames and the frame size (in number of bits). In addition, the operational range of the receiver depends on the difference between the received bit rate and the internally-generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally-generated baud rate, the receiver is not able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in the baud rate:

- First, the reference clock always has some minor instability.
- Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired.

In this case, the BAUD register value must be set to give the lowest possible error. See *Clock Generation – Baud-Rate Generator* from Related Links.

The following table provides an overview of the recommended maximum receiver baud-rate errors for various character sizes.

Table 32-3. Asynchronous Receiver Error for 16-fold Oversampling

D (Data Bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. Total Error [%]	Recommended Max. Rx Error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0

.....continued				
D (Data Bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. Total Error [%]	Recommended Max. Rx Error [%]
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

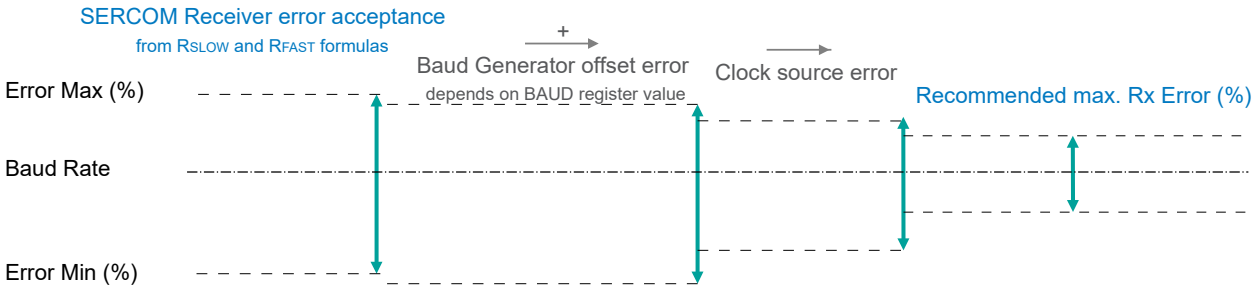
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate.
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate.
- D is the sum of the character size and parity size (D = 5 to 10 bits).
- S is the number of samples per bit (S = 16, 8 or 3).
- S_F is the first sample number used for majority voting (S_F = 7, 3 or 2) when CTRLA.SAMPA = 0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4 or 2) when CTRLA.SAMPA = 0.

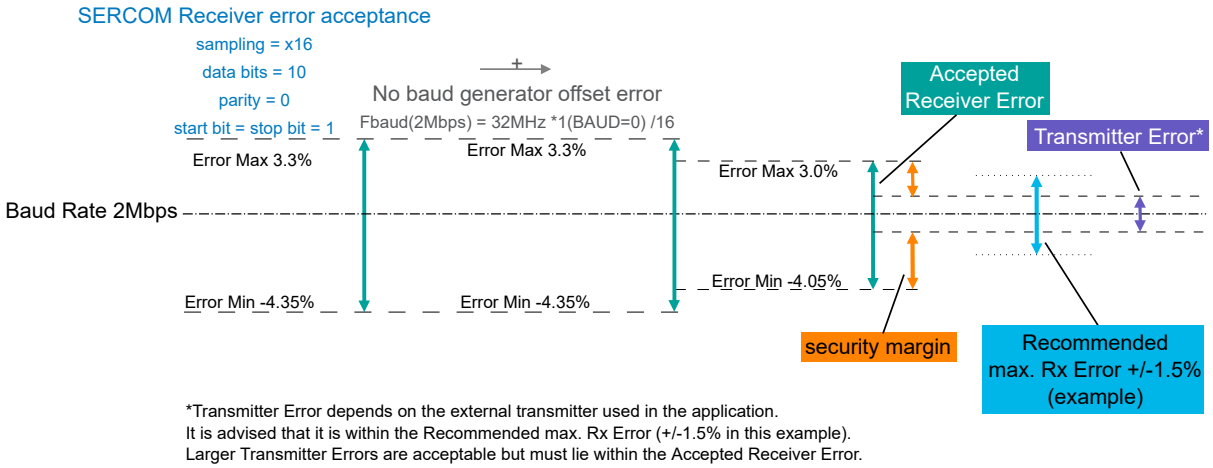
The recommended maximum RX Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 32-5. USART RX Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure illustrates an example for a baud rate of 3 Mbps:

Figure 32-6. USART RX Error Calculation Example



Related Links

31.6.2.3. Clock Generation – Baud-Rate Generator

32.6.3 Additional Features

32.6.3.1 Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If even parity is selected (CTRLB.PMODE = 0), the Parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

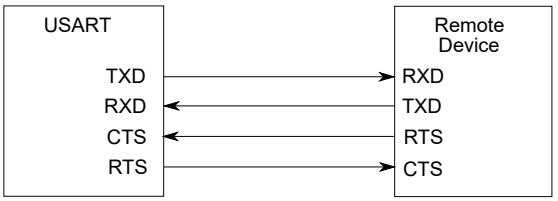
If odd parity is selected (CTRLB.PMODE = 1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '1', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the Parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

32.6.3.2 Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as illustrated in the following figure.

Figure 32-7. Connection with a Remote Device for Hardware Handshaking



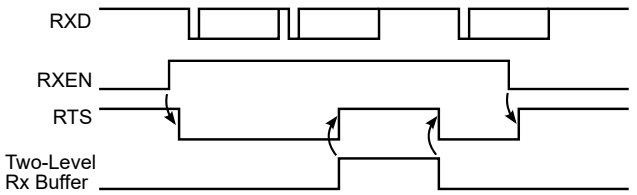
Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE = 1)
- Asynchronous mode (CTRLA.CMODE = 0)
- Flow control pinout (CTRLA.TXPO = 2)

When the receiver is disabled or the receive FIFO is full, the receiver drives the RTS pin high. This notifies the remote device to stop the transfer after the ongoing transmission. Enabling and

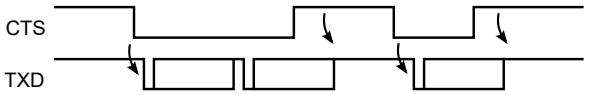
disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS is set immediately and the frame being received is stored in the Shift register until the receive FIFO is no longer full.

Figure 32-8. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission starts only if STATUS.CTS = 0. When CTS is set, the transmitter completes the ongoing transmission and stops transmitting.

Figure 32-9. Transmitter Behavior when Operating with Hardware Handshaking



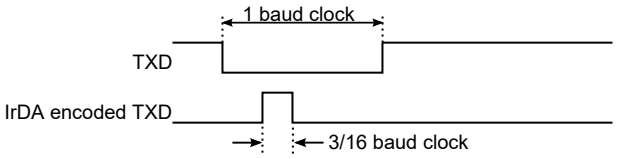
32.6.3.3 IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kbps. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC = 1)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate (CTRLA.SAMPR[0] = 0)

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the following figure.

Figure 32-10. IrDA Transmit Encoding



The reception decoder has two main functions:

- The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.
- The second main function is to decode incoming RX data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

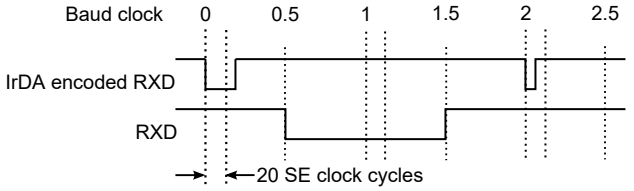
Notes: The polarity of the transmitter and receiver are opposite:

- During transmission, a '0' bit is transmitted as a '1' pulse
- During reception, an accepted '0' pulse is received as a '0' bit

> **Example:** The following figure illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width must be at least 20 SE clock cycles. When using BAUD = 0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as minimum

pulse width required. In this case, the first bit is accepted as a '0', the second bit is a '1' and the third bit is also a '1'. A low pulse is rejected because it does not meet the minimum requirement of 2/16 baud clock.

Figure 32-11. IrDA Receive Decoding



32.6.3.4 Break Character Detection and Auto-Baud

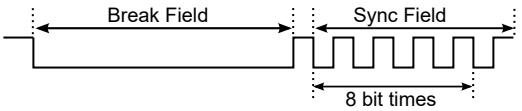
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)

The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a break field. When a break field is detected, the Receive Break Interrupt Flag (INTFLAG.RXBRK) is set and the USART expects the sync field character to be 0x55. This field is used to update the actual baud rate to stay synchronized. If the received sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error Interrupt Flag (INTFLAG.ERROR) and the baud rate is unchanged.

The auto-baud follows the LIN format. All LIN Frames start with a break field followed by a sync field.

Figure 32-12. LIN Break and Sync Fields



After a break field is detected and the Start bit of the sync field is detected, a counter is started. The counter is, then, incremented for the next 8 bit times of the sync field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 MSb of the counter (value divided by 8) gives the new clock divider (BAUD.BAUD) and the 3 LSb of this value (the remainder) gives the new Fractional Part (BAUD.FP).

When the sync field is received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the break and sync fields are received, multiple characters of data can be received.

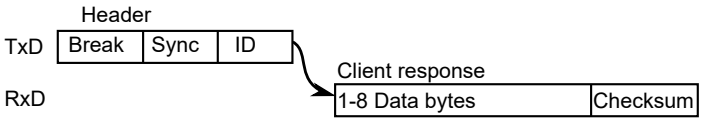
32.6.3.5 LIN Host

LIN Host is available with the following configuration:

- LIN Host format (CTRLA.FORM = 0x02)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)

LIN frames start with a header transmitted by the Host. The header consists of the break, sync and identifier fields. After the Host transmits the header, the addressed Client responds with 1-8 bytes of data plus checksum.

Figure 32-13. LIN Frame Format



Using the LIN command field (CTRLB.LINCMD), the complete header can be automatically transmitted or software can control transmission of the various header components.

When CTRLB.LINCMD = 0x1, software controls transmission of the LIN header. In this case, software uses the following sequence:

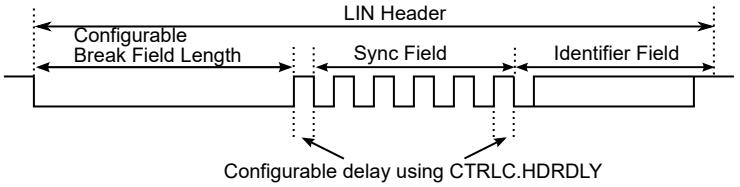
- CTRLB.LINCMD is written to 0x1.
- DATA register is written to 0x00. This triggers transmission of the break field by hardware.
  Note: Writing the DATA register with any other value is also result in the transmission of the break field by hardware.
- DATA register is written to 0x55. The 0x55 value (sync) is transmitted.
- DATA register is written to the identifier. The identifier is transmitted.

When CTRLB.LINCMD = 0x2, hardware controls transmission of the LIN header. In this case, software uses the following sequence:

- CTRLB.LINCMD is written to 0x2.
- DATA register is written to the identifier. This triggers the transmission of the complete header by hardware. First, the break field is transmitted. Next, the sync field is transmitted and, finally, the identifier is transmitted.

In LIN Host mode, the length of the break field is programmable using the break length field (CTRLC.BRKLEN). When the LIN header command is used (CTRLB.LINCMD = 0x2), the delay between the break and sync fields, in addition to the delay between the sync and ID fields, are configurable using the header delay field (CTRLC.HDRDLY). When manual transmission is used (CTRLB.LINCMD = 0x1), software controls the delay between break and sync.

Figure 32-14. LIN Header Generation



After header transmission is complete, the Client responds with 1-8 data bytes plus checksum.

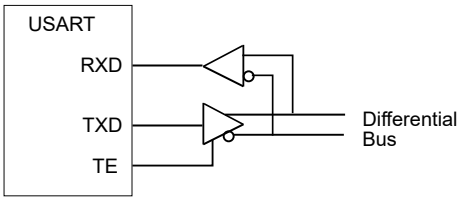
32.6.3.6 RS485

RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO = 0x3)

The RS485 feature enables control of an external line driver as illustrated in the following figure. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

Figure 32-15. RS485 Bus Connection



The TE pin remains high for the complete frame including the stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line remains driven after the last character completion. The following figure illustrates a transfer with one stop bit and CTRLC.GTIME = 3.

Figure 32-16. Example of TE Drive with Guard Time

The Transmit Complete interrupt flag (INTFLAG.TXC) is raised after the guard time is complete and TE goes low.

32.6.3.7 ISO 7816 for Smart Card Interfacing

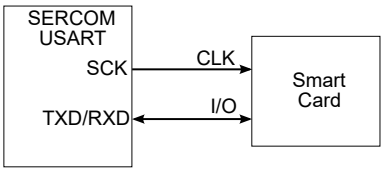
The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T = 0 and T = 1 protocols defined by the ISO 7816 specification are supported.

ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV = 1 and CTRLA.TXINV = 1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE = 0)
- 8-bit character size (CTRLB.CHSIZE = 0)
- T = 0 (CTRLA.CMODE = 1) or T = 1 (CTRLA.CMODE = 0)

ISO 7816 is a half-duplex communication on a single bidirectional line. The USART connects to a smart card as shown below. The output is only driven when the USART is transmitting. The USART is considered the host of the communication as it generates the clock.

Figure 32-17. Connection of a Smart Card to the SERCOM USART



ISO 7816 characters are specified as 8 bits with even parity. The USART must be configured accordingly.

The USART cannot operate concurrently in both receiver and transmitter modes as the communication is unidirectional. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO 7816 mode may lead to unpredictable results.

The ISO 7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value (CTRLA.RXINV = 1 and CTRLA.TXINV = 1).

Protocol T = 0

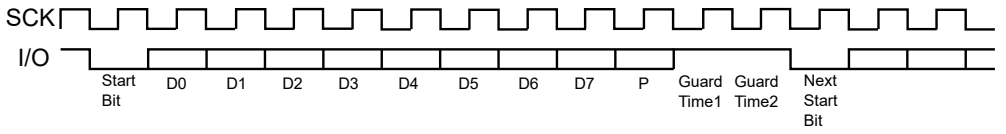
In T = 0 protocol, a character is made up of:

- One start bit
- Eight data bits
- One parity bit
- One guard time, which lasts two bit times

The transfer is synchronous (CTRLA.CMODE = 1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

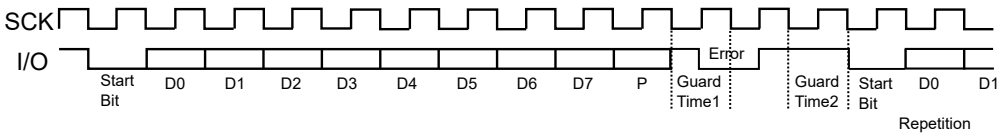
If no parity error is detected, the I/O line remains '1' during the guard time and the transmitter can continue with the transmission of the next character, as illustrated in the following figure.

Figure 32-18. T = 0 Protocol without Parity Error



If a parity error is detected by the receiver, it drives the I/O line to '0' during the guard time, as illustrated in the following figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts '1' bit time more, as the guard time length is the same and is added to the error bit time, which lasts '1' bit time.

Figure 32-19. T = 0 Protocol with Parity Error



When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

Receive Error Counter

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

Receive NACK Inhibit

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is '1', no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO and the STATUS.PERR bit is set. Inhibit Not Acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

When the USART repetition number reaches the programmed value in CTRLC.MAXITER, the STATUS.ITER bit is set and the internal iteration counter is Reset. If the repetition of the character is acknowledged by the receiver before the maximum iteration is reached, the repetitions are stopped and the iteration counter is cleared.

Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the Disable Successive NACK bit (CTRLC.DSNACK). The maximum number of NACKs transmitted is programmed in the CTRLC.MAXITER field. As soon as the maximum is reached, the character is considered correct, an acknowledge is sent on the line, the STATUS.ITER bit is set and the internal iteration counter is Reset.

Protocol T = 1

When operating in ISO7816 protocol T = 1, the transmission is asynchronous (CTRL1.CMODE = 0) with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

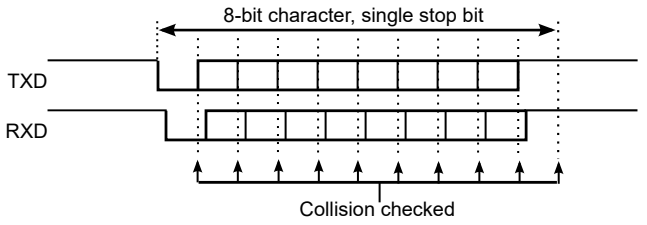
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit and the erroneous character is written to the receive FIFO. When using T = 1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

32.6.3.8 Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN = 1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN = 1 and CTRLB.TXEN = 1).

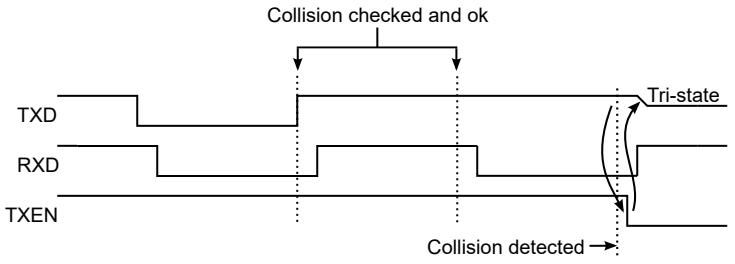
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as illustrated in the following figure. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 32-20. Collision Checking



The following figure illustrates the conditions for a collision detection. In this case, the Start bit and the first Data bit are received with the same value as transmitted. The second received Data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 32-21. Collision Detected



When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.

2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN = 0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is set until this is complete.
   - After disabling, the TxD pin is tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error Interrupt Flag (INTFLAG.ERROR).
5. Set the Transmit Complete Interrupt Flag (INTFLAG.TXC) because the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after confirming that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

32.6.3.9 Loop-Back Mode

For Loop-Back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

32.6.3.10 Start-of-Frame Detection

The USART start-of-frame detector can wake up the CPU when it detects a Start bit. In Standby Sleep mode, the internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8 MHz Internal Oscillator is powered up and the USART clock is enabled. After start-up, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast start-up internal oscillator start-up time. See *Electrical Characteristics* from Related Links for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in Asynchronous and Synchronous modes. It is enabled by writing '1' to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8 MHz internal oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the receive complete interrupt is generated.

Related Links

43. Electrical Characteristics

32.6.3.11 Sample Adjustment

In Asynchronous mode (CTRLA.CMODE = 0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in the Control A register (CTRLA.SAMPA). When CTRLA.SAMPA = 0, samples 7-8-9 are used for 16x oversampling and samples 3-4-5 are used for 8x oversampling.

32.6.3.12 32-Bit Extension

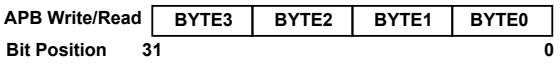
For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B). When enabled, writes and/or reads to the DATA register are 32-bit in size.

If frames are not multiples of 4 bytes, the length counter (LENGTH.LEN) and length enable (LENGTH.LENEN) must be configured before the data transfer begins, and LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using a 32-bit extension. Bytes are transmitted or received and stored in order from 0 to 3. Only 8-bit and smaller character

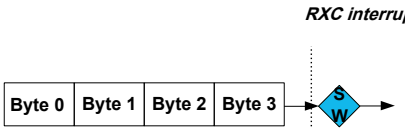
sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte is zero for received data and unused for transmit data.

Figure 32-22. 32-bit Extension Ordering



A receive transaction using a 32-bit extension is illustrated in the following figure. The Receive Complete flag (INTFLAG.RXC) is raised every four received bytes. For transmit transactions, the Data Register Empty flag (INTFLAG.DRE) is raised instead of INTFLAG.RXC.

Figure 32-23. 32-Bit Extension Receive Operation



Data Length Configuration

When the Data Length Enable bit field in the Length register (LENGTH.LENEN) is written to 0x1 or 0x2, the Data Length bit (LENGTH.LEN) determines the number of characters to be transmitted or received from 1 to 255.

Note: There is one internal length counter that can be used for either transmit (LENGTH.LENEN = 0x1) or receive (LENGTH.LENEN = 0x2) but not for both simultaneously.

The LENGTH register must be written before the frame begins. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC/DRE interrupt is raised when the last byte is received/sent. The internal length counter is reset when LENGTH.LEN is reached or when LENGTH.LENEN is written to 0x0.

Writing the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 bytes, the remainder may be lost. Attempting to use the length counter for transmit and receive at the same time produces unpredictable results.

32.6.4 DMA, Interrupts and Events

Table 32-4. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

Table 32-5. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Standard (DRE): Data Register Empty FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data is written)	Yes	NA
Standard (RXC): Receive Complete FIFO (RXC): at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed.	Yes (request cleared when data is read)	Yes	
Standard (TXC): Transmit Complete FIFO (TXC): Transmit Complete and TX FIFO is empty	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

32.6.4.1 DMA Operation

The USART generates the following DMA requests:

- Data received (RX) – The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

32.6.4.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). 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 if the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled or the USART is reset. For details on clearing Interrupt flags, see *INTFLAG* from Related Links.

The value of INTFLAG indicates which interrupt is executed.
Note: Interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)
32.8.8. INTFLAG

32.6.4.3 Events

Not applicable.

32.6.5 Sleep Mode Operation

The behavior in Sleep mode is dependent on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY = 1 – GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY = 1 – The Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY = 0 – The internal clock is disabled, after any ongoing transfer is complete. The Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY = 0 – The external clock is disconnected, after any ongoing transfer is complete. All reception is dropped.

32.6.6 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 the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See the *CTRLB* register from Related Links.

Required write synchronization is denoted by the Write-Synchronized property in the register description.

Related Links

32.8.2. CTRLB

32.7 Register Summary

See the *SERCOM0/SERCOM1/SERCOM2* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8	SAMPR[2:0]					RXINV	TXINV	IBON
		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
		15:8			PMODE			ENC	SFDE	COLDEN
		23:16							RXEN	TXEN
		31:24							LINCMD[1:0]	
0x08	CTRLC	7:0						GTIME[2:0]		
		15:8					HDRDLY[1:0]		BRKLEN[1:0]	
		23:16			MAXITER[2:0]				DSNACK	INACK
		31:24							DATA32B[1:0]	
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
		15:8								
0x1C	SYNCBUSY	7:0				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20	RXERRCNT	7:0	RXERRCNT[7:0]							
0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8							LENEN[1:0]	
0x24 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	7:0				TXSPACE[4:0]				
		15:8				RXSPACE[4:0]				
0x36	FIFOPTR	7:0					CPUWRPTR[3:0]			
		15:8					CPURDPTR[3:0]			

Related Links

8. Product Memory Mapping Overview

32.8 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 require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

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

32.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]					RXINV	TXINV	IBON
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
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity
This bit selects the relationship between data output change and data input sampling in the Synchronous mode.
This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode
This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format
These bits define the frame format.
These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2	LIN Host - Break and sync generation. See LIN Command (CTRLB.LINCMD).
0x3	Reserved
0x4	Auto-baud (LIN Client) - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6	Reserved
0x7	ISO 7816
0x8-0xF	Reserved

Bits 23:22 – SAMPA[1:0] Sample Adjustment
These bits define the sample adjustment.
These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout
These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0] Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	NA	NA
0x1	Reserved			
0x2	SERCOM PAD[0]	NA	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	NA

Bits 15:13 – SAMPR[2:0] Sample Rate
These bits select the sample rate.
These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 10 – RXINV Receive Data Invert
This bit controls whether the receive data (RxD) is inverted or not.

Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

This bit is not synchronized.

Value	Description
0	RxD is not inverted.
1	RxD is inverted.

Bit 9 – TXINV Transmit Data Invert

This bit controls whether the transmit data (TxD) is inverted or not.

Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

This bit is not synchronized.

Value	Description
0	TxD is not inverted.
1	TxD is inverted.

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in Standby Sleep mode.

This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device will not wake up on Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain.
0x1	Wake on Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits select the USART serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state and the SERCOM is disabled.

Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

32.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
							LINCMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bits 25:24 – LINCMD[1:0] LIN Command

These bits define the LIN header transmission control. This field is only valid in LIN Host mode (CTRLA.FORM = LIN Host).
These are strobe bits and always reads back as '0'.
These bits are not enable-protected.

Value	Description
0x0	Normal USART transmission.
0x1	Break field is transmitted when DATA is written.
0x2	Break, sync and identifier are automatically transmitted when DATA is written with the identifier.
0x3	Reserved

Bit 17 – RXEN Receiver Enable

Writing '0' to this bit disables the USART receiver. Disabling the receiver flushes the receive buffer and clears the FERR, PERR and BUFOVF bits in the STATUS register.
Writing '1' to CTRLB.RXEN when the USART is disabled sets CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN is cleared and SYNCBUSY.CTRLB is set and remains set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN reads back as '1'.
Writing '1' to CTRLB.RXEN when the USART is enabled sets SYNCBUSY.CTRLB, which remains set until the receiver is enabled and CTRLB.RXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or is enabled when the USART is enabled.

Bit 16 – TXEN Transmitter Enable

Writing '0' to this bit disables the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.
Writing '1' to CTRLB.TXEN when the USART is disabled sets CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN is cleared and SYNCBUSY.CTRLB is set and remains set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN reads back as '1'.
Writing '1' to CTRLB.TXEN when the USART is enabled sets SYNCBUSY.CTRLB, which remains set until the transmitter is enabled and CTRLB.TXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or is enabled when the USART is enabled.

Bit 13 – PMODE Parity Mode

This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter automatically generates and sends the parity of the transmitted data bits within each frame. The receiver generates a parity value for the incoming data and parity bit, compares it to the Parity mode and, if a mismatch is detected, sets STATUS.PERR.
This bit is not synchronized.

Value	Description
0	Even parity
1	Odd parity

Bit 10 – ENC Encoding Format

This bit selects the data encoding format.
This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE Start of Frame Detection Enable

This bit controls whether the start-of-frame detector wakes up the device when a start bit is detected on the RxD line.
This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all Sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all Sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all Sleep modes.

Bit 8 – COLDEN Collision Detection Enable

This bit enables collision detection.
This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode

This bit selects the number of stop bits transmitted.
This bit is not synchronized.

Value	Description
0	One stop bit
1	Two stop bits

Bits 2:0 – CHSIZE[2:0] Character Size
These bits select the number of bits in a character.
These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

32.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
							DATA32B[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
		MAXITER[2:0]					DSNACK	INACK
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
					HDRDLY[1:0]		BRKLEN[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bits 25:24 – DATA32B[1:0] Data 32-Bit

These bits configure the 32-bit extension for read and write transactions to the DATA register. When disabled, access is according to CTRLB.CHSIZE.

Value	Description
0x0	DATA reads (for received data) and writes (for transmit data) according to CTRLB.CHSIZE.
0x1	DATA reads according to CTRLB.CHSIZE. DATA writes using 32-bit extension.
0x2	DATA reads using 32-bit extension. DATA writes according to CTRLB.CHSIZE.
0x3	DATA reads and writes using 32-bit extension.

Bits 22:20 – MAXITER[2:0] Maximum Iterations

These bits define the maximum number of retransmit iterations.
These bits also define the successive NACKs sent to the remote transmitter when CTRLC.DSNACK is set.
This field is only valid when using ISO7816 T = 0 mode (CTRLA.MODE = 0x7 and CTRLA.CMODE = 0).

Bit 17 – DSNACK Disable Successive Not Acknowledge

This bit controls how many times NACK is sent on parity error reception.
This bit is only valid in ISO7816 T = 0 mode and when CTRLC.INACK = 0.

Value	Description
0	NACK is sent on the ISO line for every parity error received.
1	Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line.

Bit 16 – INACK Inhibit Not Acknowledge

This bit controls whether a NACK is transmitted when a parity error is received.
This bit is only valid in ISO7816 T = 0 mode.

Value	Description
0	NACK is transmitted when a parity error is received.
1	NACK is not transmitted when a parity error is received.

Bits 11:10 – HDRDLY[1:0] LIN Host Header Delay

These bits define the delay between break and sync transmission, in addition to the delay between the sync and identifier (ID) fields when in LIN Host mode (CTRLA.FORM = 0x2).
This field is only valid when using the LIN header command (CTRLB.LINCMD = 0x2).

Value	Description
0x0	Delay between break and sync transmission is 1-bit time. Delay between sync and ID transmission is 1-bit time.
0x1	Delay between break and sync transmission is 4-bit time. Delay between sync and ID transmission is 4-bit time.
0x2	Delay between break and sync transmission is 8-bit time. Delay between sync and ID transmission is 8-bit time.
0x3	Delay between break and sync transmission is 14-bit time. Delay between sync and ID transmission is 14-bit time.

Bits 9:8 – BRKLEN[1:0] LIN Host Break Length

These bits define the length of the break field transmitted when in LIN Host mode (CTRLA.FORM = 0x2).

Value	Description
0x0	Break field transmission is 13-bit times
0x1	Break field transmission is 17-bit times
0x2	Break field transmission is 21-bit times
0x3	Break field transmission is 26-bit times

Bits 2:0 – GTIME[2:0] Guard Time

These bits define the guard time when using RS485 mode (CTRLA.FORM = 0x0 or CTRLA.FORM = 0x1 and CTRLA.TXPO = 0x3) or ISO7816 mode (CTRLA.FORM = 0x7).
For RS485 mode, the guard time is programmable from 0-7-bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.
For ISO7816 T = 0 mode, the guard time is programmable from 2-9-bit times and defines the guard time between each transmitted byte.

32.8.4 Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	BAUD[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
	BAUD[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 – BAUD[15:0] Baud Value

Arithmetic Baud Rate Generation (CTRLA.SAMPR[0] = 0).
These bits control the clock generation as described in the SERCOM Baud Rate section.
If Fractional Baud Rate Generation (CTRLA.SAMPR = 1 or = 3) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation; see *Clock Generation – Baud-Rate Generator* from Related Links.

- **Bits 12:0 - BAUD[12:0]: Baud Value**

  These bits control the clock generation; see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links

31.6.2.3. Clock Generation – Baud-Rate Generator

32.8.5 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RXPL[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 7:0 – RXPL[7:0] Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC = 1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.
$PULSE \geq (\mathrm{RXPL} + 1) \cdot SE_{per}$

32.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
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 are also reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
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 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Transmit Complete Interrupt Enable bit, which disables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
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 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R	R/W	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding status flags in the STATUS register. Errors that set this flag are COLL, ISF, BUFOVF, FERR and PERR. Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 5 – RXBRK Receive Break
This flag is cleared by writing '1' to it.
This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 4 – CTSIC Clear to Send Input Change
This flag is cleared by writing '1' to it.
This flag is set when a change is detected on the CTS pin.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 3 – RXS Receive Start
This flag is cleared by writing '1' to it.
This flag is set when a Start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Start Interrupt flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.
This flag is set when there are unread data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
This flag is set after the entire frame in the Transmit Shift register is shifted out and there are no new data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready to be written.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

32.8.9 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ITER Maximum Number of Repetitions Reached
This bit is set when the maximum number of NACK repetitions or retransmissions is met in the ISO7816 T = 0 mode.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 6 – TXE Transmitter Empty
When CTRLA.FORM is set to LIN Host mode, this bit is set when any ongoing transmission is complete and TxDATA is empty.
When CTRLA.FORM is not set to LIN Host mode, this bit will always read back as '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 5 – COLL Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 4 – ISF Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 3 – CTS Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading the Data register indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 1 – FERR Frame Error
Reading this bit before reading the Data register indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if the received character had a frame error, in other words, when the first stop bit is '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 0 – PERR Parity Error
Reading this bit before reading the Data register indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

32.8.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
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
				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy
Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH is set until synchronization is complete. If the LENGTH register is written while SYNCBUSY.LENGTH is asserted, an APB error is generated.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 3 – RXERRCNT Receive Error Count Synchronization Busy
The RXERRCNT register is automatically synchronized to the APB domain upon error. When returning from sleep, this bit is raised until the new value is available to be read.

Value	Description
0	RXERRCNT synchronization is not busy.
1	RXERRCNT synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB, the bit is set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error is generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When writing to CTRLA.ENABLE, the SYNCBUSY.ENABLE bit is set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When writing to CTRLA.SWRST, the SYNCBUSY.SWRST bit is set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

32.8.11 Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: Read-Synchronized

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

Bits 7:0 – RXERRCNT[7:0] Receive Error Count
This register records the total number of parity errors and NACK errors combined in ISO7816 mode (CTRLA.FORM = 0x7).
This register is automatically cleared on read.

32.8.12 Length

Name: LENGTH
Offset: 0x22
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
							LENEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LEN[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 9:8 – LENEN[1:0] Data Length Enable
In 32-bit Extension mode, this bit field configures the length counter either for transmit or receive transactions.

Value	Description
0x0	Length counter disabled
0x1	Length counter enabled for transmit
0x2	Length counter enabled for receive
0x3	Reserved

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode, this bit field configures the data length, after which, the flags INTFLAG.RXC or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN = 0x1 or LENEN = 0x2
0x01-0xFF	Data Length

32.8.13 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
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

Reading these bits returns the contents of the Receive Data register. The register must be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS must be read before reading the DATA value to get any corresponding error.

Writing these bits writes the Transmit Data register. This register must be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.

32.8.14 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

32.8.15 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: -

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:8 – RXSPACE[4:0] RX FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] TX FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

32.8.16 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: -

This register provides a copy of internal CPU TX and RX FIFO pointers.

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

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

Bits 11:8 – CPURDPTR[3:0] RX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] TX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writting to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

33. SERCOM Serial Peripheral Interface (SERCOM SPI)

33.1 Overview

The Serial Peripheral Interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as illustrated in 33.3. Block Diagram. Each side, host and client, depicts a separate SPI containing a Shift register, a transmit buffer and a two-level receive buffer. In addition, the SPI host uses the SERCOM baud-rate generator, while the SPI client can use the SERCOM address match logic. Labels in capital letters are synchronous to PB1_CLK and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

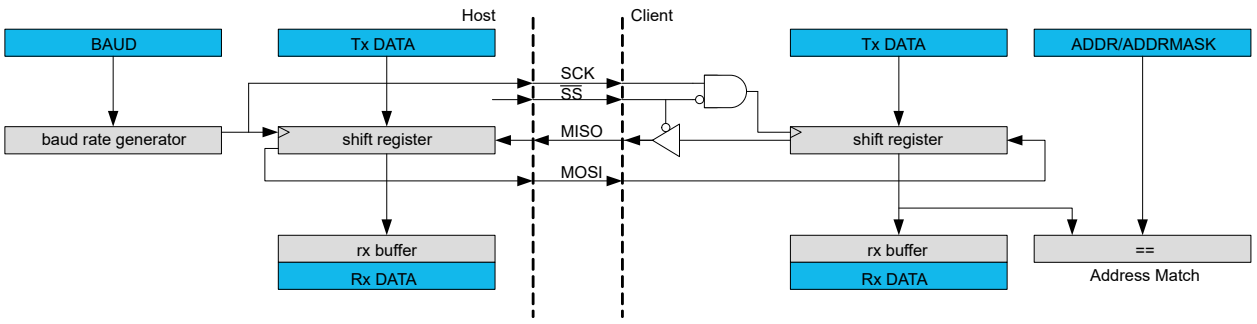
33.2 Features

- Full-Duplex, Four-Wire Interface (MISO, MOSI, SCK, $\overline{SS}$)
- One-Level Transmit Buffer, Two-Level Receive Buffer
- Supports All Four SPI Modes of Operation
- Single Data Direction Operation Allows Alternate Function on MISO or MOSI Pin
- Selectable LSB- or MSB- First Data Transfer
- Can Be Used with DMA
- 32-Bit Extension for Better System Bus Utilization
- Up to 16-Bytes Internal FIFO
- Host Operation:
  - Serial clock speed up to half the system clock
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
  - Optional inter-character spacing
- Client Operation:
  - Serial clock speed up to half the system clock
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

Note: SERCOM2 does not have SPI functionality.

33.3 Block Diagram

Figure 33-1. Full-Duplex SPI Host Client Interconnection



33.4 Signal Description

Table 33-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins. For more details on pin mapping, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

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

33.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

33.5.1 I/O Lines

To use the SERCOM's I/O lines, the I/O pins must be configured using the System Configuration registers (See *System Configuration and Register Locking (CFG)* from Related Links) (SCOM_HSEN[1:0] of CFGCON1/DEVCFG1 register) for direct or PPS configuration. If SERCOM pins are selected through PPS, the PPS registers have to be configured. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

If SCOMx_HSEN = 1, SERCOM uses dedicated pins.

If SCOMx_HSEN = 0, SERCOM uses PPS path and I/O pins are multiplexed to pins groups defined in PPS section.

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the following table. If the receiver is disabled, the data input pin can be used for other purposes. In Host mode, the SPI Select line ($\overline{SS}$) is hardware-controlled when the Host SPI Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 33-2. SPI Pin Configuration

Pin	Host SPI	Client SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN = 1)	Input

The configuration of the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the Table 33-2.

Related Links

6. I/O Ports and Peripheral Pin Select (PPS)
22. System Configuration and Register Locking (CFG)

33.5.2 Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

33.5.3 Clocks

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Clock and Reset Unit (CRU) and Configuration (CFG.CFGPCLKGEN1) registers before using the SPI.

This generic clock is asynchronous to the bus clock (PB1_CLK). Therefore, writes to certain registers require synchronization to the clock domains.

33.5.4 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 (DMAC)* from Related Links).

Related Links

26. Direct Memory Access Controller (DMAC)

33.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

33.5.6 Events

Not applicable.

33.5.7 Debug Operation

When the CPU is halted in the Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See *DBGCTRL* register from Related Links.

Related Links

33.8.13. DBGCTRL

33.5.8 Register Access Protection

Registers with write access can be write protected optionally by the PAC.

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

33.5.9 Analog Connections

Not applicable.

33.6 Functional Description

33.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral devices.

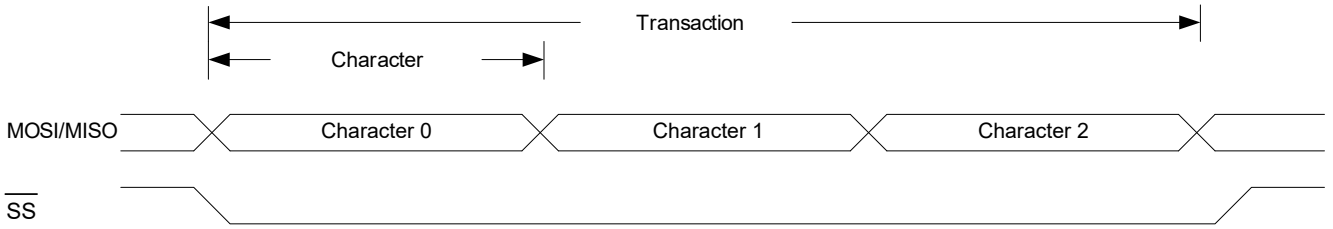
The SPI can operate as host or client. As host, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 33-2. SPI Transaction Format



The SPI host must pull the SPI Select line ($\overline{SS}$) of the desired client low to initiate a transaction if multiple clients are connected to the bus. The SPI Select line can be wired low if there is only one SPI client on the bus. The host and client prepare data to send via their respective Shift registers, and the host generates the serial clock on the SCK line.

Data is always shifted from host to client on the Host Output Client Input line (MOSI); data is shifted from client to host on the Host Input Client Output line (MISO).

Each time character is shifted out from the host, a character will be shifted out from the client simultaneously. To signal the end of a transaction, the host will pull the $\overline{SS}$ line high.

33.6.2 Basic Operation

33.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE = 0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE = 1), any writing to these registers is discarded.

When the SPI is being disabled, writing to these registers is completed after the disabling.

Enable protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in host/client operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE = 0x2 or 0x3).
2. Select Transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.

6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in Host mode:
   a. Select the desired baud rate by writing to the Baud register (BAUD).
   b. If Hardware $\overline{SS}$ control is required, write '1' to the Host SPI Select Enable bit in the CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN = 1).

33.6.2.2 Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled. See *CTRLA* register from Related Links.

Related Links

33.8.1. CTRLA

33.6.2.3 Clock Generation

In the SPI host operation (CTRLA.MODE = 0x3), the serial clock (SCK) is generated internally by the SERCOM Baud Rate Generator (BRG).

In the SPI mode, the BRG is set to Synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the Shift register (see *Clock Generation – Baud-Rate Generator* from Related Links).

In the SPI client operation (CTRLA.MODE = 0x2), the clock is provided by an external host on the SCK pin. This clock is used to clock the SPI Shift register.

Related Links

31.6.2.3. Clock Generation – Baud-Rate Generator

33.6.2.4 Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

33.6.2.5 SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in the following table and figure.

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

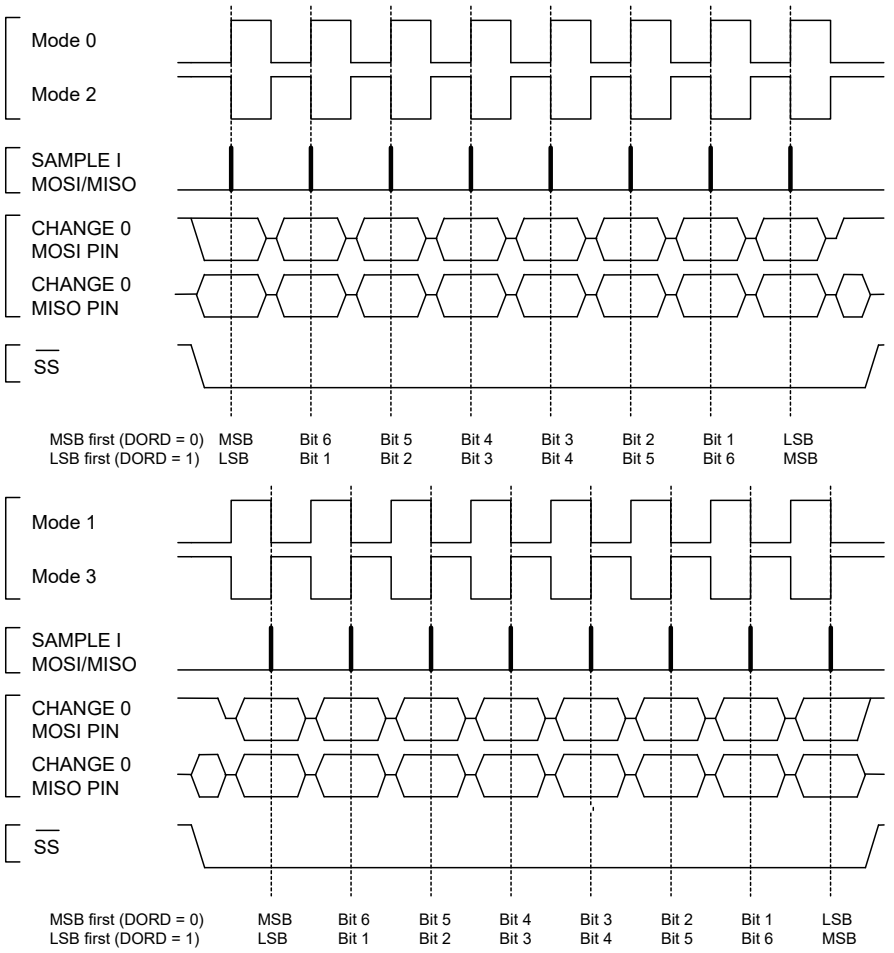
Table 33-3. SPI Transfer Modes

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Notes:

- The leading edge is the first clock edge in a clock cycle.
- The trailing edge is the second clock edge in a clock cycle.

Figure 33-3. SPI Transfer Modes



33.6.2.6 Transferring Data

33.6.2.6.1 Host

In Host mode (CTRLA.MODE = 0x3), when Host SPI Select Enable (CTRLB.MSSEN) is '1', hardware controls the $\overline{SS}$ line.

When Host SPI Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character is transferred to the Shift register when the shift register is empty. After the content of TxDATA is transferred to the Shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set. And a new character can be written to DATA.

Each time one character is shifted out from the host, another character is shifted in from the client simultaneously. If the receiver is enabled (CTRLA.RXEN = 1), the contents of the Shift register is transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The received data can be retrieved by reading DATA.

After the last character is transmitted and there are no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) is set. When the transaction is finished, the host must pull the $\overline{SS}$ line high to notify the client. If Host SPI Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

33.6.2.6.2 Client

In Client mode (CTRLA.MODE = 0x2), the SPI interface remains inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the client will sample and shift out data according to the Transaction mode set. After the content of TxDATA is loaded into the Shift register, INTFLAG.DRE is set and new data can be written to DATA.

Similar to the host, the client receives one character for each character transmitted. A character is transferred into the two-level receive buffer within the same clock cycle its last data bit received. The received character can be retrieved from DATA when the Receive Complete Interrupt flag (INTFLAG.RXC) is set.

When the host pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) is set.

After DATA is written, it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the Shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. See *Preloading of the Client Shift Register* from Related Links.

When transmitting several characters in one SPI transaction, the data have to be written into the DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character is transmitted.

When the DATA register is empty, it takes three PB1_CLK cycles for INTFLAG.DRE to be set.

Related Links

33.6.3.2. Preloading of the Client Shift Register

33.6.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). When an error happens, the bit stays set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

- If CTRLA.IBON = 1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can, then, empty the receive FIFO by reading RxDATA until the receiver complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.
- If CTRLA.IBON = 0, the Buffer Overflow condition travels with data through the receive FIFO. After the received data are read, STATUS.BUFOVF and INTFLAG.ERROR are set along with INTFLAG.RXC, and RxDATA is '0'.

33.6.3 Additional Features

33.6.3.1 Address Recognition

When the SPI is configured for client operation (CTRLA.MODE = 0x2) with address recognition (CTRLA.FORM = 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled and the transaction is processed. If the device is in Sleep mode, an address match can wake up the device to process the transaction.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the Shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN = 0) to use this mode.

33.6.3.2 Preloading of the Client Shift Register

When starting a transaction, the client will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

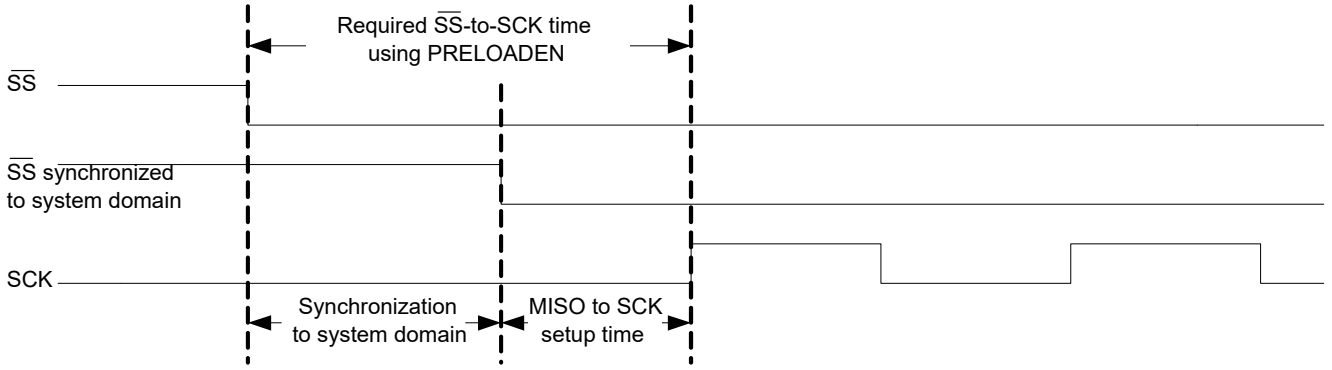
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as shown in the following figure. For timing details, see *Electrical Characteristics* from Related Links.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 33-4. Timing Using Preloading



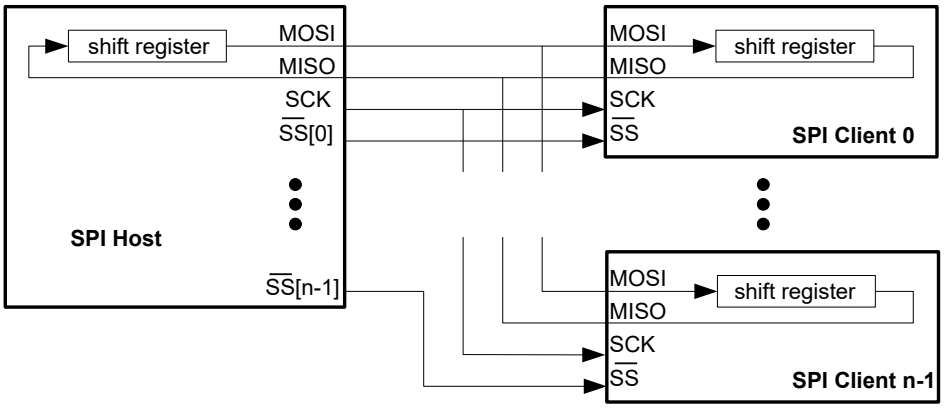
Related Links

43. Electrical Characteristics

33.6.3.3 Host with Several Clients

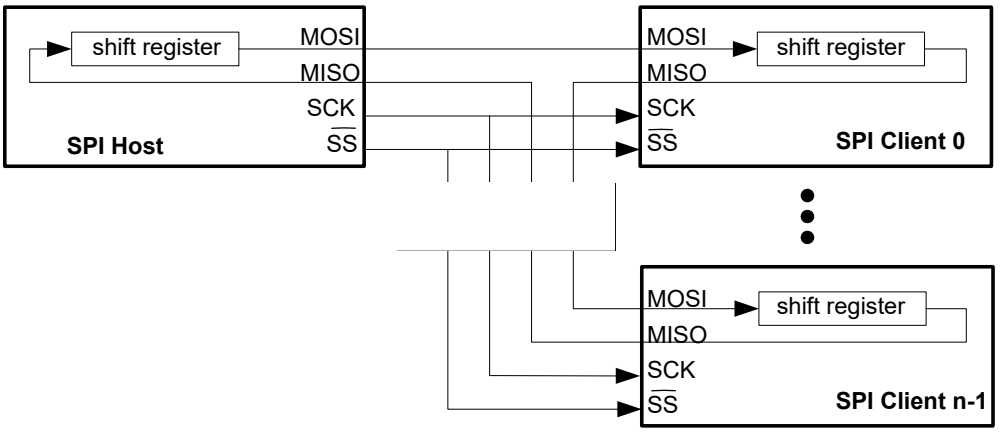
Host with multiple clients in parallel is only available when Host SPI Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI clients, a SPI host can use general purpose I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in the following figure. In this configuration, the single selected SPI client will drive the tri-state MISO line.

Figure 33-5. Multiple Clients in Parallel



Another configuration is multiple clients in series, as shown in the following figure. In this configuration, all n attached clients are connected in series. A common $\overline{SS}$ line is provided to all clients, enabling them simultaneously. The host must shift n characters for a complete transaction. Depending on the Host SPI Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO. The $\overline{SS}$ line is controlled by a normal GPIO.

Figure 33-6. Multiple Clients in Series



33.6.3.4 Loop-Back Mode

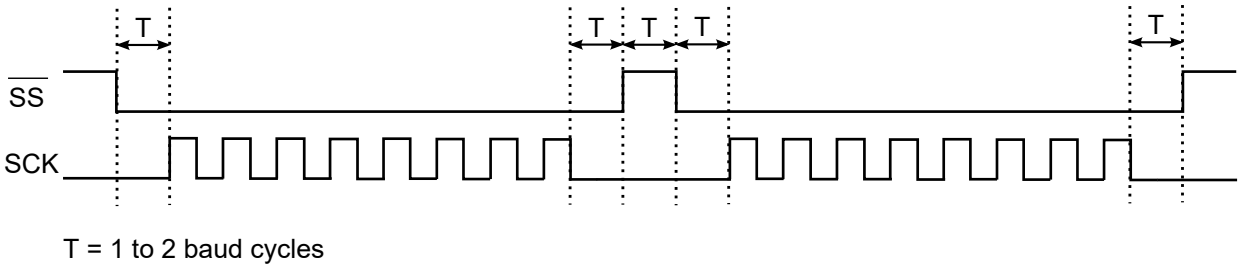
For Loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

33.6.3.5 Hardware Controlled $\overline{SS}$

In Host mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Host SPI Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In the following figure, the time, T, is between one and two baud cycles depending on the SPI Transfer mode.

Figure 33-7. Hardware Controlled $\overline{SS}$



When CTRLB.MSSEN = 0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

33.6.3.6 SPI Select Low Detection

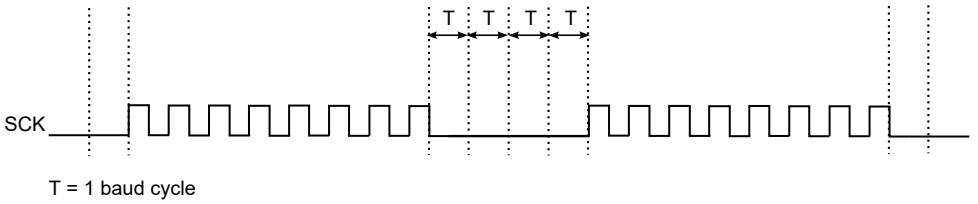
In Client mode, the SPI can wake the CPU when the SPI Select ($\overline{SS}$) goes low. When the SPI Select Low Detect is enabled (CTRLB.SSDE = 1), a high-to-low transition sets the SPI Select Low Interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

33.6.3.7 Host Inter-Character Spacing

When configured as the host, inter-character spacing can be increased by writing a non-zero value to the Inter-Character Spacing bit field in the Control C register (CTRLC.ICSPACE). When non-zero, CTRLC.ICSPACE represents the minimum number of baud cycles that the SCK clock line does not toggle and the next character is stalled.

The following figure illustrates an example for CTRLC.ICSPACE = 4; in this case, the SCK is inactive for four baud cycles.

Figure 33-8. Four Cycle Inter-Character Spacing Example



33.6.3.8 32-Bit Extension

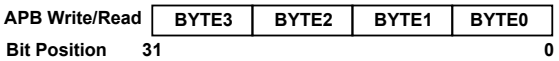
For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B = 1). When enabled, write and read transactions to/from the DATA register are 32 bits in size.

If frames are not multiples of 4 bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

Only 8-bit character size is supported.

Figure 33-9. 32-Bit Extension Byte Ordering

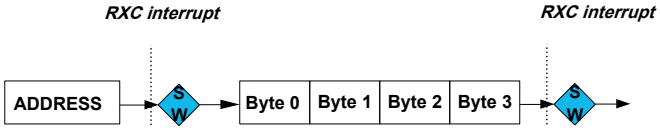


32-Bit Extension Client Operation

The following figure illustrates a transaction with 32-bit Extension enabled (CTRLC.DATA32B = 1). When address recognition is enabled (CTRLA.FORM = 0x2) and there is an address match, the

address is loaded into the FIFO as Byte zero and data begins with Byte 1. INTFLAGS.RXC will, then, be raised for every 4 bytes transferred. For transmit, there is a 32-bit holding buffer in the core domain. After DATA is registered in the core domain, INTFLAG.DRE is raised so that the next 32 bits can be written to the DATA register.

Figure 33-10. 32-Bit Extension Client Operation



When utilizing the length counter, the LENGTH register must be written before the frame begins. If the frame length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN bytes, the Length Error Status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC interrupt is raised when the last byte is received.

The length count is based on the received bytes or the number of clocks if the receiver is not enabled. If pre-loading is disabled and DATA is written for transmit before SCK starts, transmitted data are delayed by one byte, but the length counter will still increment for the first (empty) byte transmission. When the counter reaches LENGTH.LEN, the internal length counter, RX byte counter and TX byte counter are reset. If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

If there is a Length Error (STATUS.LENERR), the remaining bytes in the length are transmitted at the beginning of the next frame. If this is not desired, the SERCOM must be disabled and re-enabled to flush the TX and RX pipelines.

Writing the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not configured and a frame is not a multiple of 4 bytes (while $\overline{SS}$ is low), the remainder is transmitted in the next frame.

32-Bit Extension Host Operation

When using the SPI configured as the host, the Length and the Length Enable bit fields (LENGTH.LEN and LENGTH.LENEN) must be written before the frame begins. When LENGTH.LENEN is written to '1', the value of LENGTH.LEN determines the number of data bytes in the transaction from 1 to 255.

For receive data, INTFLAG.RXC is raised every 4 bytes received. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC is set when the final byte is received.

For transmit, there is a holding buffer for the 32-bit data in the core domain. After DATA is registered in the SCK domain, INTFLAG.DRE is raised so that the next 32 bits can be written to the DATA register.

If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

33.6.4 DMA, Interrupts and Events

Table 33-4. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data are written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data are read)	Yes	
Transmit Complete (TXC)	NA	Yes	
SPI Select Low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

Table 33-5. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Standard (DRE) – Data Register Empty FIFO (DRE) – at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data are written)	Yes	NA
Standard (RXC) – Receive Complete FIFO (RXC) – at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed.	Yes (request cleared when data are read)	Yes	
Standard (TXC) – Transmit Complete FIFO (TXC) – Transmit Complete and TX FIFO is empty	NA	Yes	
SPI Select Low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

33.6.4.1 DMA Operation

The SPI generates the following DMA requests:

- Data received (RX) – The request is set when data are available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.
- Data received (RX) – The request is set when data are available in the receive FIFO or if at least RXTRHOLD data are available in the RX FIFO when FIFO operation is enabled. The request is cleared when DATA is read.
- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty or if at least TXTRHOLD data locations are empty in the TX FIFO, when FIFO operation is enabled. The request is cleared when DATA is written.

33.6.4.2 Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake-up the device from any Sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- SPI Select Low (SSL)
- Error (ERROR)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). 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 if the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing Interrupt flags, see INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

33.8.7. INTFLAG

33.6.4.3 Events

Not applicable.

33.6.5 Sleep Mode Operation

The behavior in Sleep mode depends on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Host operation, CTRLA.RUNSTDBY = 1 – The peripheral clock GCLK_SERCOMx_CORE continues to run in Idle Sleep mode and in Standby Sleep mode. Any interrupt can wake up the device.
- Host operation, CTRLA.RUNSTDBY = 0 – GCLK_SERCOMx_CORE is disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY = 1 – The Receive Complete interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY = 0 – All reception is dropped, including the ongoing transaction.

33.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. When executing an operation that requires synchronization, the corresponding Synchronization Busy bit in the Synchronization register (SYNCBUSY) is set immediately and cleared when synchronization is complete. If an operation that requires synchronization is executed while the corresponding SYNCBUSY bit is one, a peripheral bus error is generated.

The following bits need to be synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See *CTRLB* register from Related Links.

Required write synchronization is denoted by the Write-Synchronized property in the register description.

Related Links

33.8.2. CTRLB

33.7 Register Summary

See the *SERCOM0/SERCOM1/SERCOM2* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								IBON
		23:16			DIPO[1:0]				DOPO[1:0]	
		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
		15:8	AMODE[1:0]		MSSEN				SSDE	
		23:16	FIFOCLR[1:0]						RXEN	
		31:24								
0x08	CTRLC	7:0			ICSPACE[5:0]					
		15:8								
		23:16								
		31:24								DATA32B
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
		15:8					LENERR			
0x1C	SYNCBUSY	7:0				LENGTH		CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8								LENEN
0x24	ADDR	7:0	ADDR[7:0]							
		15:8								
		23:16	ADDRMASK[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Related Links

8. Product Memory Mapping Overview

33.8 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 require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

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

33.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

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

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the shift register.
This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data are sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data are sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0] Frame Format

This bit field selects the various frame formats supported by the SPI in Client mode. When the SPI frame with address format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	—	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	—	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout

These bits define the data in (DI) pad configurations.
In host operation, DI is MISO.
In client operation, DI is MOSI.
These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout

These bits define the available pad configurations for Data Out (DO), the Serial Clock (SCK) and the SPI Select ($\overline{SS}$). In Client operation, the SPI Select line ($\overline{SS}$) is controlled by DOPO. In host operation, the SPI Select line ($\overline{SS}$) is either controlled by DOPO when CTRLB.MSSEN = 1, or by a GPIO driven by the application when CTRLB.MSSEN = 0.
In host operation, DO is MOSI.
In client operation, DO is MISO.
These bits are not synchronized.

DOPO	DO	SCK	Client $\overline{SS}$	Host $\overline{SS}$ (MSSEN = 1)	Host $\overline{SS}$ (MSSEN = 0)
0x0	PAD[0]	PAD[1]	PAD[2]	PAD[2]	Any GPIO configured by the application
0x1	Reserved				
0x2	PAD[3]	PAD[1]	PAD[2]	PAD[2]	Any GPIO configured by the application
0x3	Reserved				

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.
This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in Standby Sleep mode.
This bit is not synchronized.

RUNSTDBY	Client	Host
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in Sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.
0x2 – SPI client operation
0x3 – SPI host operation
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state and the SERCOM is disabled.
Writing '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write operation is discarded. Any register write access during the ongoing Reset results in an APB error. Reading any register returns the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

33.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]						RXEN	
Access	R/W	R/W					R/W	
Reset	0	0					0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bits 23:22 – FIFOCLR[1:0] FIFO Clear
When these bits are set, the corresponding FIFO is cleared. The bits automatically clears when SYNCBUSY.CTRLB = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit disables the SPI receiver immediately. The receive buffer is flushed, data from ongoing receptions are lost and STATUS.BUFOVF is cleared.
Writing '1' to CTRLB.RXEN when the SPI is disabled sets CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN is cleared, SYNCBUSY.CTRLB is set and remains set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN reads back as '1'.
Writing '1' to CTRLB.RXEN when the SPI is enabled sets SYNCBUSY.CTRLB, which remains set until the receiver is enabled, and CTRLB.RXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled.
1	The receiver is enabled or it is enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the Client Addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in Host mode.

These bits are not synchronized.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The client responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	—	Reserved

Bit 13 – MSSEN Host SPI Select Enable

This bit enables hardware SPI Select ($\overline{SS}$) control.
This bit is not synchronized.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE SPI Select Low Detect Enable

This bit enables wake-up when the SPI Select ($\overline{SS}$) pin transitions from high-to-low.
This bit is not synchronized.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN Client Data Preload Enable

Setting this bit enables preloading of the Client Shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it is transferred immediately to the Shift register.
This bit is not synchronized.

Bits 2:0 – CHSIZE[2:0] Character Size

These bits are not synchronized.

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	—	Reserved

33.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
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
			ICSPACE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 24 – DATA32B Data 32 Bit
This bit enables the 32-bit extension for read and write transactions to the DATA register.
When disabled, access correlates to CTRLB.CHSIZE.

Value	Description
0	Transactions from and to the DATA register correlate to CTRLB.CHSIZE
1	Transactions from and to the DATA register are 32-bit

Bits 5:0 – ICSPACE[5:0] Inter-Character Spacing
When non-zero, CTRLC.ICSPACE selects the minimum number of baud cycles the SCK line will not toggle between characters.

Value	Description
0x00	Inter-Character Spacing is disabled
0x01-0x3F	The minimum Inter-Character Spacing

33.8.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	BAUD[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 7:0 – BAUD[7:0] Baud Register
These bits control the clock generation, see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links
31.6.2.3. Clock Generation – Baud-Rate Generator

33.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the SPI Select Low Interrupt Enable bit, which disables the SPI Select Low interrupt.

Value	Description
0	SPI Select Low interrupt is disabled.
1	SPI Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Transmit Complete Interrupt Enable bit, which disables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the SPI Select Low Interrupt Enable bit, which enables the SPI Select Low interrupt.

Value	Description
0	SPI Select Low interrupt is disabled.
1	SPI Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding Status flags in the STATUS register. The BUFOVF error and the LENERR error set this Interrupt flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 3 – SSL SPI Select Low
This flag is cleared by writing '1' to it.
This bit is set when a high-to-low transition is detected on the $\overline{SS}$ pin in Client mode and SPI Select Low Detect (CTRLB.SSDE) is enabled.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.
This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction are an address.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
In Host mode, this flag is set when the data are shifted out and there are no new data in DATA.
In Client mode, this flag is set when the $\overline{SS}$ pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready for new data to transmit.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

33.8.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –

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

Bit	7	6	5	4	3	2	1	0
						BUFOVF		
Access						R/W		
Reset						0		

Bit 11 – LENERR Transaction Length Error
This bit is set in the Client mode when the length counter is enabled (LENGTH.LENEN = 1) and the transfer length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Value	Description
0	No Length Error has occurred.
1	A Length Error has occurred.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading DATA indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when a Buffer Overflow condition is detected.
When set, the corresponding RxDATA is '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

33.8.9 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
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
				LENGTH		CTRLB	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy
Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH is set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, an APB error is generated.
Note: In the Client mode, the clock is only running during data transfer, therefore, SYNCBUSY.LENGTH remains asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB = 1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB = 1, an APB error is generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE = 1 until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST = 1 until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

33.8.10 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

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

Bit	7	6	5	4	3	2	1	0
	LEN[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 8 – LENEN Data Length Enable
In 32-bit Extension mode, this bit field enables the length counter.

Value	Description
0	Length counter disabled
1	Length counter enabled

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode, this bit field configures the data length, after which, the flags INTFLAG.RCX or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN = 0x1
0x01-0xFF	Data Length

33.8.11 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[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	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR[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 23:16 – ADDRMASK[7:0] Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

33.8.12 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
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

Reading these bits returns the contents of the receive data buffer. The register must be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits writes the transmit data buffer. This register must be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set. Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.

33.8.13 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34. SERCOM Inter-Integrated Circuit (SERCOM I^{2}C)

34.1 Overview

The Inter-Integrated Circuit (I^{2}C) interface is one of the available modes in the serial communication interface (SERCOM).

The I^{2}C interface uses the SERCOM transmitter and receiver configured as illustrated in Figure 34-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

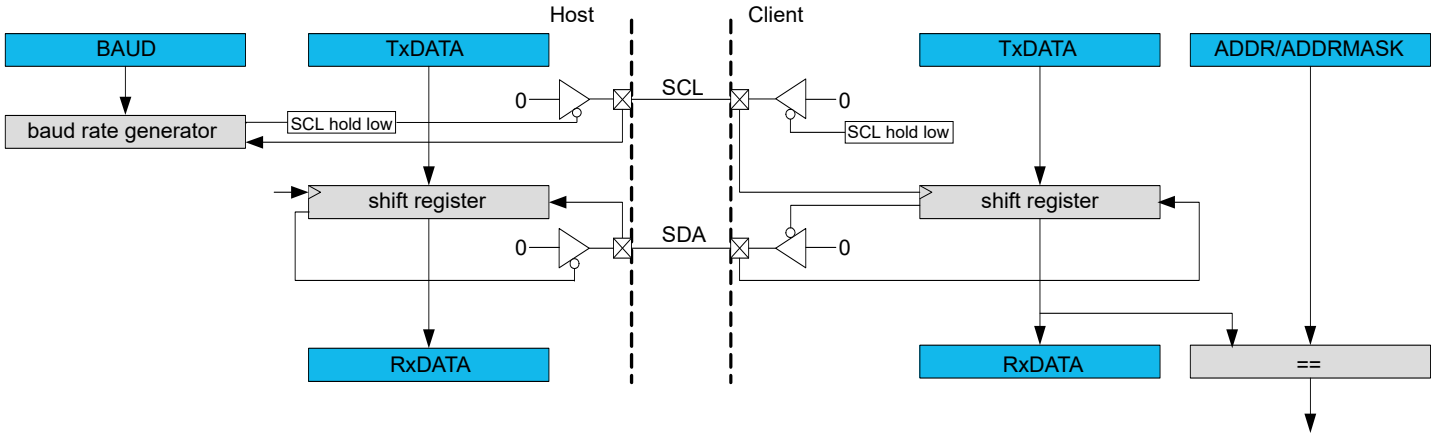
A SERCOM instance can be configured to be either an I^{2}C host or an I^{2}C client. Both host and client have an interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I^{2}C host uses the SERCOM baud-rate generator, while the I^{2}C client uses the SERCOM address match logic.

34.2 Features

- Host or Client Operation
- Can be used with DMA
- Philips I^{2}C Compatible
- SMBus Compatible
- PMBus™ Compatible
- Support 100 kHz, 400 kHz and 1 MHz at Low System Clock Frequencies
- 32-Bit Data Extension for better System Bus Utilization
- Up to 16-Bytes Internal FIFO
- Four-Wire Operation Supported
- Physical Interface Includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Client Operation:
  - Operation in all Sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
    - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

34.3 Block Diagram

Figure 34-1. I²C Single-Host Single-Client Interconnection



34.4 Signal Description

Table 34-1. Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (Four-wire operation)
PAD[3]	Digital I/O	SCL_OUT (Four-wire operation)

I²C pins on SERCOM are fixed pins and not configurable through PPS functionality. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

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

34.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

34.5.1 I/O Lines

To use the SERCOM's I/O lines, the I/O pins must be configured as direct using the System Configuration registers. (See *System Configuration and Register Locking (CFG)* from Related Links.) I²C does not operate through PPS. (See the DEVCFG1 configuration bits SCOMn_HSEN in Configuration Bits Fuses and also the CFGCON1 SCOMn_HSEN in CFGCON1(L) register.)

Note: SERCOM2 has only I²C functionality and, therefore, the direct configuration using System configuration register is not required.

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. In I²C mode, pull-up resistors are disabled. External pull-up resistors are required for proper function.

Related Links

22. System Configuration and Register Locking (CFG)

34.5.2 Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

34.5.3 Clocks

Two generic clocks are used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a host. The slow clock (GCLK_SERCOMx_SLOW) is required only for certain functions, for example, for SMBus timing 32KHz_LPCLK must be configured as this is the source for GCLK_SERCOMx_SLOW clock. These two clocks must be configured and enabled in the CRU registers before using the I^2C.

These generic clocks are asynchronous to the bus clock (PBx_CLK). Due to this asynchronicity, writes to certain registers requires synchronization between the clock domains.

34.5.4 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 (DMAC)* from Related Links).

Related Links

26. Direct Memory Access Controller (DMAC)

34.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

34.5.6 Events

Not applicable.

34.5.7 Debug Operation

When the CPU is halted in Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See the *DBGCTRL* register from Related Links.

Related Links

34.10.12. DBGCTRL

34.5.8 Register Access Protection

Registers with write access can be write protected optionally by the PAC.

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR) in Host mode

Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

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

34.5.9 Analog Connections

Not applicable.

34.6 Functional Description

34.6.1 Principle of Operation

The I^2C interface uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer
- Serial Clock Line (SCL) for the bus clock

A transaction starts with the I^2C host sending the Start condition, followed by a 7-bit address and a direction bit (read or write to/from the client).

The addressed I^2C client will then Acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is Not Acknowledged (NACK), whether by the I^2C client or host, the I^2C host takes action by either terminating the transaction by sending the Stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

Figure 34-2. Transaction Diagram Symbols

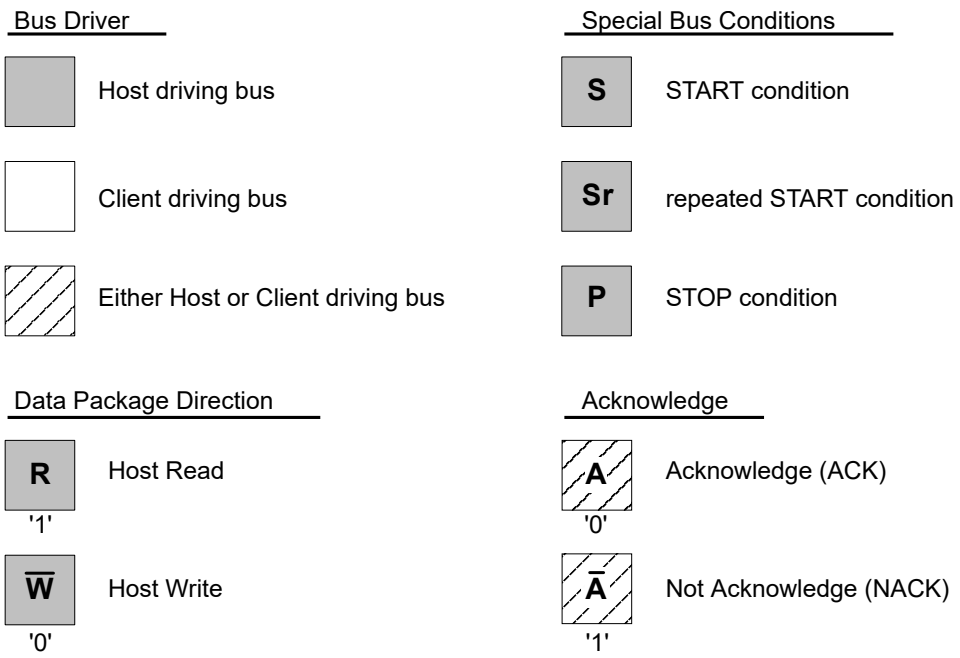
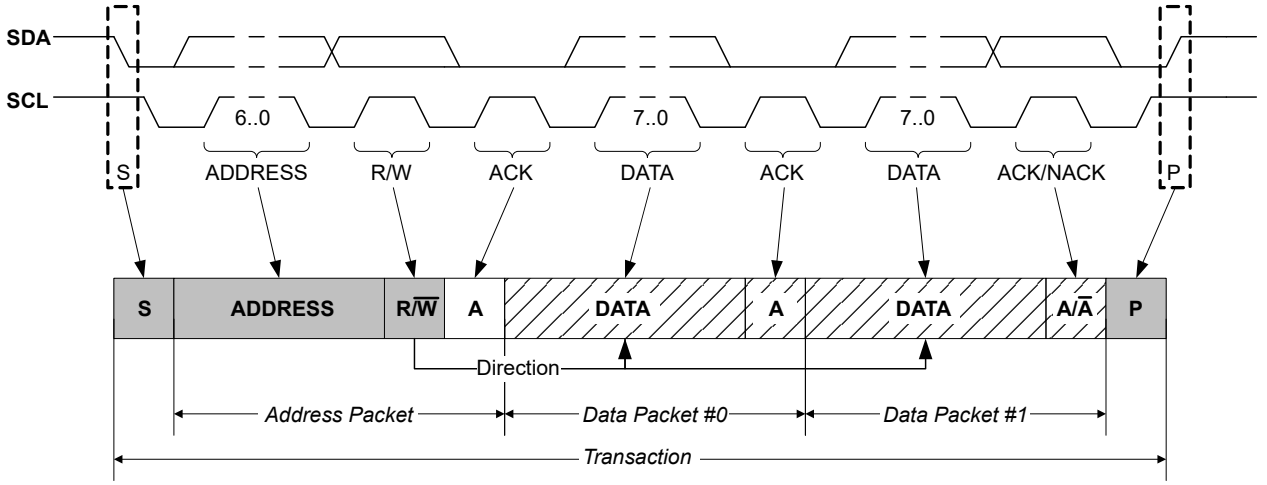


Figure 34-3. Basic I²C Transaction Diagram



34.6.2 Basic Operation

34.6.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE = 0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in client operation

When the I²C is enabled or is being enabled (CTRLA.ENABLE = 1), writing to these registers is discarded. If the I²C is being disabled, writing to these registers is completed after the disabling.

Enable protection is denoted by the Enable-Protection property in the register description.

Before the I²C is enabled, it must be configured as outlined by the following steps:

1. Select I²C Host or Client mode by writing 0x4 (Client mode) or 0x5 (Host mode) to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time in the CFGCON1 register (CFGCON1.I2CDSELx).
3. In Client mode, the minimum client setup time for the SDA can be selected in the SDA Setup Time bit group in the Control C register (CTRLC.SDASETUP).
4. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
5. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
6. In Host mode:
   a. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   b. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Client mode:
   a. Configure the address match configuration by writing the Address mode value in the CTRLB register (CTRLB.AMODE).

b. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

34.6.2.2 Enabling, Disabling and Resetting

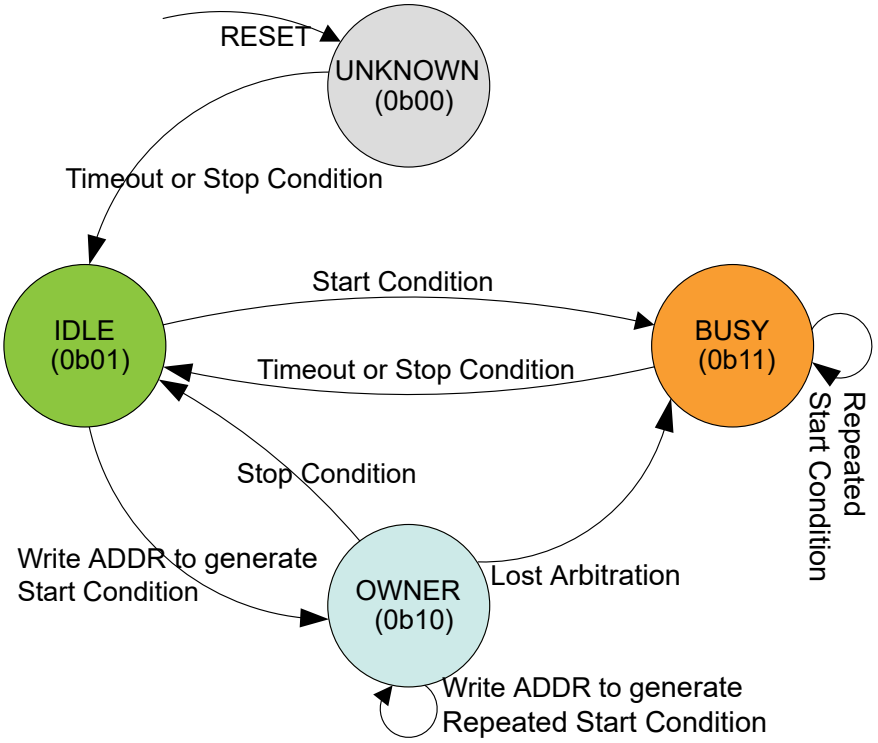
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

34.6.2.3 I²C Bus State Logic

The Bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all sleep modes with running GCLK_SERCOMx_CORE clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current Bus state. The following figure illustrates how the Bus state is determined. Software can get the current Bus state by reading the Host Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the following figure is shown in binary.

Figure 34-4. Bus State Diagram



The Bus state machine is active when the I²C host is enabled.

After the I²C host is enabled, the Bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by writing 0b01 to STATUS.BUSSTATE
- A Stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs

Note: When a known Bus state is established, the Bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE, it is ready for a new transaction. If a Start condition is issued on the bus by another I²C host in a multi-host setup, the bus becomes BUSY (0b11). The bus re-enters IDLE either when a Stop condition is detected or when a time-out occurs (inactive bus time-out needs to be configured).

If a Start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, in other words, arbitration was not lost, the I²C host can issue a Stop condition, which changes the Bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the Bus state becomes BUSY until a Stop condition is detected. A repeated Start condition changes the Bus state only if arbitration is lost while issuing a repeated start.

Note: Violating the protocol may cause the I²C to hang. If this happens, it is possible to recover from this state by a software Reset (CTRLA.SWRST = 1). See the *CTRLA* register from Related Links.

Related Links

34.10.1. CTRLA

34.6.2.4 I²C Host Operation

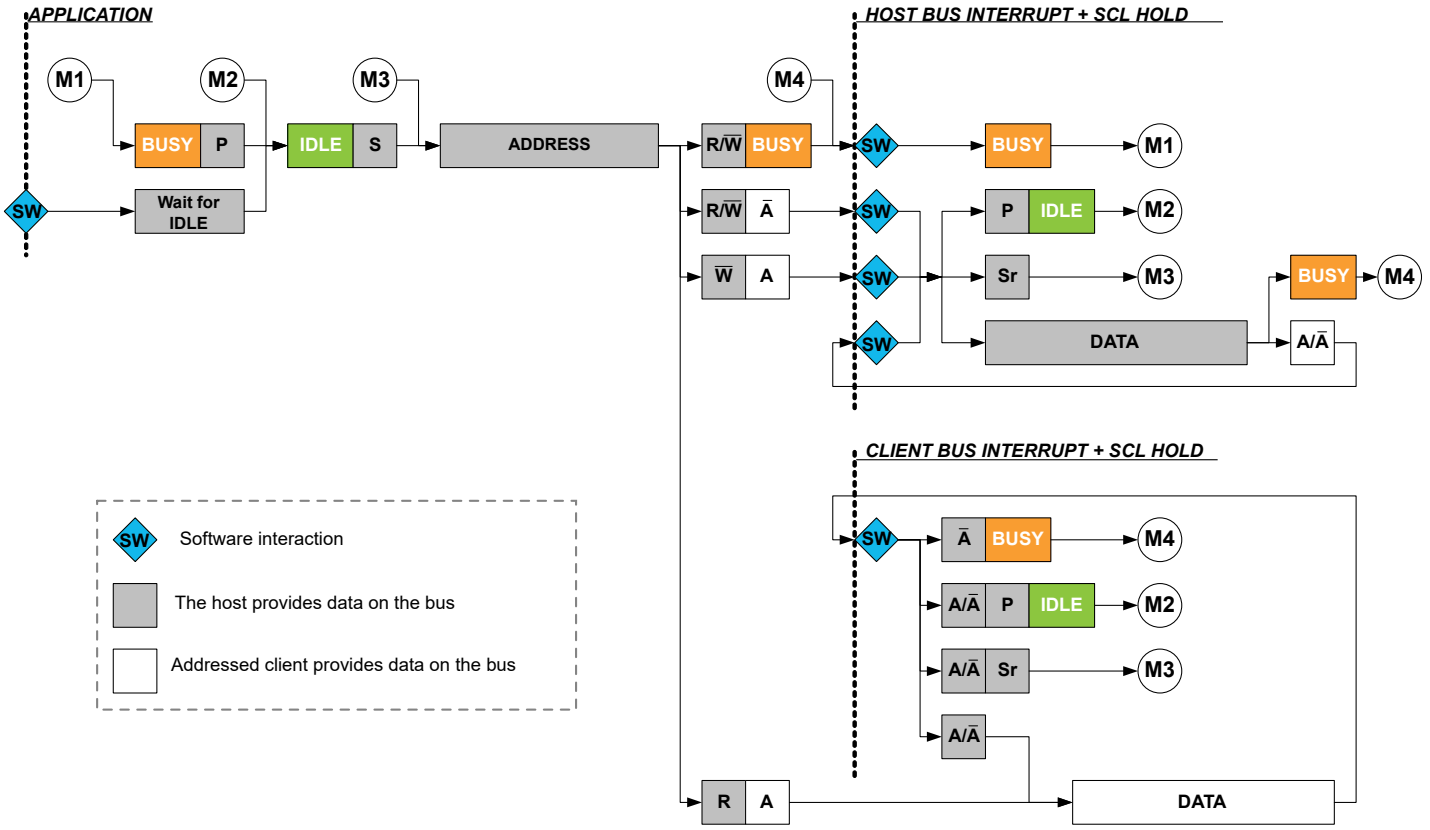
The I²C host is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations and a Special Smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C host has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the Acknowledge bit . In this mode, the I²C host operates according to Figure 34-5. The circles labeled M*n* (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

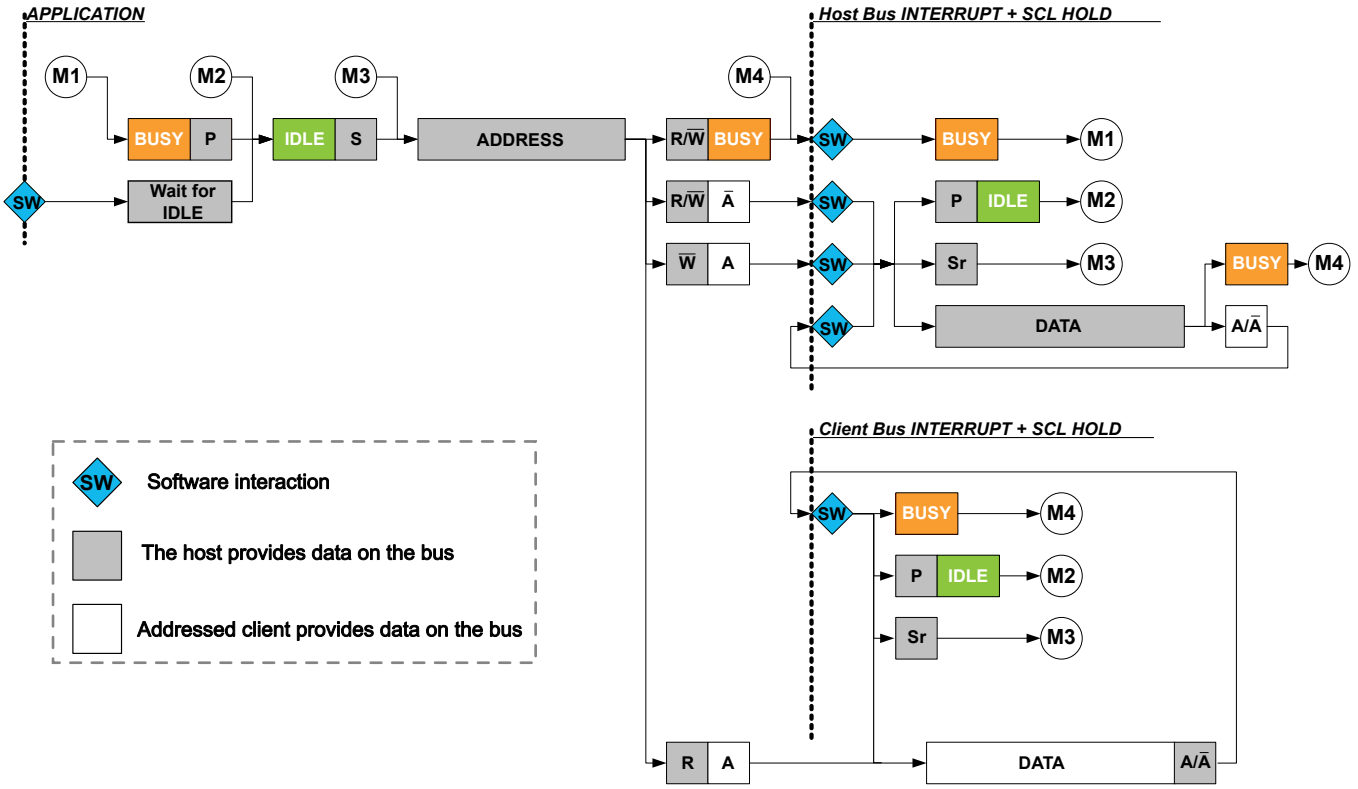
This diagram is used as a reference for the description of the I²C host operation throughout the document.

Figure 34-5. I²C Host Behavioral Diagram (SCLSM = 0)



In the second strategy (CTRLA.SCLSM = 1), interrupts only occur after the ACK bit, as illustrated in Figure 34-6. This strategy can be used when it is not necessary to check DATA before acknowledging.

Figure 34-6. I²C Host Behavioral Diagram (SCLSM = 1)



34.6.2.4.1 Host Clock Generation

The SERCOM peripheral supports several I²C bidirectional modes:

- Standard mode (*Sm*) up to 100 kHz
- Fast mode (*Fm*) up to 400 kHz
- Fast mode Plus (*Fm+*) up to 1 MHz

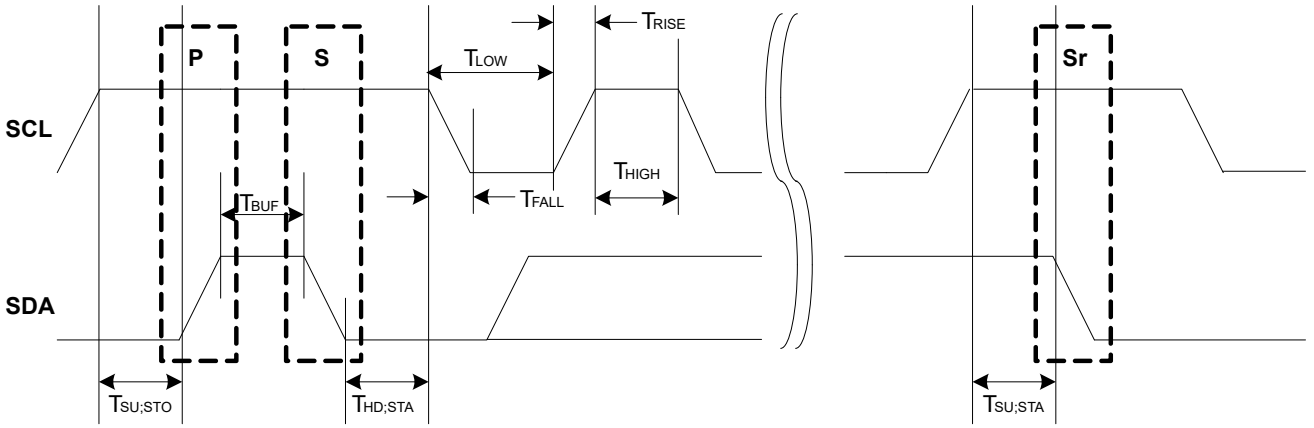
The Host clock configuration for *Sm, Fm* and *Fm+* are described in *Clock Generation (Standard-Mode, Fast-Mode and Fast-Mode Plus)* as follows.

Clock Generation (Standard-Mode, Fast-Mode and Fast-Mode Plus)

In I²C *Sm, Fm* and *Fm+* modes, the Host clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} is considered as part of T_{LOW}. Likewise, T_{RISE} is in a state between T_{LOW} and T_{HIGH} until a high state is detected.

Figure 34-7. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW) when BAUD.BAUDLOW = 0 or the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Setup time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Setup time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero
  See *Electrical Characteristics* from Related Links.

The SCL frequency is given by:

$$f_{\mathrm{SCL}} = \frac{1}{T_{\mathrm{LOW}} + T_{\mathrm{HIGH}} + T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case, the following formula gives the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + 2BAUD + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + BAUD + BAUDLOW + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{\mathrm{LOW}} = \frac{BAUDLOW + 5}{f_{\mathrm{GCLK}}}$$

$$T_{\mathrm{HIGH}} = \frac{BAUD + 5}{f_{\mathrm{GCLK}}}$$

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD must be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Start-up Timing: The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater start-up time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by the user, the Smart mode cannot be enabled.

Related Links
43. Electrical Characteristics

34.6.2.4.2 Transmitting Address Packets

The I²C host starts a bus transaction by writing the I²C client address to ADDR.ADDR and the direction bit, as described in Principle of Operation, see *Principle of Operation* from Related Links. If the bus is busy, the I²C host will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C host will issue a start condition on the bus. The I²C host will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C host, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Host on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C host is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the Arbitration Lost condition. In this case, the MB Interrupt flag and Host Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Host Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the Interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C client device responding to the address packet, then the INTFLAG.MB Interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C client is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a Stop condition (recommended) or resending the address packet by a repeated Start condition. When using SMBus logic, the client must ACK the address. If there is no response, it means that the client is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Host on Bus set

If the I²C host receives an acknowledge response from the I²C client, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.

- Transmit a new address packet by writing ADDR.ADDR. A repeated Start condition will automatically be inserted before the address packet.
- Issue a Stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Client on Bus set

If the I^2C host receives an ACK from the I^2C client, the I^2C host proceeds to receive the next byte of data from the I^2C client. When the first data byte is received, the Client on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I^2C operation to continue:

- Let the I^2C host continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in Smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a Stop condition.

Note: An ACK or NACK will be automatically transmitted if Smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK must be sent.

Related Links

34.6.1. Principle of Operation

34.6.2.4.3 Transmitting Data Packets

When an address packet with direction Host Write (see Figure 34-3) was transmitted successfully , INTFLAG.MB will be set. The I^2C host will start transmitting data via the I^2C bus by writing to DATA.DATA, and monitor continuously for packet collisions.

If a collision is detected, the I^2C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I^2C host will receive an ACK bit from the I^2C client, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I^2C Host on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I^2C host is not allowed to continue transmitting data packets if a NACK is received from the I^2C client.

34.6.2.4.4 Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I^2C host will already have received one data packet. The I^2C host must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

34.6.2.4.5 Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I^2C host will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the Smart mode.

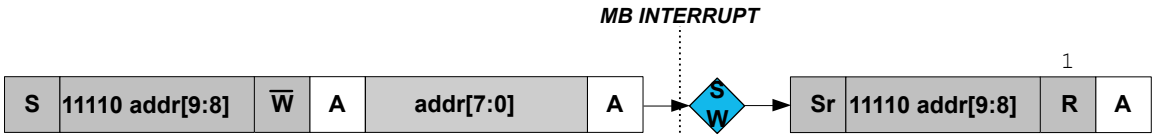
34.6.2.4.6 10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN = 1) and the Address bit field ADDR.ADDR is written, the two address bytes are transmitted as illustrated in the following figure. The addressed client acknowledges the two address

bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the host must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being '0'.

If the host receives a NACK after the first byte, the Write Interrupt flag is raised and the STATUS.RXNACK bit is set. If the first byte is acknowledged by one or more clients, the host proceeds to transmit the second address byte and the host will first see the Write Interrupt flag after the second byte is transmitted. If the transaction direction is read-from-client, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 34-8. 10-Bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. When

   the Host on the Bus interrupt is asserted, write the ADDR[7:0] register to '11110 address[9:8] 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

34.6.2.5 I²C Client Operation

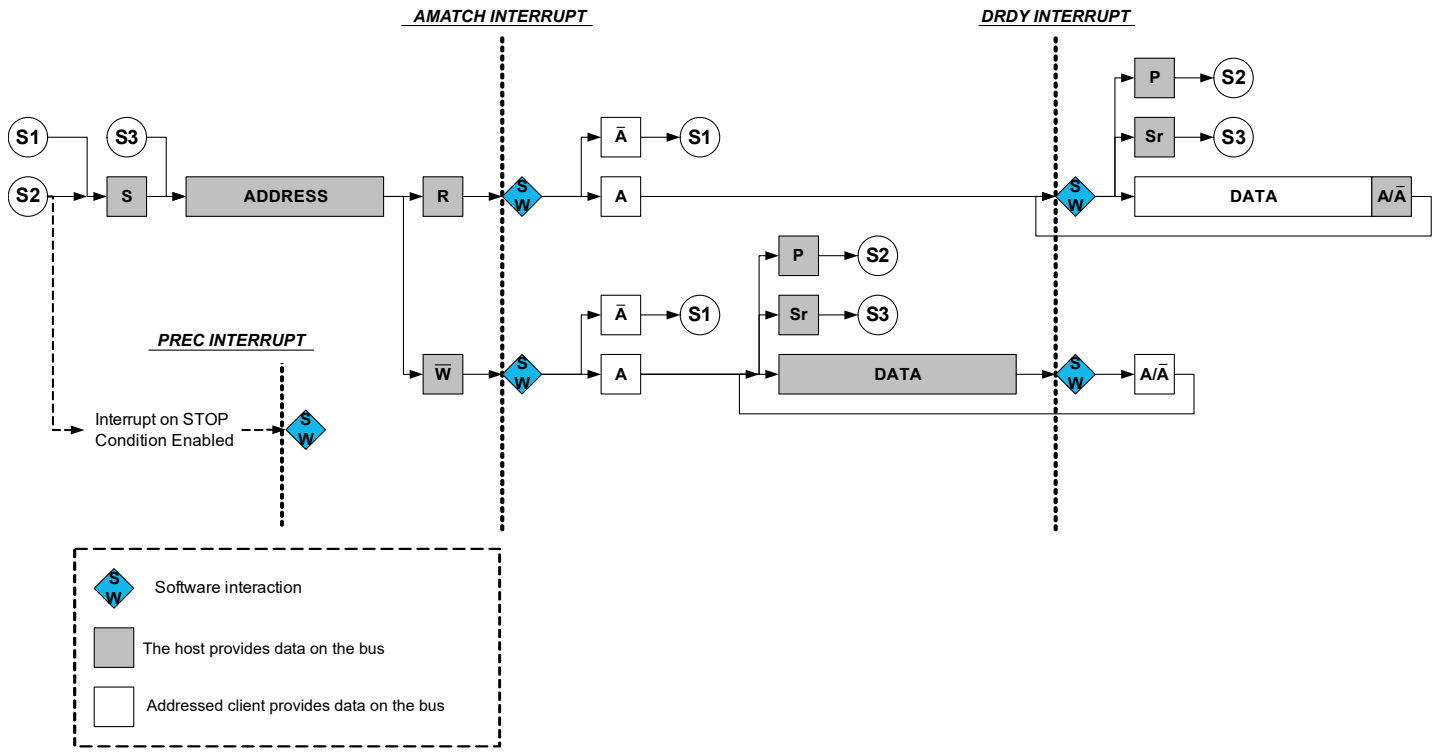
The I²C client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C client has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C client operates according to Figure 34-9. The circles labelled S*n* (S1, S2..) indicate the nodes the bus logic can jump to based on software or hardware interaction.

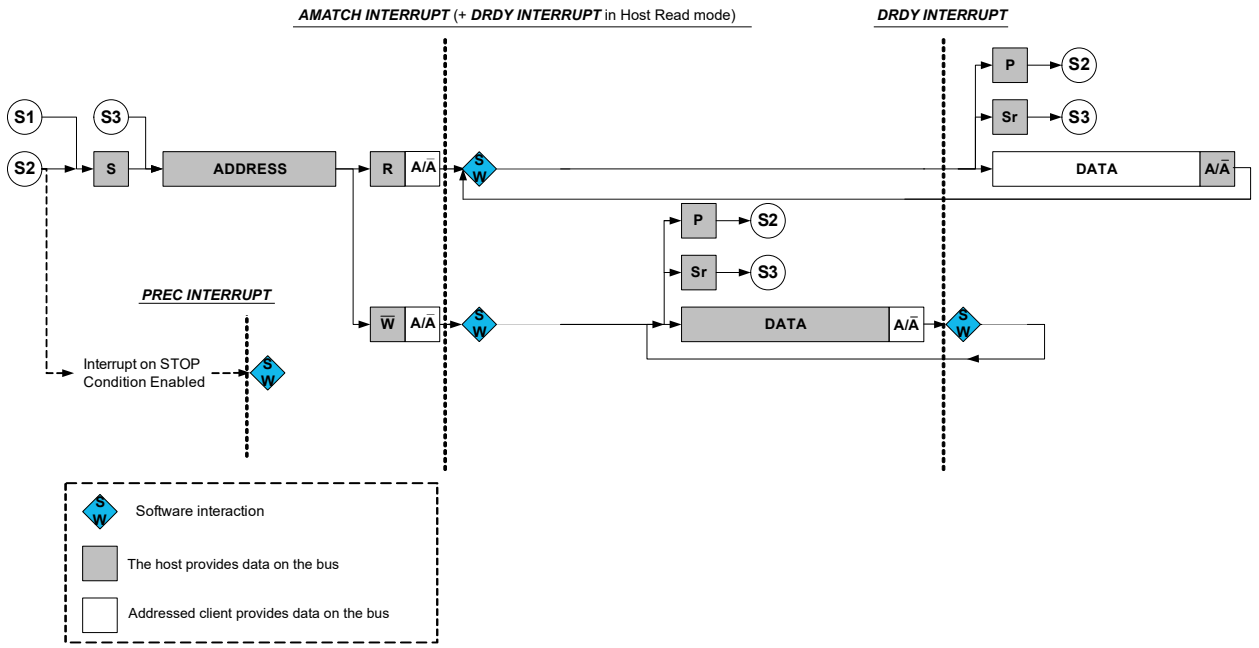
This diagram is used as a reference for the description of the I²C client operation throughout the document.

Figure 34-9. I²C Client Behavioral Diagram (SCLSM = 0)



In the second strategy (CTRLA.SCLSM = 1), interrupts only occur after the ACK bit is sent as illustrated Figure 34-10. This strategy can be used when it is not necessary to check DATA before acknowledging. For host reads, an address and data interrupt is issued simultaneously after the address acknowledge. However, for host writes, the first data interrupt is seen after the first data byte is received by the client and the acknowledge bit is sent to the host.

Figure 34-10. I²C Client Behavioral Diagram (SCLSM = 1)



34.6.2.5.1 Receiving Address Packets (SCLSM = 0)

When CTRLA.SCLSM = 0, the I²C client stretches the SCL line according to Figure 34-9. When the I²C client is properly configured, it waits for a Start condition.

When a Start condition is detected, the successive address packet is received and checked by the address match logic. If the received address is not a match, the packet is rejected, and the I²C client waits for a new Start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set.

SCL is stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit is updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet is received from the I²C host, one of two cases arises based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C client hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C client will wait for a new Start condition and address match.

Typically, software immediately acknowledges the address packet by sending an ACK/NACK bit. The I²C client Command bit field in the Control B register (CTRLB.CMD) can be written to 0x3 for both read and write operations, as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I²C host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C client waits for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C client waits for a new Start condition and address match. Typically, software immediately acknowledges the address packet by sending an ACK/NACK. The I²C client command CTRLB.CMD = 3 can be used for both read and write operations, as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH also causes an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

34.6.2.5.2 Receiving Address Packets (SCLSM = 1)

When SCLSM = 1, the I²C client stretches the SCL line only after an ACK as illustrated in Figure 34-10. When the I²C client is properly configured, it waits for a Start condition to be detected.

When a Start condition is detected, the successive address packet is received and checked by the address match logic.

If the received address is not a match, the packet is rejected and the I²C client waits for a new Start condition.

If the address matches, the acknowledge action, as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT), is sent and the Address Match bit in the Interrupt Flag register

(INTFLAG.AMATCH) is set. SCL is stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit is updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet is received from the I²C host, INTFLAG.AMATCH can be set to '1' to clear it.

34.6.2.5.3 Receiving and Transmitting Data Packets

After the I²C client receives an address packet, it responds according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY is set. After receiving data, the I²C client sends an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set and SCL is held low, pending software interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY Interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK = 1, the I²C client must expect a stop or a repeated start to be received. The I²C client must release the data line to allow the I²C host to generate a stop or repeated start. Upon detecting a Stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) is set and the I²C client returns to Idle state.

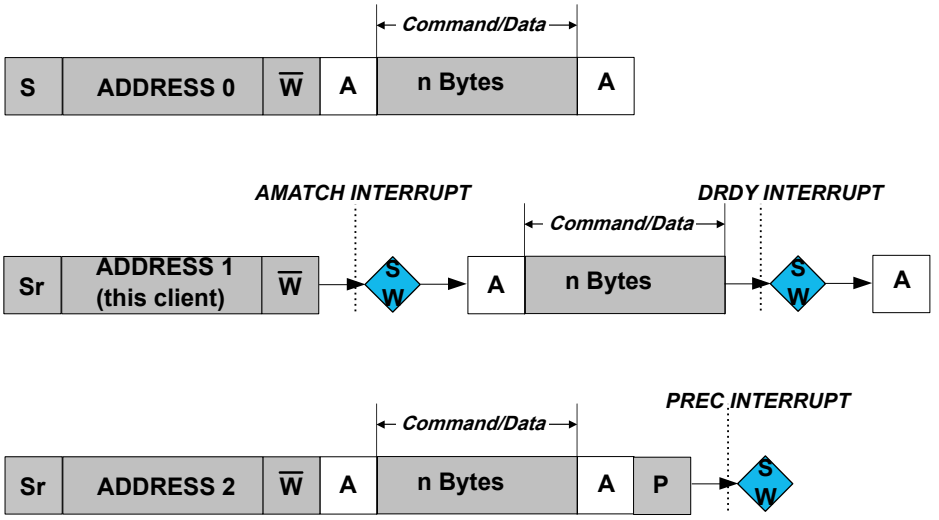
34.6.2.5.4 PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD = 1) and 7-bit addressing is used, INTFLAG.PREC is set if the client was addressed since the last STOP condition. When CTRLB.GCMD = 0, a STOP condition without an address match is not set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the clients addressed during the group command, they all begin executing the command they received.

The following figure illustrates an example where this client, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple clients addressed before and after this client. Eventually, at the end of the group command, a single STOP is generated by the host. At this point, a STOP interrupt is asserted.

Figure 34-11. PMBus Group Command Example



34.6.3 Additional Features

34.6.3.1 SMBus

The I^{2}C includes three hardware SCL low time-outs that allow a time-out to occur for SMBus SCL low time-out, host extend time-out and client extend time-out. This allows for SMBus functionality. These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32 KHz_LPCLK. The I^{2}C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25-35 ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a client device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the host device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

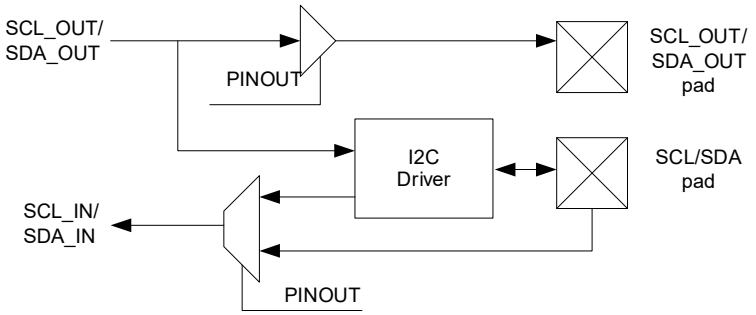
34.6.3.2 Smart Mode

The I^{2}C interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^{2}C protocol. The Smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

34.6.3.3 Four-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) enables Four-Wire mode operation. In this mode, the internal I^{2}C tri-state drivers are bypassed and an external I^{2}C compliant tri-state driver is needed when connecting to an I^{2}C bus.

Figure 34-12. I²C Pad Interface



34.6.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding Interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the client acknowledges the address. At this point, the software can either issue a Stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

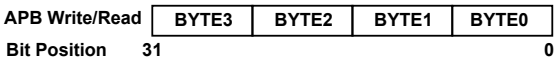
34.6.3.5 32-Bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B = 1). When enabled, write and read transactions to/from the DATA register are 32 bits in size.

If frames are not multiples of 4 bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0-3.

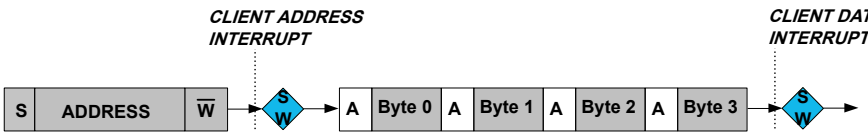
Figure 34-13. 32-Bit Extension Byte Ordering



32-Bit Extension Client Operation

The following figure illustrates a transaction with 32-bit Extension enabled (CTRLC.DATA32B = 1). In client operation, the Address Match interrupt in the Interrupt Flag Status and Clear register (INTFLAG.AMATCH) is set after the address is received and available in the DATA register. The Data Ready interrupt (INTFLAG.DRDY) will, then, be raised for every 4 bytes transferred.

Figure 34-14. 32-Bit Extension Client Operation



The LENGTH register can be written before the frame begins or when the AMATCH interrupt is set. If the frame size is not LENGTH.LEN bytes, the Length Error status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.DRDY interrupt is raised when the last byte is received for host reads. For host writes, the last data byte is automatically NACKed. On address recognition, the internal length counter is reset in preparation for the incoming frame.

When SCL clock stretch mode is selected (CTRLA.SCLSM = 1) and the transaction is a host write, the selected Acknowledge Action (CTRLB.ACKACT) will only be used to ACK/NACK each 4th byte. All other

bytes are ACKed. This allows the user to write CTRLB.ACKACT = 1 in the final interrupt, so that the last byte in a 32-bit word is NACKed.

Writing to the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 bytes, the remainder is lost.

32-Bit Extension Host Operation

When using the I^2C configured as host, the Address register must be written with the desired address (ADDR.ADDR) and, optionally, the transaction Length and transaction Length Enable bits (ADDR.LEN and ADDR.LENEN) can be written. When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. Then, the ADDR.LEN bytes are transferred, followed by an automatically generated NACK (for host reads) and a STOP.

The INTFLAG.SB or INTFLAG.MB are raised for every 4 bytes transferred. If the transaction is a host read and ADDR.LEN is not a multiple of 4 bytes, the final INTFLAG.SB is set when the last byte is received.

When the SCL Clock Stretch mode is enabled (CTRLA.SCLSM = 1) and the transaction is a host read, the selected Acknowledge Action (CTRLB.ACKACT) is used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to set CTRLB.ACKACT = 1 in the final interrupt, so that the last byte in a 32-bit word is NACKed.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP is automatically generated and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

34.6.4 DMA, Interrupts and Events

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the interrupt flag is cleared, the interrupt is disabled or the I^2C is reset. See the *INTFLAG* (Client) or *INTFLAG* (Host) register from Related Links for details on how to clear interrupt flags.

Table 34-2. Module Request for SERCOM I^2C Client

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Client transmit mode)	Yes (request cleared when data is written)	—	NA
Data received (RX) (Client receive mode)	Yes (request cleared when data is read)	—	
Data Ready (DRDY)	—	Yes	
Address Match (AMATCH)	—	Yes	
Stop received (PREC)	—	Yes	
Error (ERROR)	—	Yes	

Table 34-3. Module Request for SERCOM I²C Host

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Host transmit mode)	Yes (request cleared when data is written)	—	NA
Data needed for transmit (RX) (Host transmit mode)	Yes (request cleared when data is read)	—	
Host on Bus (MB)	—	Yes	
Stop received (SB)	—	Yes	
Error (ERROR)	—	Yes	

Related Links
34.8.6. INTFLAG

34.6.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN = 1.

34.6.4.1.1 Host DMA

When using the I²C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN) and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0-255. DMA is, then, used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for host reads) and a STOP.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP is automatically generated and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

The I²C host generates the following requests:

- Write data received (RX) – The request is set when host write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX) – The request is set when data is needed for a host read operation. The request is cleared when DATA is written.
- Write data received (RX) – If the FIFO is disabled, the request is set when host write data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached (CTRLC.RXTRHOLD). The request is cleared when DATA is read.
- Read data needed for transmit (TX) – If the FIFO is disabled, the request is set when data are needed for a host read operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

34.6.4.1.2 Client DMA

When using the I²C client with DMA, an address match causes the address Interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt is serviced, the data transfer is performed through DMA.

The I²C client generates the following requests:

- Read data received (RX) – The request is set when host read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX) – The request is set when data are needed for a host write operation. The request is cleared when DATA is written.

- Read data received (RX) – If the FIFO is disabled, the request is set when host read data are received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached. The request is cleared when DATA is read.
- Write data needed for transmit (TX) – If the FIFO is disabled, the request is set when data are needed for a host write operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

34.6.4.2 Interrupts

The I^2C client has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I^2C host has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- Client on Bus (SB)
- Host on Bus (MB)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

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 I^2C is reset. For details on how to clear Interrupt flags, see *INTFLAG* register from Related Links.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)
34.8.6. INTFLAG

34.6.4.3 Events

Not applicable.

34.6.4.4 Sleep Mode Operation

I^2C Host Operation

The generic clock (GCLK_SERCOMx_CORE) continues to run in the Idle Sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE also runs in the Standby Sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY = 0, the GLK_SERCOMx_CORE is disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I^2C Client Operation

Writing CTRLA.RUNSTDBY = 1 allows the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY = 0, all receptions are dropped.

34.7 Register Summary

See the *SERCOM0/SERCOM1/SERCOM2* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN							PINOUT
		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		23:16	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
		31:24								
0x08	CTRLC	7:0					SDASETUP[3:0]			
		15:8								
		23:16								
		31:24								DATA32B
0x0C ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
		15:8					LENERR		SEXTTOUT	
0x1C	SYNCBUSY	7:0				LENGTH		SYSOP	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8								LENEN
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
		15:8						ADDR[9:7]		
		23:16	ADDRMASK[6:0]							
		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x33	Reserved									
0x34	FIFOSPACE	7:0				TXSPACE[4:0]				
		15:8				RXSPACE[4:0]				
0x36	FIFOPTR	7:0					CPUWRPTR[3:0]			
		15:8					CPURDPTR[3:0]			

Related Links

8. Product Memory Mapping Overview

34.8 Register Description - I²C Client

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 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.

34.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN							PINOUT
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out

This bit enables the SCL low time-out. If SCL is held low for 25-35 ms, the client releases its clock hold, if enabled, and resets the internal state machine. Any interrupt flags set at the time of time-out remain set.
This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL is stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-9.
1	SCL stretch only after ACK bit according to Figure 34-10.

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	Reserved
0x3	Reserved

Bit 23 – SEXTTOEN Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25 ms from the initial START to a STOP, the client releases its clock hold, if enabled, and resets the internal state machine. Any interrupt flags set at the time of time-out remain set. If the address was recognized, PREC is set when a STOP is received.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	Four-wire operation disabled
1	Four-wire operation enabled

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in the Standby Sleep mode.
This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x04 to select the I^2C client serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state and the SERCOM is disabled.
Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.
Note: During SWRST, access to registers/bits without SWRST are disallowed until the hardware clears SYNCBUSY.SWRST.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

34.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
Access	R/W	R/W				R/W	W	W
Reset	0	0				0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:22 – FIFOCLR[1:0] FIFO Clear
When these bits are set, the corresponding FIFO is cleared. The bits automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Bit 18 – ACKACT Acknowledge Action
This bit defines the client's acknowledge behavior after an address or data byte is received from the host. The acknowledge action is executed when a command is written to the CMD bits. If Smart mode is enabled (CTRLB.SMEN = 1), the acknowledge action is performed when the DATA register is read.
ACKACT might not be updated more than once between each peripheral interrupts request.
This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command
This bit field triggers the client operation as follows. The CMD bits are strobe bits and always read as '0'. The operation is dependent on the client interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.

All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.
This bit is not enable-protected.

Table 34-4. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Host read)	Wait for any start (S/Sr) condition
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute acknowledge action succeeded by client data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the Addressing mode.

Value	Name	Description
0x0	MASK	The client responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK.
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The client responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	—	Reserved.

Bit 10 – AACKEN Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Received interrupt flag (INTFLAG.PREC) is set when a STOP condition is detected if the client was addressed since the last STOP condition on the bus.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable
When Smart mode is enabled, data are acknowledged automatically when DATA.DATA is read.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
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
					SDASETUP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 24 – DATA32B Data 32 Bit
This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transaction to/from DATA are 8-bit in size
1	Data transaction to/from DATA are 32-bit in size

Bits 3:0 – SDASETUP[3:0] SDA Setup Time
These bits select the minimum SDA-to-SCL setup time, measured from the release of SDA to the release of SCL:
$t_{\mathrm{SU:DAT}} = (GCLK\_SERCOMx \times \mathrm{APB\ period}\ (PBx\_CLK)) \times (6 + 16 \times \mathrm{SDASETUP})$

34.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
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 register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 4 – RXFF RX FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

Value	Description
0	The RX FIFO Full interrupt is disabled.
1	The RX FIFO Full interrupt is enabled.

Bit 3 – TXFE TX FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

Value	Description
0	The TX FIFO Empty interrupt is disabled.
1	The TX FIFO Empty interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Ready bit, which disables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

34.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 4 – RXFF RX FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

Value	Description
0	The RX FIFO Full interrupt is disabled.
1	The RX FIFO Full interrupt is enabled.

Bit 3 – TXFE TX FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

Value	Description
0	The TX FIFO Empty interrupt is disabled.
1	The TX FIFO Empty interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Ready Interrupt Enable bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit sets the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

34.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 7 – ERROR Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are LENERR, SEXTTOUT, LOWTOUT, COLL and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 4 – RXFF RX FIFO Full
This flag is set when RX FIFO Threshold locations are fulfilled.
The flag is cleared when the RX FIFO is empty.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the RX FIFO Full interrupt flag.

Bit 3 – TXFE TX FIFO Empty
This flag is set when TX FIFO Threshold locations are available.
The flag is cleared when the TX FIFO is full.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the TX FIFO Empty interrupt flag.

Bit 2 – DRDY Data Ready
This flag is set when a I^2C client byte transmission is successfully completed.
The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with Smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready Interrupt flag.

Bit 1 – AMATCH Address Match
This flag is set when the I^2C client address match logic detects that a valid address has been received.
The flag is cleared by hardware when CTRL.CMD is written.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match Interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received
This flag is set when a Stop condition is detected for a transaction being processed. A Stop condition detected between a bus host and another client will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).

This flag is cleared by hardware after a command is issued on the next address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received Interrupt flag.

34.8.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
					LENERR		SEXTTOUT	
Access					R/W		R/W	
Reset					0		0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 11 – LENERR Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Bit 11 – LENERR Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No length error has occurred.
1	Length error has occurred.

Bit 9 – SEXTTOUT Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD Clock Hold
The client Clock Hold bit (STATUS.CLKHOLD) is set when the client is holding the SCL line low, stretching the I2C clock. Software must consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.
This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT SCL Low Time-out
This bit is set if an SCL low time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR Repeated Start

When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.
This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR Read / Write Direction

The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a host .

Value	Description
0	Host write operation is in progress.
1	Host read operation is in progress.

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Host responded with ACK.
1	Host responded with NACK.

Bit 1 – COLL Transmit Collision

If set, the I2C client was not able to transmit a high data or NACK bit, the I2C client will immediately release the SDA and SCL lines and wait for the next packet addressed to it.
This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and must be treated as a bus error.
Note: This status will not trigger any interrupt, and must be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR Bus Error

The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.
This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No bus error detected.

Value	Description
1	Bus error detected.

34.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
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
				LENGTH		SYSOP	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy
Writing LENGTH requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, an APB error will be generated.
Note: In client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.FIFOCLR when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE = 1 until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST = 1 until synchronization is complete.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.8.9 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

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

Bit	7	6	5	4	3	2	1	0
	LEN[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 8 – LENEN Data Length Enable
In 32-bit Extension mode (CTRLC.DATA32B = 1), this bit field enables the length counter.

Value	Description
0	Length counter is disabled.
1	Length counter is enabled.

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode (CTRLC.DATA32B = 1) with Data Length counting enabled (LENGTH.LENEN), this bit field configures the data length from 0 to 255 bytes, after which, the flag INTFLAG.DRDY is raised.

34.8.10 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
						ADDR[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
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 26:17 – ADDRMASK[9:0] Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bits 10:1 – ADDR[9:0] Address
These bits contain the I^2C client address used by the client address match logic to determine if a host has addressed the client.
When using 7-bit addressing, the client address is represented by ADDR[6:0].
When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (host write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

34.8.11 Data

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

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

The client data register I/O location (DATA.DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the client (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

34.8.12 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: -

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:8 – RXSPACE[4:0] RX FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] TX FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

34.8.13 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: -

This register provides a copy of internal CPU TX and RX FIFO pointers.

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

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

Bits 11:8 – CPURDPTR[3:0] RX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] TX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writting to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

34.9 Register Summary

See the *SERCOM0/SERCOM1/SERCOM2* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN	MEXTTOEN						PINOUT
		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8							QCEN	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08	CTRLC	7:0								
		15:8								
		23:16								
		31:24								DATA32B
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		23:16								
		31:24								
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR			RXFF	TXFE		SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR			RXFF	TXFE		SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR			RXFF	TXFE		SB	MB
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
		15:8	TENBITEN		LENEN			ADDR[10:8]		
		23:16	LEN[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	7:0				TXSPACE[4:0]				
		15:8				RXSPACE[4:0]				
0x36	FIFOPTR	7:0					CPUWRPTR[3:0]			
		15:8					CPURDPTR[3:0]			

Related Links

8. Product Memory Mapping Overview

34.10 Register Description - I²C Client

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 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.

34.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
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
	SEXTTOEN	MEXTTOEN						PINOUT
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out

This bit enables the SCL low time-out. If SCL is held low for 25-35 ms, the host releases its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted. INTFLAG.SB or INTFLAG.MB is set as normal but the clock hold is released. The STATUS.LOWTOUT and STATUS.BUSERR status bits is set.
This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0] Inactive Time-Out

If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic is set to Idle. An inactive bus arises when either an I^2C host or client is holding the SCL low.
Enabling this option is necessary for SMBus compatibility but can also be used in a non-SMBus setup.
Calculated time-out periods are based on a 100 kHz baud rate.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60 µs)
0x2	105US	10-11 SCL cycle time-out (100-110 µs)
0x3	205US	20-21 SCL cycle time-out (200-210 µs)

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL is stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-5.
1	SCL stretch only after ACK bit according to Figure 34-6.

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	Reserved
0x3	Reserved

Bit 23 – SEXTTOEN Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25 ms from the initial START to a STOP, the host releases its clock hold, if enabled, and completes the current transaction. A STOP will automatically be transmitted.
SB or MB is set as normal but CLKHOLD is release. The MEXTTOUT and BUSERR status bits is set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN Host SCL Low Extend Time-Out

This bit enables the host SCL low extend time-out. If SCL is cumulatively held low for greater than 10 ms from START-to-ACK, ACK-to-ACK or ACK-to-STOP the host releases its clock hold if enabled and completes the current transaction. A STOP will automatically be transmitted.
SB or MB is set as normal but CLKHOLD is released. The MEXTTOUT and BUSERR status bits is set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	Four-wire operation disabled.
1	Four-wire operation enabled.

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in the Standby Sleep mode.
This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C host will not operate in the Standby Sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all Sleep modes.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x5 to select the I^2C host serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately and the Synchronization Enable

Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state and the SERCOM is disabled.
Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation is discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until the hardware clears SYNCBUSY.SWRST.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

34.10.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	W	W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action

This bit defines the I^2C host's acknowledge behavior after a data byte is received from the I^2C client. The acknowledge action is executed when a command is written to CTRLB.CMD or if Smart mode is enabled (CTRLB.SMEN is written to one) when DATA.DATA is read.
This bit is not enable-protected.
This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command

Writing these bits triggers a host operation as described below. The CMD bits are strobe bits and always read as '0'. The acknowledge action is only valid in Host Read mode. In Host Write mode, a command will only result in a repeated Start or Stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and, then, the acknowledge action is updated before the command is triggered.
Commands can only be issued when either the Client on Bus interrupt flag (INTFLAG.SB) or Host on Bus interrupt flag (INTFLAG.MB) is '1'.
If CMD 0x1 is issued, a repeated start is issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This triggers a repeated start followed by the transmission of the new address.
Issuing a command sets the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 34-5. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable

This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable

When Smart mode is enabled, acknowledge action is sent when DATA.DATA is read.
This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.10.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
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
Access								
Reset								

Bit 24 – DATA32B Data 32 Bit

This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transactions to/from DATA are 8-bit in size
1	Data transactions to/from DATA are 32-bit in size

34.10.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, 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
	BAUDLOW[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
	BAUD[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 15:8 – BAUDLOW[7:0] Host Baud Rate Low

If this bit field is non-zero, the SCL low time will be described by the value written.
For more details on how to calculate the frequency, see *Clock Generation – Baud-Rate Generator* from Related Links.

Bits 7:0 – BAUD[7:0] Host Baud Rate

This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.
For more details on how to calculate the frequency, see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links

31.6.2.3. Clock Generation – Baud-Rate Generator

34.10.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
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 are also reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	R/W			R/W	R/W		R/W	R/W
Reset	0			0	0		0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 4 – RXFF RX FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

Value	Description
0	The RX FIFO Full interrupt is disabled.
1	The RX FIFO Full interrupt is enabled.

Bit 3 – TXFE TX FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

Value	Description
0	The TX FIFO Empty interrupt is disabled.
1	The TX FIFO Empty interrupt is enabled.

Bit 1 – SB Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Client on Bus Interrupt Enable bit, which disables the Client on Bus interrupt.

Value	Description
0	The Client on Bus interrupt is disabled.
1	The Client on Bus interrupt is enabled.

Bit 0 – MB Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Host on Bus Interrupt Enable bit, which disables the Host on Bus interrupt.

Value	Description
0	The Host on Bus interrupt is disabled.
1	The Host on Bus interrupt is enabled.

34.10.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
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 are also reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	R/W			R/W	R/W		R/W	R/W
Reset	0			0	0		0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 4 – RXFF RX FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

Value	Description
0	The RX FIFO Full interrupt is disabled.
1	The RX FIFO Full interrupt is enabled.

Bit 3 – TXFE TX FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

Value	Description
0	The TX FIFO Empty interrupt is disabled.
1	The TX FIFO Empty interrupt is enabled.

Bit 1 – SB Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Client on Bus Interrupt Enable bit, which enables the Client on Bus interrupt.

Value	Description
0	The Client on Bus interrupt is disabled.
1	The Client on Bus interrupt is enabled.

Bit 0 – MB Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Host on Bus Interrupt Enable bit, which enables the Host on Bus interrupt.

Value	Description
0	The Host on Bus interrupt is disabled.
1	The Host on Bus interrupt is enabled.

34.10.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	R/W			R/W	R/W		R/W	R/W
Reset	0			0	0		0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 4 – RXFF RX FIFO Full
This flag is set when RX FIFO Threshold locations are fulfilled.
The flag is cleared when the RX FIFO is empty.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the RX FIFO Full interrupt flag.

Bit 3 – TXFE TX FIFO Empty
This flag is set when TX FIFO Threshold locations are available.
The flag is cleared when the TX FIFO is full.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the TX FIFO Empty Interrupt flag.

Bit 1 – SB Client on Bus
The Client on Bus flag (SB) is set when a byte is successfully received in the Host Read mode, for example, no arbitration lost or bus error occurred during the operation. When this flag is set, the host forces the SCL line low, stretching the I^2C clock period. The SCL line is released and SB is cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location clears the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

Bit 0 – MB Host on Bus
This flag is set when a byte is transmitted in Host Write mode. The flag is set regardless of the occurrence of a bus error or an Arbitration Lost condition. MB is also set when arbitration is lost during sending of a NACK in the Host Read mode or when issuing a Start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the host forces the SCL line low, stretching the I^2C clock period. The SCL line is released and MB is cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location clears the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

34.10.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	R/W	R/W	R/W		R	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR Transaction Length Error
This bit is set when automatic length is used for a DMA and/or 32-bit transaction and the client sends a NACK before ADDR.LEN bytes are written by the host.
Writing '1' to this bit location clears STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 9 – SEXTTOUT Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is automatically cleared when writing to the ADDR register.
Writing '1' to this bit location clears SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 8 – MEXTTOUT Host SCL Low Extend Time-Out
This bit is set if a Host SCL low time-out occurs.
Writing '1' to this bit location clears STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 7 – CLKHOLD Clock Hold
This bit is set when the host is holding the SCL line low, stretching the I^2C clock. Software must consider this bit when INTFLAG.SB or INTFLAG.MB is set.
This bit is cleared when the corresponding Interrupt flag is cleared and the next operation is given.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Bit 6 – LOWTOUT SCL Low Time-Out
This bit is set if an SCL low time-out occurs.
Writing '1' to this bit location clears this bit. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0] Bus State

These bits indicate the current I²C Bus state.
When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the Idle state. The Bus state cannot be forced into any other state.
Writing BUSSTATE to Idle sets SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The Bus state is unknown to the I²C host and waits for a Stop condition to be detected or wait to be forced into an Idle state by software
0x1	IDLE	The Bus state is waiting for a transaction to be initialized
0x2	OWNER	The I²C host is the current owner of the bus
0x3	BUSY	Some other I²C host owns the bus

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last address or data packet sent was acknowledged or not.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	Client responded with ACK.
1	Client responded with NACK.

Bit 1 – ARBLOST Arbitration Lost

This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit or while issuing a Start or Repeated Start condition on the bus. The Host on Bus Interrupt flag (INTFLAG.MB) is set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register automatically clears STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.
This bit is not write-synchronized.

Bit 0 – BUSERR Bus Error

This bit indicates that an illegal Bus condition occurred on the bus, regardless of bus ownership. An illegal Bus condition is detected if a protocol violating start, repeated start or stop is detected on the I²C bus lines. A Start condition directly followed by a Stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and sets BUSERR.
If the I²C host is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB are set in addition to BUSERR.
Writing the ADDR.ADDR register automatically clears the BUSERR flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.
This bit is not write-synchronized.

34.10.9 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
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								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.10.10 Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[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	15	14	13	12	11	10	9	8
	TENBITEN		LENEN			ADDR[10:8]		
Access	R/W		R/W			R/W	R/W	R/W
Reset	0		0			0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[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 23:16 – LEN[7:0] Transaction Length
These bits define the transaction length of a DMA and/or 32-bit transaction from 0-255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 13 – LENEN Transfer Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:0 – ADDR[10:0] Address
When ADDR is written, the consecutive operation depends on the Bus state:

Table 34-6.

State	Description
UNKNOWN	INTFLAG.MB and STATUS.BUSERR are set and the operation is terminated.
BUSY	The I^2C host awaits further operation until the bus becomes IDLE.

.....continued

State	Description
IDLE	The I^2C host issues a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low and STATUS.CLKHOLD and INTFLAG.MB are set.
OWNER	A repeated start sequence is performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.
Note: STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB is cleared when ADDR is written.	

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the host logic to perform any bus protocol related operations.
The I^2C host control logic uses bit '0' of ADDR as the bus protocol's read/write flag (R/W); '0' for write and '1' for read.

34.10.11 Data

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

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

The host data register I/O location (DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the host (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

34.10.12 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34.10.13 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: -

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:8 – RXSPACE[4:0] RX FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] TX FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

34.10.14 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: -

This register provides a copy of internal CPU TX and RX FIFO pointers.

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

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

Bits 11:8 – CPURDPTR[3:0] RX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] TX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writting to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

35. Quad Serial Peripheral Interface (QSPI)

35.1 Overview

The Quad SPI Interface (QSPI) circuit is a synchronous serial data link that provides communication with external devices in Host mode.

The QSPI can be used in SPI mode to interface serial peripherals, such as ADCs, DACs, LCD controllers and sensors or in Serial Memory mode to interface serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to SRAM. The serial Flash memory mapping is visible in the system as other memories (ROM, SRAM, embedded Flash memories and so on). XIP mode is not available for a secured or code protected device for security reasons. This is achieved by blocking off AHB accesses (only instruction) on the CPU side from QSPI. QSPI returns an error on the transactions where the CPU is requesting instructions from QSPI-connected external memories.

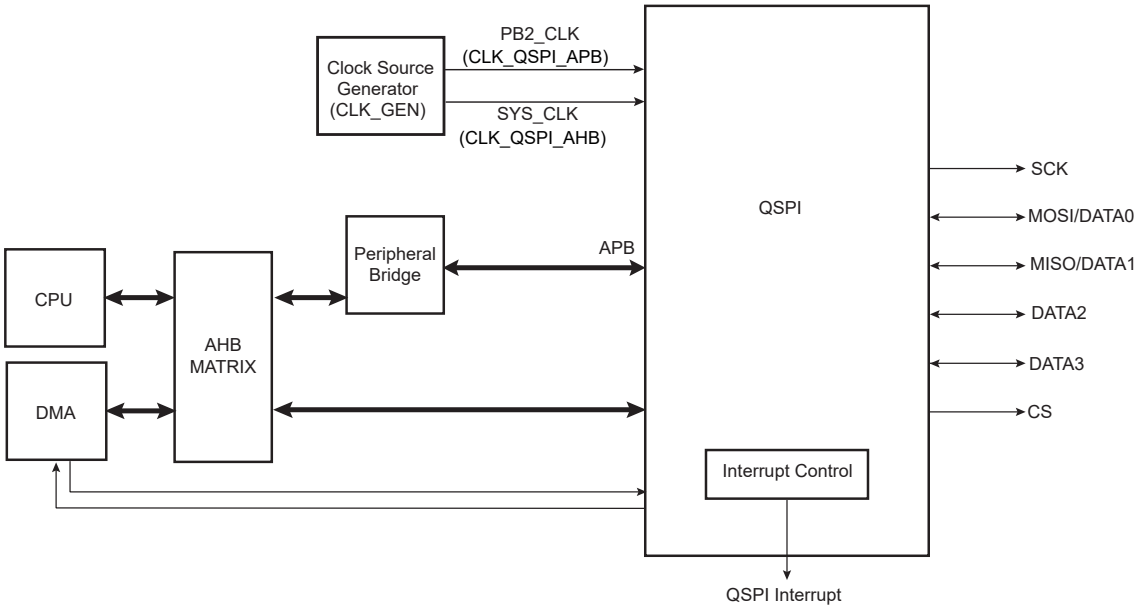
With the support of the Quad-SPI protocol, the QSPI allows the system to use high-performance serial Flash memories that are small and inexpensive in place of larger and more expensive parallel Flash memories.

35.2 Features

- Host SPI Interface:
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode:
  - To use serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit, 16-bit or 32-bit programmable data length
- Serial Memory Mode:
  - To use serial Flash memories operating in single-bit SPI, Dual SPI and Quad SPI
  - Supports Execute in Place (XIP). The system can execute code directly from a Serial Flash memory
  - Flexible instruction register to be compatible with all serial Flash memories
  - 32-bit Address mode (default is 24-bit address) to support serial Flash memories larger than 128 Mbit
  - Continuous Read mode
  - Scrambling/Unscrambling On-the-Fly
  - Double data rate support (Read only)
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver and one channel for the transmitter
- Register Write Protection

35.3 Block Diagram

Figure 35-1. QSPI Block Diagram



35.4 Signal Description

Table 35-1. Quad-SPI Signals

Signal	Description	Type
SCK	Serial Clock	Output
CS	Chip Select	Output
MOSI(DATA0)	Data Output (Data Input Output 0)	Output (Input/Output)
MISO(DATA1)	Data Input (Data Input Output 1)	Input (Input/Output)
DATA2	Data Input Output 2	Input/Output
DATA3	Data Input Output 3	Input/Output

Notes:

1. MOSI and MISO are used for single-bit SPI operation.
2. DATA0-DATA1 are used for Dual SPI operation.
3. DATA0-DATA3 are used for Quad SPI operation.

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links for details on the pin mapping for the QSPI peripheral.

Related Links

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

35.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

35.5.1 I/O Lines

Using the QSPI I/O lines requires the I/O pins to be configured using the System Configuration registers (see *System Configuration and Register Locking (CFG)* from Related Links) (QSPI_HSEN of

CFGCON1/DEVCFG1 register) for direct or PPS. If QSPI pins are selected through PPS, the PPS registers have to be configured (see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links).

If QSPI_HSEN = 1, QSPI uses dedicated pins.

If QSPI_HSEN = 0, QSPI uses PPS path and I/O pins are multiplexed to pins groups defined in PPS section.

Related Links

6. I/O Ports and Peripheral Pin Select (PPS)
22. System Configuration and Register Locking (CFG)

35.5.2 Power Management

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

18. Power Management Unit (PMU)

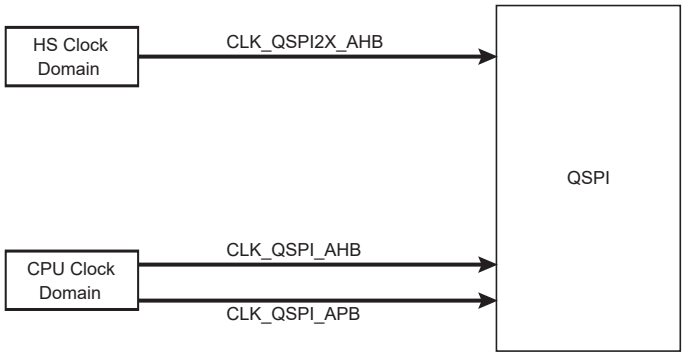
35.5.3 Clocks

An AHB clock (CLK_QSPI_AHB) is required to clock the QSPI. In PIC32CX-BZ3, SYS_CLK is the AHB clock and can be configured in the CRU.

A FAST clock (CLK_QSPI2X_AHB) is required to clock the QSPI. This clock can be enabled and disabled in the CFGCON1 register, bit 29 (CFGCON1.QSPIDDRM). When using QSPI DDR (Double Data Rate) mode, the System Clock (SYS_CLK) must be less than or equal to 32 MHz.

PB2_CLK is the CLK_QSPI_APB clock and can be configured in CRU registers.

Figure 35-2. QSPI Clock Organization



Important: The CLK_QSPI2x_AHB must be two times faster to CLK_QSPI_AHB when the QSPI is operated in the DDR mode. In Single Data Rate (SDR), the CLK_QSPI2x_AHB is not used.

The CLK_QSPI_APB, CLK_QSPI_AHB and CLK_QSPI2X_AHB, respectively, are all synchronous but can be divided by a prescaler.

35.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). Using the QSPI DMA requests requires the DMA Controller to be configured first.

Note: DMAC write access must be 32-bit aligned. If a single byte is to be written in a 32-bit word, the rest of the word must be filled with 'ones'.

35.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the QSPI interrupts requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

35.5.6 Events

Not applicable.

35.5.7 Debug Operation

When the CPU is halted in debug mode the QSPI continues normal operation. If the QSPI 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.

35.5.8 Register Access Protection

All registers with write access are optionally write protected by the PAC, except the following registers:

- Control A (CTRLA) register
- Transmit Data (TXDATA) register
- Interrupt Flag Status and Clear (INTFLAG) register
- Scrambling Key (SCRAMBKEY) register

PAC write protection is denoted by the PAC Write-Protection property in the register description.

Write-protection does not apply to accesses through an external debugger.

35.6 Functional Description

35.6.1 Principle of Operation

The QSPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral or serial memory devices.

The QSPI operates as a host. It initiates and controls all data transactions.

When transmitting, the TXDATA register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the RXDATA register, and the receiver is ready for a new character.

35.6.2 Basic Operation

35.6.2.1 Initialization

After Power-On Reset, this peripheral is enabled .

35.6.2.2 Enabling, Disabling and Resetting

The peripheral is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE).

The peripheral is disabled by writing a '0' to CTRLA.ENABLE.

The peripheral is reset by writing a '1' to the Software Reset bit (CTRLA.SWRST).

35.6.3 Transfer Data Rate

By default, the QSPI module is enabled in single data rate mode. In this operating mode, the CLK_QSPI2X_AHB clock is not used and must be disabled.

The dual data rate operating mode is enabled by writing a '1' to the Double Data Rate Enable bit in the CFGCON1 register (CFGCON1.QSPIDDRM). This operating mode requires the CLK_QSPI2X_AHB clock and must be enabled before writing the DDREN bit.

35.6.4 Serial Clock Baud Rate

The QSPI Baud rate clock is generated by dividing the module clock (CLK_QSPI_AHB) by a value between 1 and 255.

This allows a maximum operating baud rate at up to Host Clock and a minimum operating baud rate of CLK_QSPI_AHB divided by 255.

At reset, BAUD = 0 and the user has to program a valid value before performing the first transfer.

35.6.5 Serial Clock Phase and Polarity

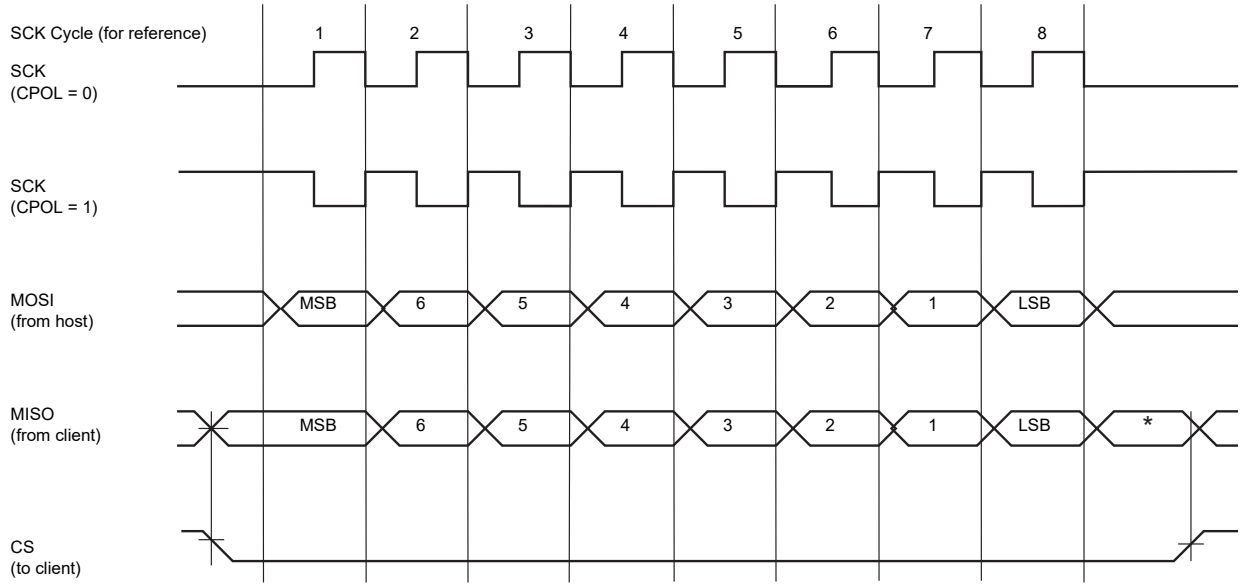
Four combinations of polarity and phase are available for data transfers. Writing the Clock Polarity bit in the QSPI Baud register (BAUD.CPOL) selects the polarity. The Clock Phase bit in the BAUD register programs the clock phase (BAUD.CPHA). These two parameters determine the edges of the clock signal where data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations.
Note: The polarity/phase combinations are incompatible. Thus, the interfaced client must use the same parameter values to communicate.

Table 35-2. SPI Transfer Mode

Clock Mode	BAUD.CPOL	BAUD.CPHA	Shift SCK Edge	Capture SCK Edge	SCK Inactive Level
0	0	0	Falling	Rising	Low
1	0	1	Rising	Falling	Low
2	1	0	Rising	Falling	High
3	1	1	Falling	Rising	High

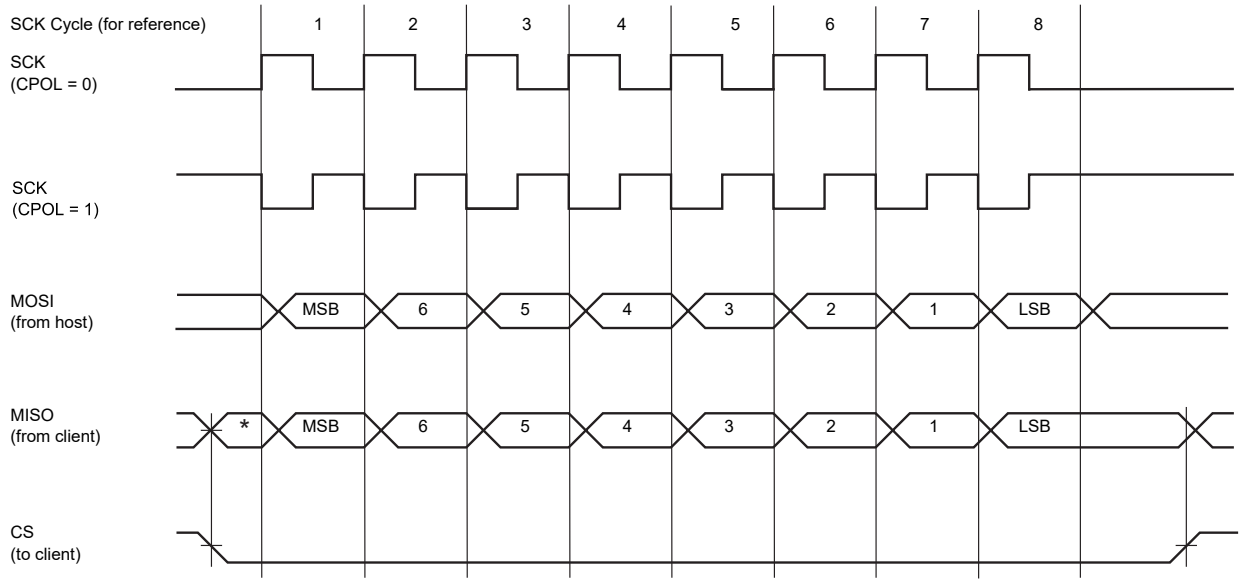
Figure 35-3. QSPI Transfer Modes (BAUD.CPHA = 0, 8-bit transfer)

Figure 35-4.



* Not defined, but normally MSB of previous character received

QSPI Transfer Modes (BAUD.CPHA = 1, 8-bit transfer)



* Not defined, but normally LSB of previous character received

35.6.6 Transfer Delays

The QSPI supports several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

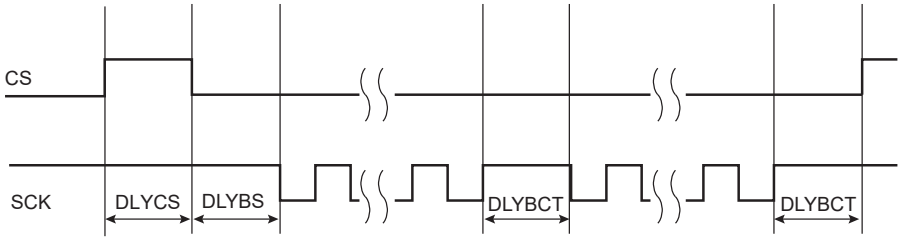
- The delay between the inactivation and the activation of CS is programmed by writing the Minimum Inactive CS Delay bit field in the Control B register (CTRLB.DLYCS), allowing to tune the minimum time of CS at high level.
- The delay between consecutive transfers is programmed by writing the Delay Between Consecutive Transfers bit field in the Control B register (CTRLB.DLYBCT), allowing to insert a delay

between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT settings are ignored.

- The delay before SCK is programmed by writing the Delay Before SCK bit field in the BAUD register (BAUD.DLYBS), allowing to delay the start of SPCK after the chip select has been asserted.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 35-5. Programmable Delay



35.6.7 QSPI SPI Mode

In this mode, the QSPI acts as a regular SPI host.

To activate this mode, the MODE bit in the Control B register must be cleared (CTRLB.MODE = 0).

35.6.7.1 SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the client connected to the SPI bus. The QSPI drives the chip select line to the client (CS) and the serial clock signal (SCK).

The QSPI features a single internal shift register and two holding registers: the Transmit Data Register (TXDATA) and the Receive Data Register (RXDATA). The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the TXDATA. The written data is immediately transferred into the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted into the internal shift register. Receiving data cannot occur without transmitting data.

If new data is written in TXDATA during the transfer, it stays in TXDATA until the current transfer is completed. Then, the received data is transferred from the internal shift register to the RXDATA, the data in TXDATA is loaded into the internal shift register, and a new transfer starts.

The transfer of data written in TXDATA in the internal shift register is indicated by the Transmit Data Register Empty (DRE) bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE). When new data is written in TXDATA, this bit is cleared. The DRE bit is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the Transmission Complete flag (INTFLAG.TXC). If the transfer delay for the last transfer was configured to be greater than 0 (CTRLB.DLYBCT), TXC is set after the completion of the delay. The module clock (CLK_QSPI_AHB) can be switched off at this time.

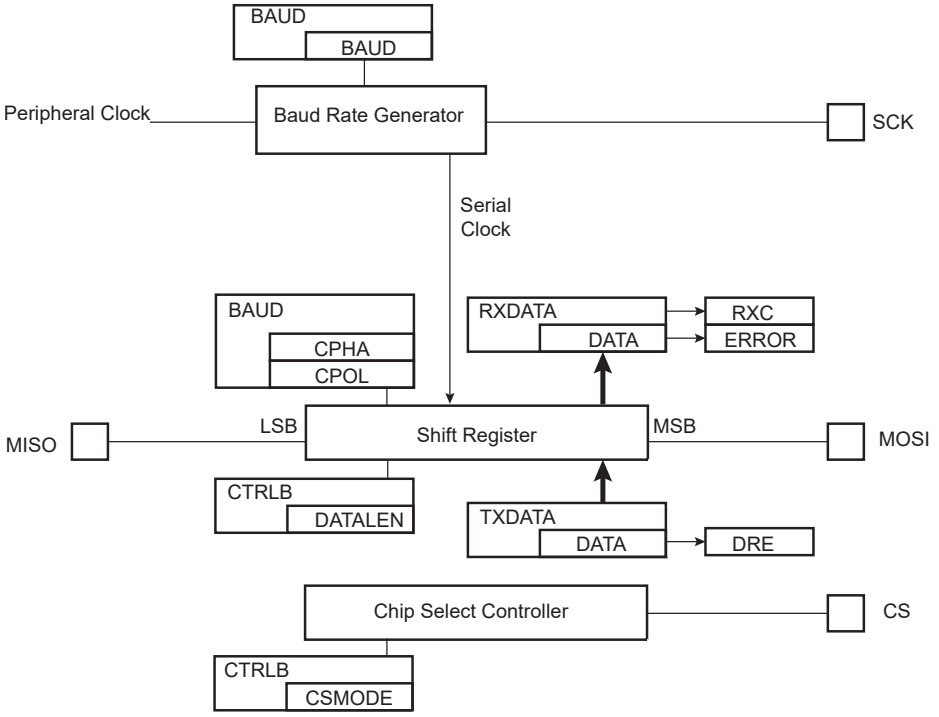
Ongoing transfer of received data from the internal shift register into RXDATA is indicated by the Receive Data Register Full flag (INTFLAG.RXC). When the received data is read, the RXC bit is cleared.

If the RXDATA has not been read before new data is received, the Overrun Error flag in INTFLAG register (INTFLAG.ERROR) is set. As long as this flag is set, data is loaded in RXDATA.

The SPI Mode Block Diagram shows a flow chart describing how transfers are handled.

35.6.7.2 SPI Mode Block Diagram

Figure 35-6. SPI Mode Block Diagram



35.6.7.3 SPI Mode Flow Diagram

Figure 35-7. SPI Mode Flow Diagram

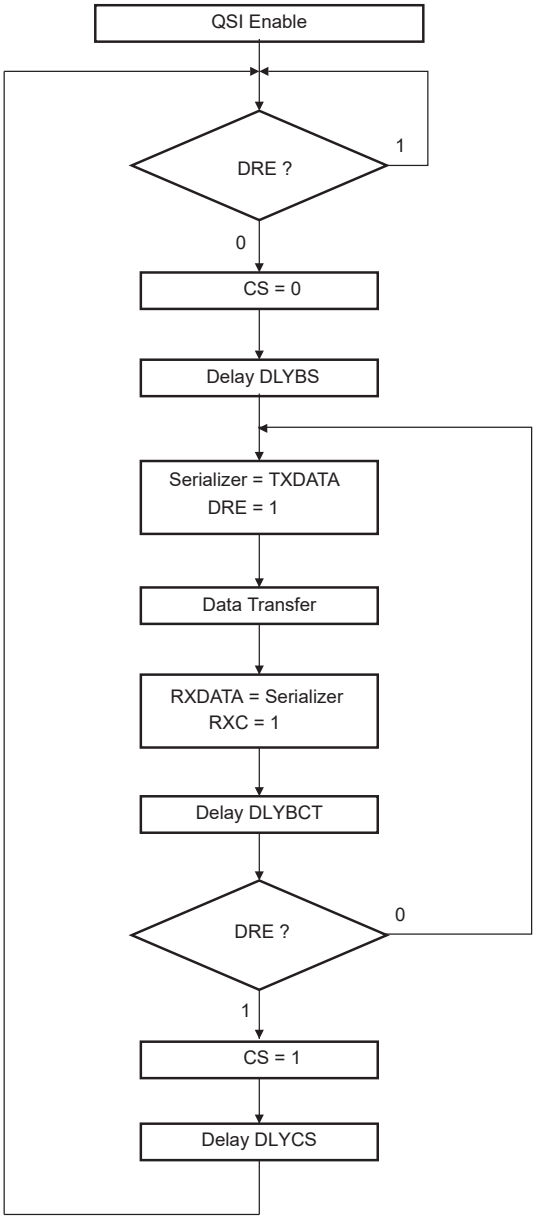
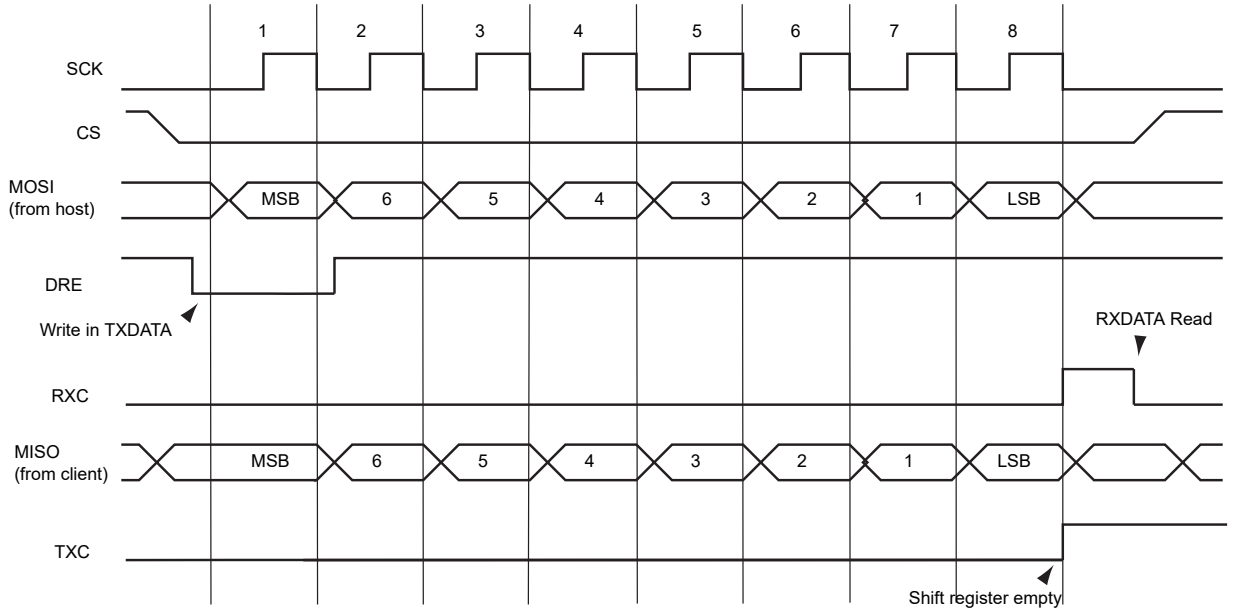


Figure 35-8. Interrupt Flags Behaviour



35.6.7.4 Peripheral Deselection with DMA

When the Direct Memory Access Controller is used, the Chip Select line will remain low during the whole transfer since the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is managed by the DMA itself. The reloading of the TXDATA by the DMA is done as soon as INTFLAG.DRE flag is set. In this case, setting the Chip Select Mode bit field in the Control B register (CTRLB.CSMODE) to 0x1 is not mandatory.

However, it may happen that when other DMA channels connected to other peripherals are in use as well, the QSPI DMA could be delayed by another DMA transfer with a higher priority on the bus. Having DMA buffers in slower memories like flash memory or SDRAM (compared to fast internal SRAM), may lengthen the reload time of the TXDATA by the DMA as well. This means that TXDATA might not be reloaded in time to keep the Chip Select line low. In this case the Chip Select line may toggle between data transfer and according to some SPI Client devices, and the communication might get lost. Writing CTRLB.CSMODE=0x1 can prevent this loss.

When CTRLB.CSMODE=0x0, the CS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the INTFLAG.DRE flag is raised as soon as the content of the TXDATA is transferred into the internal shifter. When this flag is detected the TXDATA can be reloaded. if this reload occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers. This may lead to difficulties for interfacing with some serial peripherals requiring the Chip Select to be de-asserted after each transfer. To facilitate interfacing with such devices, it is recommended to write CTRLB.CSMODE to 0x2.

35.6.7.5 Peripheral Deselection without DMA

During multiple data transfers on a Chip Select without the DMA, the TXDATA is loaded by the processor, and the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) rises as soon as the content of the RXDATA is transferred into the internal shift register. When this flag is detected high, the TXDATA can be reloaded. If this reload-by-processor occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers.

Depending on the application software handling the flags or servicing other interrupts or other tasks, the processor may not reload the TXDATA in time to keep the Chip Select active (low). A null

Delay Between Consecutive Transfer bit field value in the CTRLB register (CTRLB.DLYBCT) will give even less time for the processor to reload the TXDATA. With some SPI Client peripherals, requiring the Chip Select line to remain active (low) during a full set of transfers might lead to communication errors.

To facilitate interfacing with such devices, the Chip Select Mode bit field in the CTRLB register (CTRLB.CSMODE) can be written to 0x1. This allows the Chip Select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer bit in the CTRLA register (CTRLA.LASTXFER). Even if the TXDATA is not reloaded the Chip Select will remain active. To have the Chip Select line rise at the end of the last data transfer, the LASTXFER bit in the CTRLA must be set before writing the last data to transmit into the TXDATA.

35.6.8 QSPI Serial Memory Mode

In this mode, the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock and so on) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual-SPI and Quad-SPI protocols.

To activate this mode, the MODE bit in Control B register must be set to one (CTRLB.MODE = 1).

In serial memory mode, data cannot be transferred by the TXDATA and the RXDATA but by writing or reading the QSPI memory space (0x0400 0000–0x0500 0000).

Caching can be enabled using the CMCC module along with configuring the CFGCON1.QSCHE_EN bit.

Important: The QSPI memory space region can be cached to improve data transfer speed.
However, external Flash devices that have command/status registers mapped in the QSPI memory space region must be managed carefully by applying any one of the following configurations:

- The data cache must be disabled.
- If the data cache is required, then the cache line must be invalidated before reading the status register.

35.6.8.1 Instruction Frame

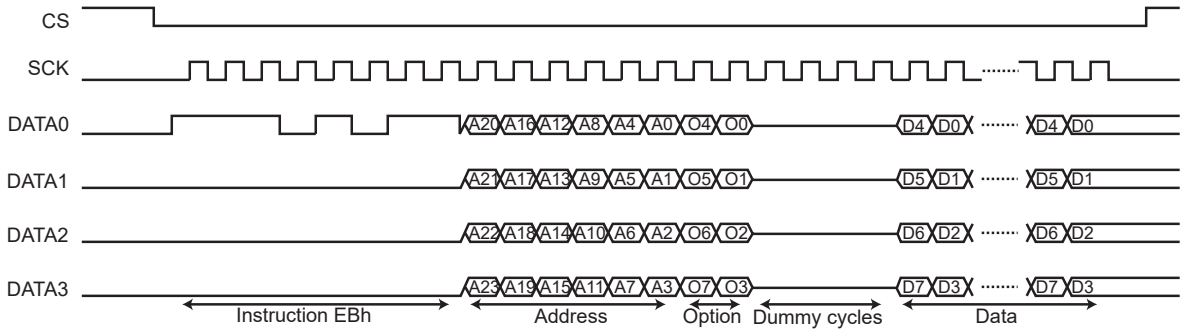
To control serial Flash memories, the QSPI is able to send instructions by the SPI bus (for example, READ, PROGRAM, ERASE, LOCK and so on). The instruction set implemented in serial Flash memories is memory vendor dependent; therefore, the QSPI includes a complete instruction registers, which makes it very flexible and compatible with all serial Flash memories.

An instruction frame includes:

- **An instruction code (size: 8 bits):** The instruction can be optional in some cases.
- **An address (size: 24 bits or 32 bits):** The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default, the address is 24 bits long but it can be 32 bits long to support serial Flash memories larger than 128 Mbit (16 Mbyte).
- **An option code (size: 1/2/4/8 bits):** The option code is optional but is useful for activating the XIP mode or the Continuous Read mode for READ instructions in some serial Flash memory devices. These modes permit data read latency improvement.
- **Dummy cycles:** Dummy cycles are optional but required by some READ instructions.
- **Data bytes are optional:** Data bytes are present for data transfer instructions such as READ or PROGRAM.

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.

Figure 35-9. Instruction Frame



35.6.8.2 Instruction Frame Sending

To send an instruction frame, the user must first configure the address to send by writing the field ADDR in the Instruction Address Register (INSTRADDR.ADDR). This step is required if the instruction frame includes an address and no data. When data are present, the address of the instruction is defined by the address of the data accesses in the QSPI memory space and not by the INSTRADDR register.

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the INST and OPTCODE bit fields in the Instruction Control Register (INSTRCTRL.OPTCODE, INSTRCTRL.INSTR).

Then, the user must write the Instruction Frame Register (INSTRFRAME) to configure the instruction frame depending on which instruction must be sent. If the instruction frame does not include data, writing in this register triggers the send of the instruction frame in the QSPI. If the instruction frame includes data, the send of the instruction frame is triggered by the first data access in the QSPI memory space.

The instruction frame is configured by the following bits and fields of INSTRFRAME:

- The WIDTH field is used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data. It is possible to use two unidirectional data lanes (MISO-MOSI Single-bit SPI), two bidirectional data lanes (DATA0 - DATA1 Dual SPI) or four bidirectional data lanes (DATA0 – DATA3).

Table 35-3. WIDTH Encoding

INSTRFRAME	Instruction	Address/Option	Data
0	Single-bit SPI	Single-bit SPI	Single-bit SPI
1	Single-bit SPI	Single-bit SPI	Dual SPI
2	Single-bit SPI	Single-bit SPI	Quad SPI
3	Single-bit SPI	Dual SPI	Dual SPI
4	Single-bit SPI	Quad SPI	Quad SPI
5	Dual SPI	Dual SPI	Dual SPI
6	Quad SPI	Quad SPI	Quad SPI
7	Reserved		

- The INSTREN bit enables sending an instruction code
- The ADDREN bit enables sending of an address after the instruction code
- The OPTCODEEN bit enables sending of an option code after the address
- The DATAEN bit enables the transfer of data (READ or PROGRAM instruction)

- The OPTCODELEN field configures the option code length (0 -> 1-bit / 1 -> 2-bit / 2 -> 4-bit / 3 -> 8-bit). The value written in OPTCODELEN must be consistent with the value written in the field WIDTH. For example, OPTCODELEN = 0 (1-bit option code) is not coherent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4-bit).
- The ADDRLEN bit configures the address length (0 -> 24 bits / 1-> 32 bits)
- The TFRTYPE field defines which type of data transfer must be performed
- The DUMMYLEN field configures the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.

If data transfer is enabled, the user can access the serial memory by reading or writing the QSPI memory space following these rules:

- Reading from the serial memory, but not memory data (for example reading the JEDEC-ID or the STATUS), requires TFRTYPE to be written to 0x0
- Reading from the serial memory, particularly memory data, requires TFRTYPE to be written to '1'
- Writing to the serial memory, but not memory data (for example, writing the configuration or STATUS), requires TFRTYPE to be written to 0x2
- Writing to the serial memory, particularly memory data, requires TFRTYPE to be written to 0x3

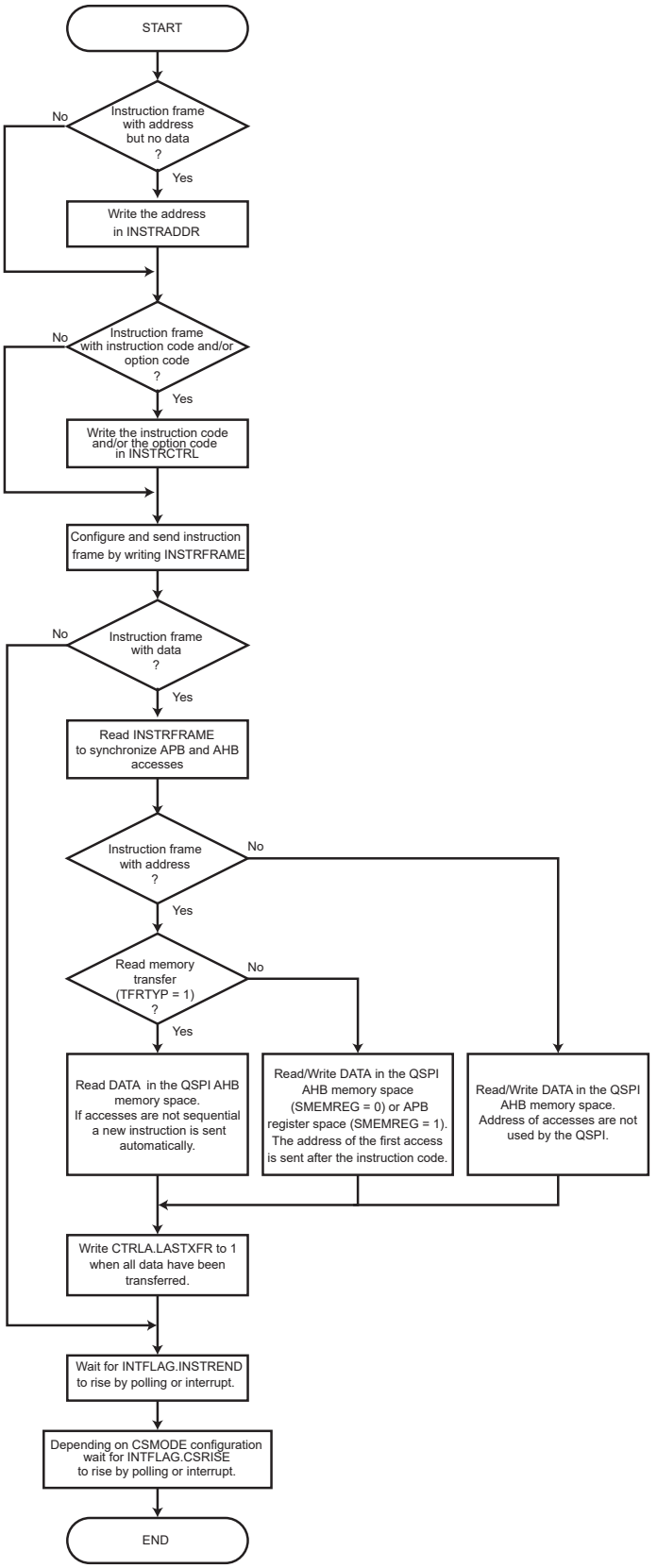
If TFRTYP has a value other than 0x1 and CTRLB.SMEMREG = 0, the address sent in the instruction frame is the address of the first system bus accesses. The addresses of the subsequent access actions are not used by the QSPI. At each system bus access, an SPI transfer is performed with the same size. For example, a half-word system bus access leads to a 16-bit SPI transfer and a byte system bus access leads to an 8-bit SPI transfer.

If CTRLB.SMEMREG = 1, accesses are made via the QSPI registers and the address sent in the instruction frame is the address defined in the INSTRADDR register. Each time the INSTRFRAME or TXDATA registers are written, an SPI transfer is performed with a byte size. Another byte is read each time the RXDATA register is read or written each time the TXDATA register is written. The SPI transfer ends by writing the LASTXFER bit in the Control A register (CTRLA.LASTXFER).

If TFRTYP = 0x1, the address of the first instruction frame is one of the first read access in the QSPI memory space. Each time the read accesses becomes non-sequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus read accesses.

When data transfer is not enabled, the end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set. When data transfer is enabled, the user must indicate when the data transfer is complete in the QSPI memory space by setting the bit LASTXFR in the CTRLA. The end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set.

Figure 35-10. Instruction Transmission Flow Diagram



35.6.8.3 Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with DATAEN = 1 and TFRTYP = 0x1 in the Instruction Frame register (INSTRFRAME).

In this mode, the QSPI is able to read data at a random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

To fetch data, the user must first configure the instruction frame by writing the INSTRFRAME. Then, data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses matches the address of the data inside the serial Flash memory.

When Fetch mode is enabled, several instruction frames can be sent before writing the bit LASTXFR in the CTRLA. Each time the system bus read accesses become non-sequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

35.6.8.4 Continuous Read Mode

The QSPI is compatible with Continuous Read Mode (CRM), which is implemented in some Serial Flash memories.

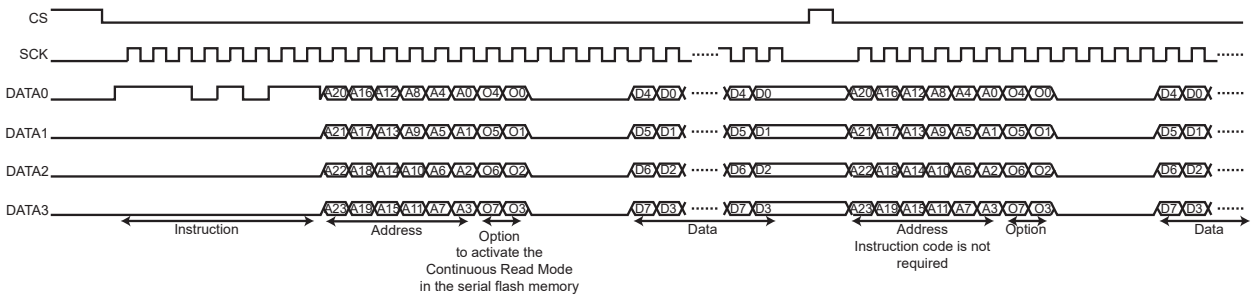
The CRM provides reduction in instruction overhead by excluding the instruction code from the instruction frame. When CRM is activated in a Serial Flash memory (by a specific option code), the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required, as the memory uses the stored one.

In the QSPI, CRM is used when reading data from the memory (INSTFRAME.TFRTYPE = 0x1). The addresses of the system bus read accesses are often non-sequential; this leads to many instruction frames with the same instruction code. By disabling the sending of the instruction code, the CRM reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the Serial Flash memory. The CRM is enabled in the QSPI by setting the CRM bit in the INSTRFRAME register (INSTFRAME.CRMODE = 1, INSTFRAME.TFRTYPE must be 0x1). The CRM is enabled in the Serial Flash memory by sending a specific option code.

CAUTION If the CRM is not supported by the Serial Flash memory or disabled, the CRMODE bit must not be set. Otherwise, data read out from the Serial Flash memory is not valid.

Figure 35-11. Continuous Read Mode



35.6.8.5 Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (BAUD.CPOL = 0 and BAUD.CPHA = 0). All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

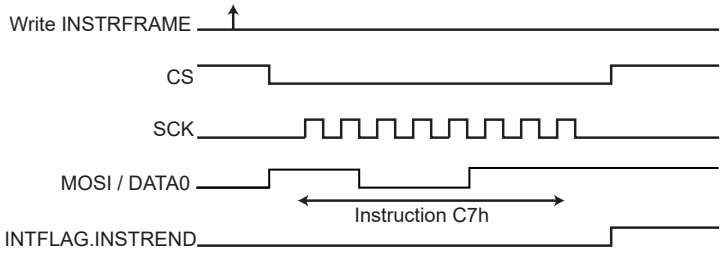
Example 35-1. Scenario 1

Instruction in Single-bit SPI, without address, without option, without data.

Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 to INSTRCTRL register.
- Write 0x0000_0010 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-12. Instruction Transmission Waveform 1



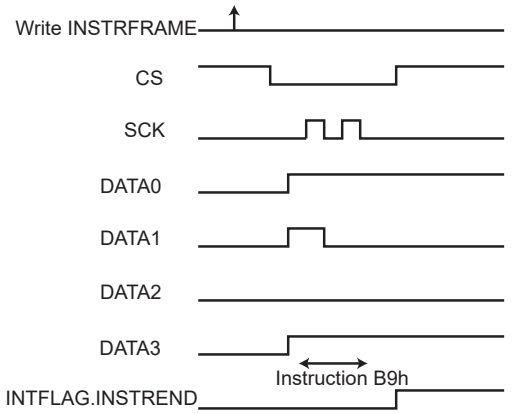
Example 35-2. Scenario 2

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 to INSTRCTRL register.
- Write 0x0000_0016 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-13. Instruction Transmission Waveform 2



Example 35-3. Scenario 3

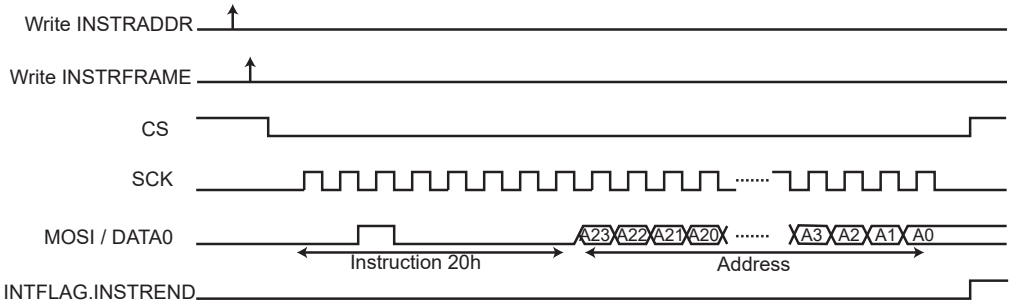
Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) to QSPI_AR.
- Write 0x0000_0020 to INSTRCTRL register.

- Write 0x0000_0030 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-14. Instruction Transmission Waveform 3



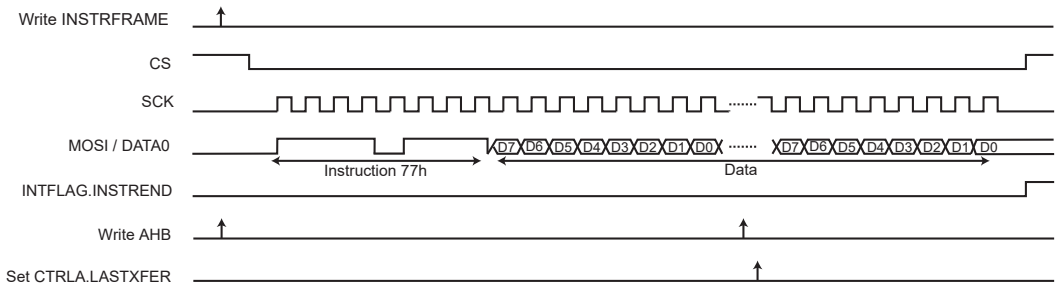
Example 35-4. Scenario 4

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 to INSTRCTRL register.
- Write 0x0000_2090 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Write data to the system bus memory space (0x0400_0000–0x0500_0000). The address of the system bus write accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-15. Instruction Transmission Waveform 4



Example 35-5. Scenario 5

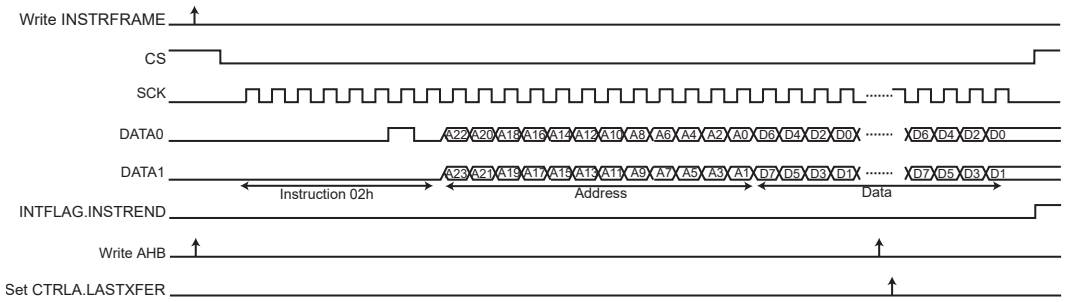
Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 to INSTRCTRL register.
- Write 0x0000_30B3 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.

- Write data to the QSPI system bus memory space (0x040_00000–0x0500_0000). The address of the first system bus write access is sent in the instruction frame. The address of the next system bus write accesses is not used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-16. Instruction Transmission Waveform 5



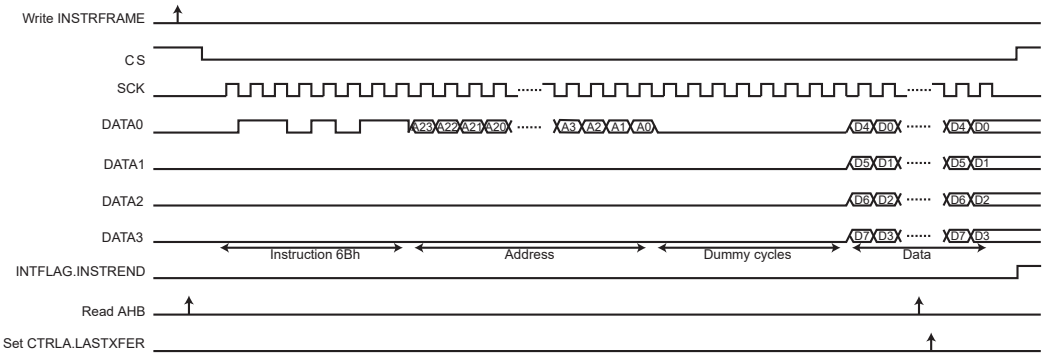
Example 35-6. Scenario 6

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B to INSTRCTRL register.
- Write 0x0008_10B2 to INSTRFRAME register.
- Read QSPI_IR (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040_00000–0x0500_0000). The address of the first system bus read access is sent in the instruction frame. The address of the next system bus read accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-17. Instruction Transmission Waveform 6



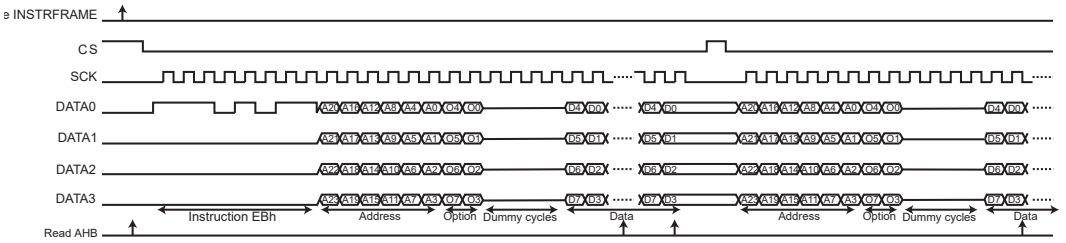
Example 35-7. Scenario 7

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read from Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB to INSTRCTRL register.
- Write 0x0004_33F4 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040_00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-18. Instruction Transmission Waveform 7



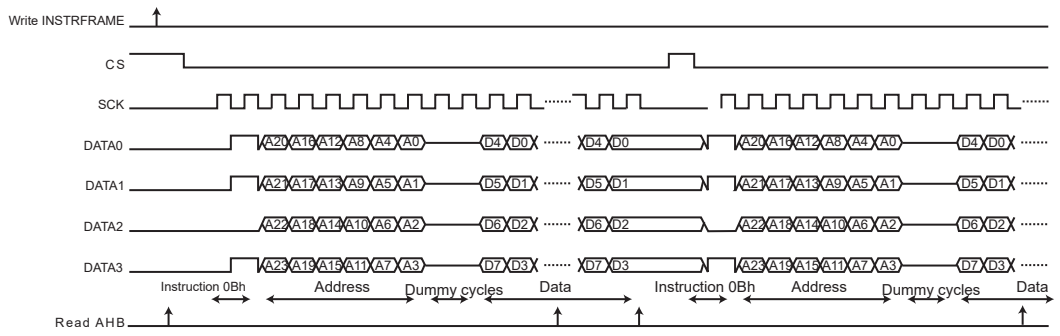
Example 35-8. Scenario 8

Instruction in Quad SPI, with address in Quad SPI, without option, with data read from Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B to INSTRCTRL register.
- Write 0x0002_20B6 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x040_00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-19. Instruction Transmission Waveform 8



35.6.9 Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data are scrambled when written to memory and unscrambled when data are read.

The external data lines can be scrambled to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the micro-controller or the QSPI client device (for example, memory).

The scrambling/unscrambling function can be enabled by writing a '1' to the ENABLE bit in the Scrambling Control register (SCRAMBCTRL.ENABLE).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable Scrambling User Key in the Scrambling Key register (SCRAMBKEY.KEY). This register is only accessible in the Write mode.

By default, the scrambling and unscrambling algorithm includes the scrambling user key, plus a device-dependent random value. This random value is not included when the Scrambling/Unscrambling Random Value Disable bit in the Scrambling Mode register (SCRAMBCTRL.RANDOMDIS) is written to '1'.

The random value is neither user-configurable nor readable. If SCRAMBCTRL.RANDOMDIS = 0, data scrambled by a given circuit cannot be unscrambled by a different circuit.

If SCRAMBCTRL.RANDOMDIS = 1, the scrambling/unscrambling algorithm includes only the scrambling user key, making it possible to manage data by different circuits.

The scrambling user key must be securely stored in a reliable Non-Volatile Memory to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

35.6.10 DMA Operation

The QSPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the RXDATA register, and cleared when RXDATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TXDATA) is empty, and cleared when TXDATA is written.

Note: If DMA and RX memory modes are selected, a QSPI memory space read operation is required to force the first triggering.

If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted.

35.6.11 Interrupts

The QSPI has the following interrupt source:

- Interrupt Request (INTREQ) – Indicates that at least one bit in the Interrupt Flag Status and Clear register (INTFLAG) is set to '1'

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 interrupt flag is cleared, the interrupt is disabled or the QSPI 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.

35.7 Register Summary

See the *QSPI* 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	CTRLA	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								LASTXFER
0x04	CTRLB	7:0			CSMODE[1:0]		SMEMREG	WDRBT	LOOPEN	MODE
		15:8					DATALEN[3:0]			
		23:16	DLYBCT[7:0]							
		31:24	DLYCS[7:0]							
0x08	BAUD	7:0							CPHA	CPOL
		15:8	BAUD[7:0]							
		23:16	DLYBS[7:0]							
		31:24								
0x0C	RXDATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16								
		31:24								
0x10	TXDATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16								
		31:24								
0x14	INTENCLR	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x18	INTENSET	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x1C	INTFLAG	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x20	STATUS	7:0							ENABLE	
		15:8							CSSTATUS	
		23:16								
		31:24								
0x24 ... 0x2F	Reserved									
0x30	INSTRADDR	7:0	ADDR[7:0]							
		15:8	ADDR[15:8]							
		23:16	ADDR[23:16]							
		31:24	ADDR[31:24]							
0x34	INSTRCTRL	7:0	INSTR[7:0]							
		15:8								
		23:16	OPTCODE[7:0]							
		31:24								
0x38	INSTRFRAME	7:0	DATAEN	OPTCODEEN	ADDREN	INSTREN		WIDTH[2:0]		
		15:8	DDREN	CRMODE	TFRTYPE[1:0]			ADDRLEN	OPTCODELEN[1:0]	
		23:16				DUMMYLEN[4:0]				
		31:24								
0x3C ... 0x3F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	SCRAMBCTRL	7:0							RANDOMDIS	ENABLE
		15:8								
		23:16								
		31:24								
0x44	SCRAMBKEY	7:0	KEY[7:0]							
		15:8	KEY[15:8]							
		23:16	KEY[23:16]							
		31:24	KEY[31:24]							

Related Links

8. Product Memory Mapping Overview

35.8 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.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

See *Peripheral Access Controller (PAC)* from Related Links.

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

Related Links

24. Peripheral Access Controller (PAC)

35.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								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
							ENABLE	SWRST
Access							W	W
Reset							0	0

Bit 24 – LASTXFER Last Transfer

Value	Description
0	No effect.
1	The chip select is de-asserted after the character written in TD has been transferred.

Bit 1 – ENABLE Enable

Writing a '0' to this bit disables the QSPI.
Writing a '1' to this bit enables the QSPI to transfer and receive data.
As soon as ENABLE is reset, QSPI finishes its transfer.
All pins are set in Input mode and no data are received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the QSPI. A software-triggered hardware Reset of the QSPI interface is performed.
DMAC channels are not affected by software Reset.

35.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DLYCS[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	23	22	21	20	19	18	17	16
	DLYBCT[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	15	14	13	12	11	10	9	8
					DATALEN[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			CSMODE[1:0]		SMEMREG	WDRBT	LOOPEN	MODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 31:24 – DLYCS[7:0] Minimum Inactive CS Delay
This bit field defines the minimum delay between the inactivation and the activation of CS. The DLYCS time guarantees the client minimum deselect time.
If DLYCS is 0x00, one CLK_QSPI_AHB period is inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × fperipheral clock

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0x00, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers. In Serial Memory mode (MODE = 1), DLYBCT is ignored and no delay is inserted. Otherwise, the following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × fperipheral clock)/32

Bits 11:8 – DATALEN[3:0] Data Length
The DATALEN field determines the number of data bits transferred. Reserved values must not be used.

Value	Name	Description
0x0	8BITS	8 bits transfer
0x1	9BITS	9 bits transfer
0x2	10BITS	10 bits transfer
0x3	11BITS	11 bits transfer
0x4	12BITS	12 bits transfer
0x5	13BITS	13 bits transfer
0x6	14BITS	14 bits transfer

Value	Name	Description
0x7	15BITS	15 bits transfer
0x8	16BITS	16 bits transfer
0x9-0xF	—	Reserved

Bits 5:4 – CSMODE[1:0] Chip Select Mode

The CSMODE field determines how the chip select is de-asserted.

Value	Name	Description
0x0	NORELOAD	The chip select is de-asserted if TD was not reloaded before the end of the current transfer.
0x1	LASTXFER	The chip select is de-asserted when the bit LASTXFER is written at '1' and the character written in TD was transferred.
0x2	SYSTEMATICALLY	The chip select is de-asserted systematically after each transfer.
0x3	—	Reserved

Bit 3 – SMEMREG Serial Memory Register Mode

Value	Description
0	Serial memory registers are written via AHB access.
1	Serial memory registers are written via APB access. Reset the QSPI.

Bit 2 – WDRBT Wait Data Read Before Transfer

This bit determines the Wait Data Read Before Transfer option.

Bit 1 – LOOPEN Local Loopback Enable

This bit defines if the Local Loopback is enabled or disabled.

LOOPEN controls the local loopback on the data serializer for testing in SPI Mode only. (MISO is internally connected on MOSI).

Value	Description
0	Local Loopback is disabled.
1	Local Loopback is enabled.

Bit 0 – MODE Serial Memory Mode

This bit defines if the QSPI is in SPI mode or Serial Memory mode.

Value	Name	Description
0	SPI	SPI operating mode
1	MEMORY	Serial Memory operating mode

35.8.3 Baud Rate

Name: BAUD
Offset: 0x08
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
	DLYBS[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	15	14	13	12	11	10	9	8
	BAUD[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
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before SCK
This field defines the delay from CS valid to the first valid SCK transition.
When DLYBS equals zero, the CS valid to SCK transition is 1/2 the SCK clock period.
Otherwise, the following equation determines the delay:

Equation 35-1. Delay Before SCK

$$Delay\ \ Before\ SCK = \frac{DLYBS}{MCK}$$

Bits 15:8 – BAUD[7:0] Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the SCK baud rate from the module clock (MCK) CLK_QSPI_AHB. The Baud rate is selected by writing a value from 1 to 255 in the BAUD field. The following equation determines the SCK baud rate:

Equation 35-2. SCK Baud Rate

$$SCK\ \ Baud\ \ Rate = \frac{MCK}{(BAUD + 1)}$$

Bit 1 – CPHA Clock Phase
CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data are captured on the leading edge of SCK and changed on the following edge of SCK.
1	Data are changed on the leading edge of SCK and captured on the following edge of SCK.

Bit 0 – CPOL Clock Polarity

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with CPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of SCK is logic level '0'.
0	The inactive state value of SCK is logic level '1'.

35.8.4 Receive Data

Name: RXDATA
Offset: 0x0C
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
	DATA[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
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Receive Data
Data received by the QSPI is stored in this register right-justified. Unused bits read zero.

35.8.5 Transmit Data

Name: TXDATA
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
	DATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Transmit Data
Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

35.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The INSTREND interrupt is disabled.
1	The INSTREND interrupt is enabled.

Bit 8 – CSRISE Chip Select Rise Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The CSRISE interrupt is disabled.
1	The CSRISE interrupt is enabled.

Bit 3 – ERROR Overrun Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The ERROR interrupt is disabled.
1	The ERROR interrupt is enabled.

Bit 2 – TXC Transmission Complete Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The TXC interrupt is disabled.
1	The TXC interrupt is enabled.

Bit 1 – DRE Transmit Data Register Empty Interrupt Disable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The DRE interrupt is disabled.
1	The DRE interrupt is enabled.

Bit 0 – RXC Receive Data Register Full Interrupt Disable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

Value	Description
0	The RXC interrupt is disabled.
1	The RXC interrupt is enabled.

35.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x18
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The INSTREND interrupt is disabled.
1	The INSTREND interrupt is enabled.

Bit 8 – CSRISE Chip Select Rise Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The CSRISE interrupt is disabled.
1	The CSRISE interrupt is enabled.

Bit 3 – ERROR Overrun Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The ERROR interrupt is disabled.
1	The ERROR interrupt is enabled.

Bit 2 – TXC Transmission Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The TXC interrupt is disabled.
1	The TXC interrupt is enabled.

Bit 1 – DRE Transmit Data Register Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The DRE interrupt is disabled.
1	The DRE interrupt is enabled.

Bit 0 – RXC Receive Data Register Full Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

Value	Description
0	The RXC interrupt is disabled.
1	The RXC interrupt is enabled.

35.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1C
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End
This bit is set when an Instruction End was detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 8 – CSRISE Chip Select Rise
The bit is set when a Chip Select Rise was detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 3 – ERROR Overrun Error
This bit is set when an ERROR has occurred.
An ERROR occurs when RXDATA is loaded at least twice from the serializer.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 2 – TXC Transmission Complete

Value	Description
0	As soon as data are written in TXDATA.
1	TXDATA and internal shifter are empty. If a transfer delay was defined, TXC is set after the completion of such delay.

Bit 1 – DRE Transmit Data Register Empty
This bit is '0' when the QSPI is disabled or at Reset.
The bit is set as soon as ENABLE bit is set.

Value	Description
0	Data were written to TXDATA and not yet transferred to the serializer.

Value	Description
1	The last data written in the TXDATA were transferred to the serializer.

Bit 0 – RXC Receive Data Register Full

Value	Description
0	No data were received since the last read of RXDATA.
1	Data were received and the received data were transferred from the serializer to RXDATA since the last read of RXDATA.

35.8.9 Status

Name: STATUS
Offset: 0x20
Reset: 0x00000200
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
							CSSTATUS	
Access							R	
Reset							1	

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

Bit 9 – CSSTATUS Chip Select

Value	Description
0	Chip Select is asserted.
1	Chip Select is not asserted.

Bit 1 – ENABLE Enable

Value	Description
0	QSPI is disabled.
1	QSPI is enabled.

35.8.10 Instruction Address

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

Bit	31	30	29	28	27	26	25	24
	ADDR[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
	ADDR[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
	ADDR[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
	ADDR[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 – ADDR[31:0] Instruction Address
Address to send to the serial flash memory in the instruction frame.

35.8.11 Instruction Code

Name: INSTRCTRL
Offset: 0x34
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	OPTCODE[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	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTR[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 23:16 – OPTCODE[7:0] Option Code
These bits define the option code to send to the serial Flash memory.

Bits 7:0 – INSTR[7:0] Instruction Code
Instruction code to send to the serial Flash memory.

35.8.12 Instruction Frame

Name: INSTRFRAME
Offset: 0x38
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				DUMMYLEN[4:0]				
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
	DDREN	CRMODE	TFRTYPE[1:0]			ADDRLEN	OPTCODELEN[1:0]	
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	6	5	4	3	2	1	0
	DATAEN	OPTCODEEN	ADDREN	INSTREN		WIDTH[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

Bits 20:16 – DUMMYLEN[4:0] Dummy Cycles Length
The DUMMYLEN field defines the number of dummy cycles required by the serial Flash memory before data transfer.

Bit 15 – DDREN Double Data Rate Enable
Note: Double data rate operating is only supported in Read.

Value	Description
0	Double Data Rate operating mode is disabled.
1	Double Data Rate operating mode is enabled.

Bit 14 – CRMODE Continuous Read Mode
This bit defines if the Continuous Read mode is enabled or disabled.

Value	Description
0	Continuous Read Mode is disabled.
1	Continuous Read Mode is enabled.

Bits 13:12 – TFRTYPE[1:0] Data Transfer Type
These bits define the data type transfer.

Value	Name	Description
0x0	READ	Read transfer from the serial memory. Scrambling is not performed. Read at random location (fetch) in the serial Flash memory is not possible.
0x1	READMEMORY	Read data transfer from the serial memory. If enabled, scrambling is performed. Read at random location (fetch) in the serial Flash memory is possible.
0x2	WRITE	Write transfer into the serial memory. Scrambling is not performed.
0x3	WRITEMEMORY	Write data transfer into the serial memory. If enabled, scrambling is performed.

Bit 10 – ADDRLEN Address Length

The ADDRLEN bit determines the length of the address.

Value	Name	Description
0x0	24BITS	24-bit address length
0x1	32BITS	32-bit address length

Bits 9:8 – OPTCODELEN[1:0] Option Code Length

The OPTCODELEN field determines the length of the option code. The value written in OPTCODELEN must be coherent with value written in the field WIDTH. For example, OPTCODELEN = 0 (1-bit option code) is not coherent with WIDTH = 6 (option code sent with QuadSPI protocol, thus, the minimum length of the option code is 4 bits).

Value	Name	Description
0x0	1BIT	1-bit length option code
0x1	2BITS	2-bit length option code
0x2	4BITS	4-bit length option code
0x3	8BITS	8-bit length option code

Bit 7 – DATAEN Data Enable

Value	Description
0	No data are sent/received to/from the serial Flash memory.
1	Data are sent/received to/from the serial Flash memory.

Bit 6 – OPTCODEEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTREN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 2:0 – WIDTH[2:0] Instruction Code, Address, Option Code and Data Width

This field defines the width of the instruction code, the address, the option and the data.

Value	Name	Description
0x0	SINGLE_BIT_SPI	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Single-bit SPI
0x1	DUAL_OUTPUT	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Dual SPI
0x2	QUAD_OUTPUT	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Quad SPI
0x3	DUAL_IO	Instruction: Single-bit SPI/Address-Option: Dual SPI/Data: Dual SPI
0x4	QUAD_IO	Instruction: Single-bit SPI/Address-Option: Quad SPI/Data: Quad SPI
0x5	DUAL_CMD	Instruction: Dual SPI/Address-Option: Dual SPI/Data: Dual SPI
0x6	QUAD_CMD	Instruction: Quad SPI/Address-Option: Quad SPI/Data: Quad SPI
0x7	—	Reserved

35.8.13 Scrambling Mode

Name: SCRAMBCTRL
Offset: 0x40
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RANDOMDIS	ENABLE
Access							R/W	R/W
Reset							0	0

Bit 1 – RANDOMDIS Scrambling/Unscrambling Random Value Disable

Value	Description
0	The scrambling/unscrambling algorithm includes the scrambling user key plus a random value that may differ from chip to chip.
1	The scrambling/unscrambling algorithm includes only the scrambling user key.

Bit 0 – ENABLE Scrambling/Unscrambling Enable

This bit defines if the scrambling/unscrambling is enabled or disabled.

Value	Description
0	Scrambling/unscrambling is disabled.
1	Scrambling/unscrambling is enabled.

35.8.14 Scrambling Key

Name: SCRAMBKEY
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	KEY[31:24]							
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
	KEY[23:16]							
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
	KEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY[31:0] Scrambling User Key
This field defines the user key value.

36. Analog-to-Digital Converter (ADC)

36.1 Overview

The PIC32CX-BZ3 has one shared ADC module. This ADC module incorporates a multiplexer on the input to facilitate a group of inputs and provides a flexible automated scanning option through the input scan logic.

For the ADC module, the analog inputs are connected to the Sample and Hold (S&H) capacitor. The ADC module performs the conversion of the input analog signal based on the configurations set in the registers. When the conversion is complete, the final result is stored in the result buffer for the specific analog input and is passed to the digital filter and digital comparator if configured to use data from this particular sample.

36.2 Features

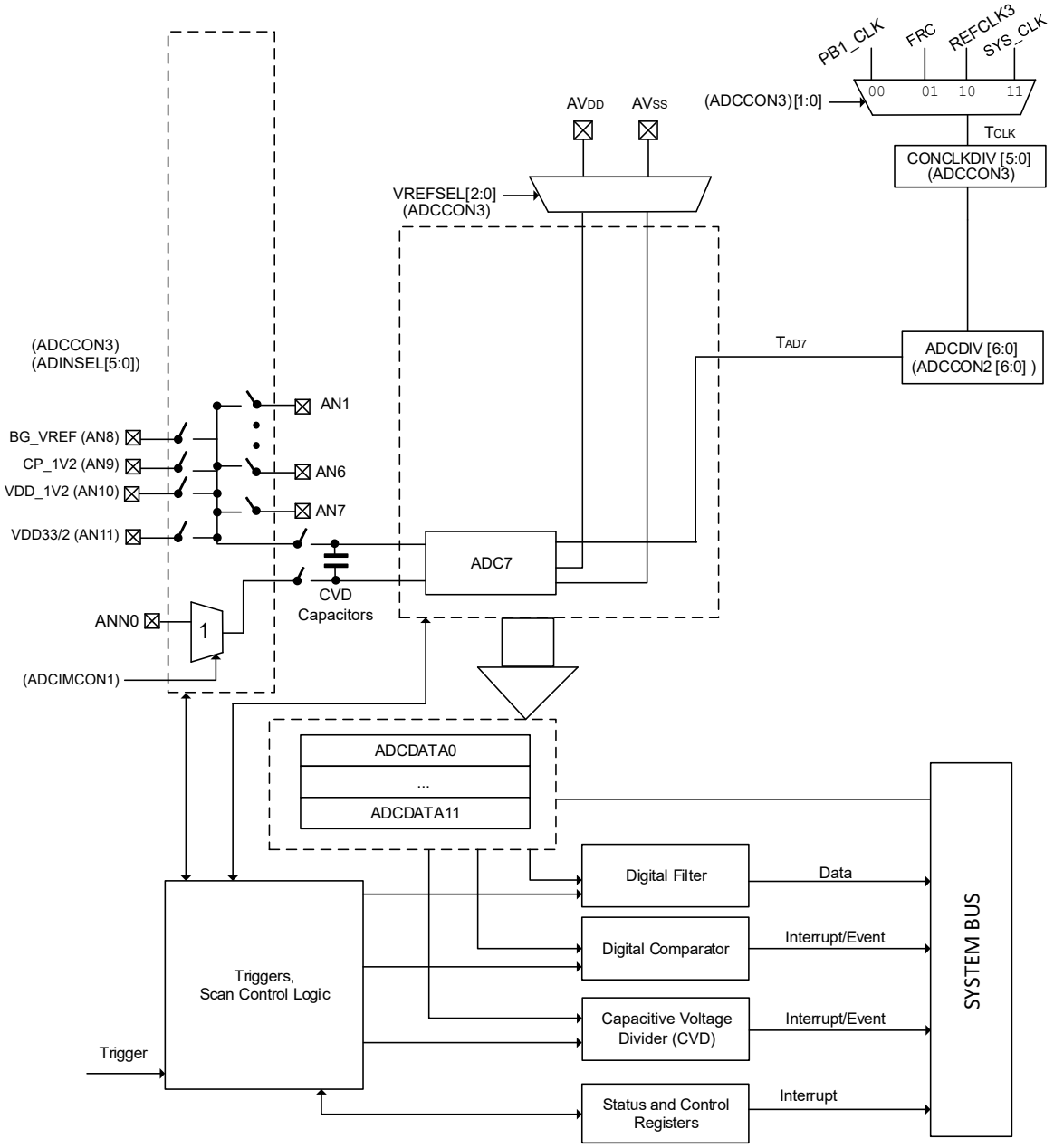
The PIC32CX-BZ3 12-bit High Speed Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) includes the following features:

- 12-Bit Resolution
- One ADC Module, Up to 2 Msps Conversion Rate
- Single-Ended and/or Differential Input
- Supported in Standby Sleep Mode
- Two Digital Comparators
- Two Digital Filters Supporting Two Modes:
  - Oversampling mode
  - Averaging mode
- Designed for Motor Control, Power Conversion and General Purpose Applications

36.3 Block Diagram

A block diagram of the ADC module is illustrated in the following figure.

Figure 36-1. ADC Block Diagram



36.4 ADC Operation

The High Speed Successive Approximation Register (SAR) ADC is designed to support power conversion and motor control applications and consists of one shared ADC module. The shared ADC module has multiple analog inputs connected to its S&H circuit through a multiplexer. Multiple analog inputs share this ADC; therefore, it is termed the shared ADC module. The shared ADC module is used to measure analog signals of lower frequencies and signals that are static in nature (in other words, do not change significantly with time). However, this ADC module is capable of up to 2 Msps sample rate.

The analog inputs connected to the shared ADC module are Class 2 and Class 3 inputs. The number of inputs designated for each class depends on the specific device. For the PIC32CX-BZ3, the following arrangement is provided.

- Class 2 = AN0 to AN7
- Class 3 = AN8 to AN11

The property of each class of analog input is described in the following table.

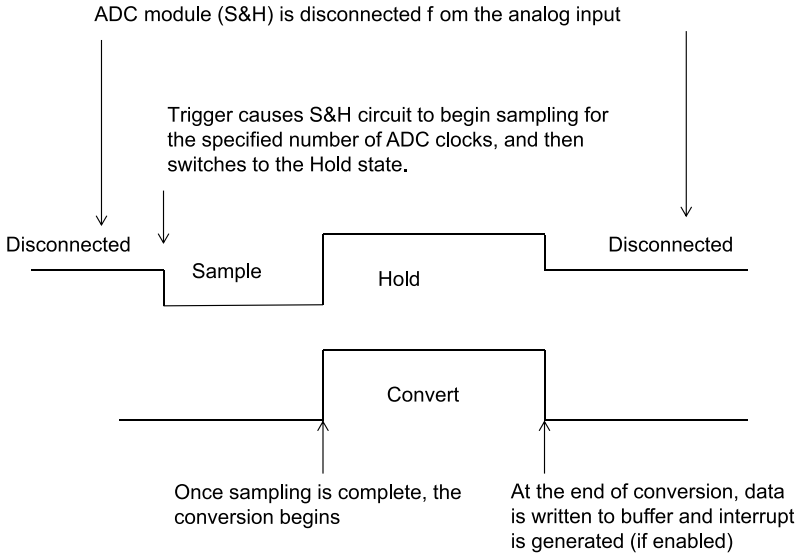
Table 36-1. Analog Input Class

ADC Module	Analog Input Class	Trigger	Trigger Action
Shared ADC module	Class 2	Individual trigger source or scan trigger	Starts sampling sequence or begins scan sequence
Shared ADC module with input scan	Class 3	Scan trigger	Starts scan sequence

Class 2 and Class 3 analog input properties:

- Class 2 inputs are used on the shared ADC module, either individually triggered or as part of a scan list. When used individually, they are triggered by their unique trigger selected by the ADCTRGx register.
- The analog inputs on the shared ADC have a natural order of priority (for example, AN6 has a higher priority than AN7).
- Class 3 inputs are used exclusively for scanning and share a common trigger source (scan trigger).
- Class 3 analog inputs share both the ADC module and the trigger source; therefore, the only method possible to convert them is to scan them sequentially for each incoming scan trigger event, where scanning occurs in the natural order of priority.
- The arrival of a trigger in the shared ADC module only starts the sampling. When the trigger arrives, the ADC module goes into the Sampling mode for the sampling time decided by the SAMC[9:0] bits (ADCCON2[25:16]). At the end of sampling, the ADC starts conversion. Upon completion of conversion, the ADC module is used to convert the next in line Class 2 or Class 3 inputs according to the natural order of priority. When a shared analog input (Class 2 or Class 3) has completed all conversion and no trigger is pending, the ADC module is disconnected from all analog inputs

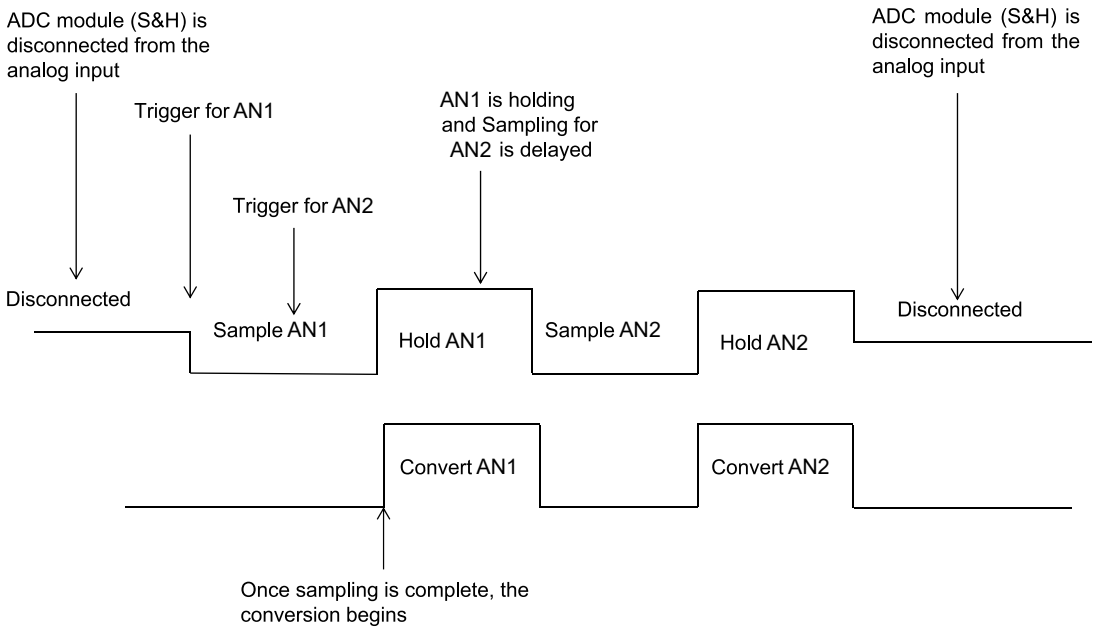
Figure 36-2. Sample and Conversion Sequence for Shared ADC Modules



36.4.1 Class 2 Triggering

When a single Class 2 input is triggered, it is sampled and converted by the shared S&H using the sequence illustrated in Sample and Conversion Sequence for the Shared ADC Modules figure; see *Sample and Conversion Sequence for Shared ADC Modules* figure in the *ADC Operation* from Related Links. When multiple Class 2 inputs are triggered, it is important to understand the consequences of trigger timing. If a conversion is underway and another Class 2 trigger occurs, then the sample-hold-conversion for the new trigger is stalled until the in-process, sample-hold cycle is complete, as shown in the following figure.

Figure 36-3. Multiple Independent Class 2 Trigger Conversion Sequence



When multiple inputs to the shared S&H are triggered simultaneously, the processing order is determined by their natural priority (the lowest numbered input has the highest priority). As an example, if AN1, AN2 and AN3 are triggered simultaneously, AN1 is sampled and converted first, followed by AN2 and finally, AN3. When using the independent Class 2 triggering on the shared S&H, the SAMC[9:0] bits (ADCCON2[25:16]) determine the sample time for all inputs while the appropriate TRGSRC[4:0] bits in the ADCTRGx Register (see *ADCTRG1* register from Related Links) determine the trigger source for each input.

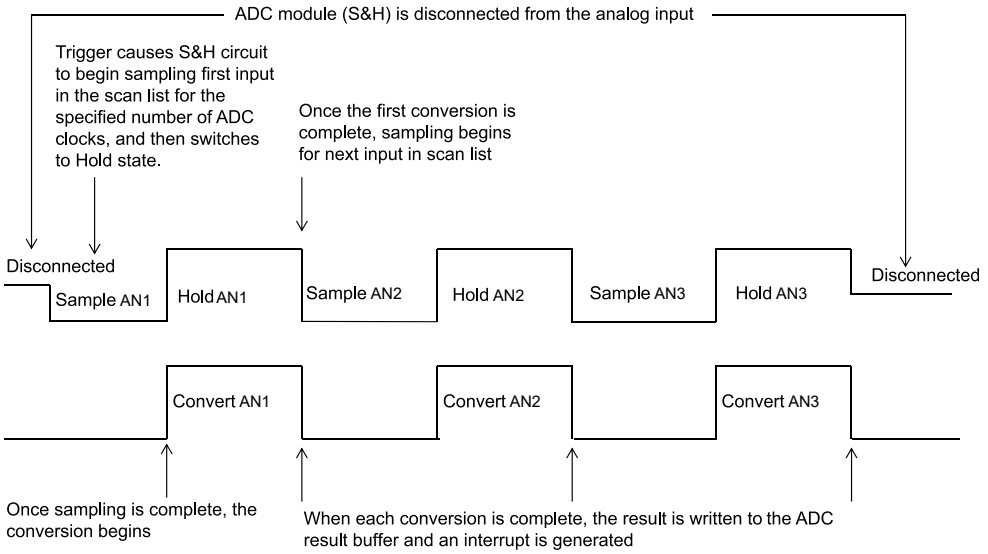
Related Links

36.4. ADC Operation
36.13.14. ADCTRG1

36.4.2 Input Scan

Input scanning is a feature that allows an automated scanning sequence of multiple Class 2 or Class 3 inputs. All Class 2 and Class 3 inputs are scanned using the single shared S&H. The selection of analog inputs for scanning is done with the CSSx bits of the ADCCSS1 registers. Class 2 inputs are triggered using STRIG selection in the ADCTRGx register, and Class 3 inputs are triggered using the STRGSRC[4:0] of the ADCCON1[20:16] register. When a trigger occurs for Class 2 or Class 3 inputs, the sampling and conversion occur in the natural input order is used; lower number inputs are sampled before higher number inputs.

Figure 36-4. Input Scan Conversion Sequence for Three Class 2 Inputs

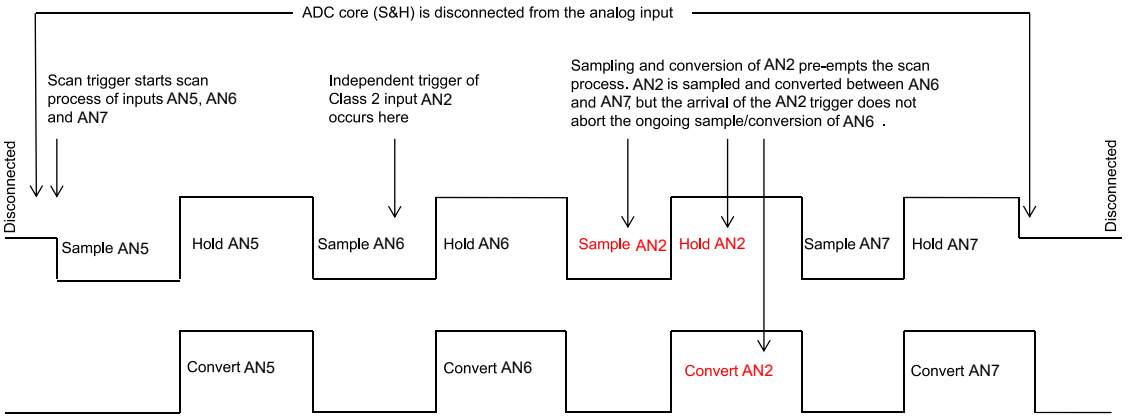


When using the shared analog inputs in scan mode, the SAMC[9:0] bits in the ADC Control Register 2 (ADCCON2[25:16]) determine the sample time for all inputs, while the Scan Trigger Source Selection bits (STRGSRC[4:0]) in the ADC Control Register 1 (ADCCON1[20:16]) determine the trigger source.

To ensure predictable results, a scan must not be retriggered until a sampling of all inputs is complete. Ensure system design to preclude retriggering a scan while a scan is in progress.

Individual Class 2 triggers that occur during a scan preempts the scan sequence if they are a higher priority than the sample currently being processed. In the following figure, a scan of AN5, AN6 and AN7 is underway when an independent trigger of Class 2 input AN2 takes place. The scan is interrupted for the sampling and conversion of AN2.

Figure 36-5. Scan Conversion Pre-empted by Class 2 Input Trigger



36.5 ADC Module Configuration

Operation of the ADC module is directed through bit settings in the specific registers. The following instructions summarize the actions and the settings. The options and details for each configuration step are provided in the subsequent sections.

To configure the ADC module, perform the following steps:

1. Configure the analog port pins as described in 36.5.1. Configuring the Analog Port Pins.
2. Select the analog inputs to the ADC multiplexers as described in 36.5.2. Selecting the ADC Multiplexer Analog Inputs.
3. Select the format of the ADC result as described in 36.5.3. Selecting the Format of the ADC Result.
4. Select the conversion trigger source as described in 36.5.4. Selecting the Conversion Trigger Source.
5. Select the voltage reference source as described in 36.5.5. Selecting the Voltage Reference Source.
6. Select the scanned inputs as described in 36.5.6. Selecting the Scanned Inputs.
7. Select the ADC clock source and prescaler as described in 36.5.7. Selecting the Analog-to-Digital Conversion Clock Source and Prescaler.
8. Specify any additional acquisition time (if required) as described in 36.5.7. Selecting the Analog-to-Digital Conversion Clock Source and Prescaler.
9. Turn on the ADC module as described in 36.5.8. Turning ON the ADC.
10. Poll (or wait for the interrupt) for the voltage reference to be ready as described in 36.5.5. Selecting the Voltage Reference Source.
11. Enable the analog and bias circuit for the required ADC modules and, after the ADC module wakes up, enable the digital circuit as described in 36.8.3. Low-Power Mode.
12. Configure the ADC interrupts (if required) as described in 36.7. Interrupts.

36.5.1 Configuring the Analog Port Pins

The ANSELx registers for the I/O ports associated with the analog inputs are used to configure the corresponding pin as an analog or a digital pin. A pin is configured as an analog input when the corresponding ANSELx bit is '1'. When the ANSELx bit is '0', the pin is set to digital control. The ANSELx registers are set when the device comes out of Reset, causing the ADC input pins to be configured as analog inputs by default.

The TRISx registers control the digital function of the port pins. The port pins that are required as analog inputs must have their corresponding bit set in the specific TRISx register, configuring the pin as an input. If the I/O pin associated with an ADC input is configured as an output by clearing the TRISx bit, the port's digital output level (V_{OH} or V_{OL}) is converted. After a device Reset, all of the TRISx bits are set. For more information on port pin configuration, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Note: When reading a PORT register that shares pins with the ADC, any pin configured as an analog input reads as '0' when the PORT latch is read. Analog levels on any pin that is defined as a digital input but not configured as an analog input, may cause the input buffer to consume the current that exceeds the device specification.

Related Links

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

36.5.2 Selecting the ADC Multiplexer Analog Inputs

The ADC module has two inputs, referred to as the positive and negative inputs. Input selection options vary as described in the following sections.

36.5.2.1 Selection of Positive Inputs

For the shared ADC module, the positive input is shared among all Class 2 and Class 3 inputs. Input connection of the analog input ANx to the shared ADC is automatic for either the Class 2 input trigger or during a scan of Class 2 and or Class 3 inputs. Selecting inputs for scanning is described in *Selecting the Scanned Inputs* from Related Links.

Related Links
36.5.6. Selecting the Scanned Inputs

36.5.2.2 Selection of Negative Inputs

Negative input selection is determined by the setting of the DIFFx bit of the ADCIMCON1 register. The DIFFx bit allows the inputs to be rail-to-rail and either single-ended or differential. The SIGNx and DIFFx bits in the ADCIMCON1 register scale the internal ADC analog inputs and reference voltages and configure the digital result to align with the expected full-scale output range.

For the shared ADC module, the analog inputs have individual settings for the DIFFx bit. Therefore, the user has the ability to select certain inputs as single-ended and others as differential while being connected to the same shared ADC module. While sampling, the signal changes on-the-fly as single-ended or differential according to its corresponding DIFFx bit setting.

Table 36-2. Negative Input Selection

ADCIMCON1		Input Configuration	Input Voltage		Output
DIFFx	SIGNx				
1	1	Differential 2's complement	Minimum input	$V_{IN}P - V_{IN}N = -V_{REF}$	-2048
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+2047
1	0	Differential unipolar	Minimum input	$V_{IN}P - V_{IN}N = -V_{REF}$	0
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+4095
0	1	Single-ended 2's complement	Minimum input	$V_{IN}P = V_{REF}$	-2048
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+2047
0	0	Single-ended unipolar	Minimum input	$V_{IN}P = V_{REF}$	0
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+4095

Legend:

- $V_{IN}P$ = Positive S&H input
- $V_{IN}N$ = Negative S&H input
- $V_{REF} = V_{REFH} - V_{REFL}$

Note: For proper operation and to prevent device damage, input voltage levels must not exceed the limits listed in the Electrical Specifications.

36.5.3 Selecting the Format of the ADC Result

The data in the ADC Result register can be read in any of the four supported data formats. The user can select from unsigned integer, signed integer, unsigned fractional or signed fractional. Integer data is right-justified and fractional data is left-justified.

- The integer or fractional data format selection is specified globally for all analog inputs using the Fractional Data Output Format bit, FRACT (ADCCON1[23]).
- The signed or unsigned data format selection can be independently specified for each individual analog input using the SIGNx bits in the ADCIMCONx registers.

The following table provides details on how a result is formatted.

Table 36-3. ADC Result Format

FRACT	SIGNx	Description	32-bit Output Data Format			
0	0	Unsigned integer	0000	0000	0000	0000
			0000	dddd	dddd	dddd

.....continued						
FRACT	SIGNx	Description	32-bit Output Data Format			
0	1	Signed integer	ssss	ssss	ssss	ssss
			ssss	sddd	dddd	dddd
1	0	Fractional	dddd	dddd	dddd	0000
			0000	0000	0000	0000
1	1	Signed fractional	sddd	dddd	dddd	dddd
			0000	0000	0000	0000

The following code is an example for ADC Class 2 configuration and fractional format.

```
int main(int argc, char** argv) {
int result[3];

/* Configure ADCCON1 */
ADCCON1bits.FRACT = 1;    // use Fractional output format ADCCON1bits.SELRES = 3;   // ADC
resolution is 12 bits ADCCON1bits.STRGSRC = 0;  // No scan trigger.

/* Configure ADCCON2 */
ADCCON2bits.SAMC = 5;         // ADC sampling time = 5 * TAD7
ADCCON2bits.ADCDIV = 1;       // ADC clock freq is half of control clock = TAD7

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5;  // Wakeup exponent = 32 * TAD7

/* Clock setting */ ADCCON3 = 0;
ADCCON3bits.ADCSEL = 0;       // Select input clock source
ADCCON3bits.CONCLKDIV = 1;    // Control clock frequency is half of input clock
ADCCON3bits.VREFSEL = 0;      // Select AVDD and AVSS as reference source

/* Select ADC input mode */
ADCIMCON1bits.SIGN7 = 0;  // unsigned data format ADCIMCON1bits.DIFF7 = 0;

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;           // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;              // No scanning is used

/* Configure ADCCMPCONx */
ADCCMPCON1 = 0; // No digital comparators are used. Setting the ADCCMPCONx
ADCCMPCON2 = 0; // register to '0' ensures that the comparator is disabled.

/* Configure ADCFLTRx */
ADCFLTR1 = 0;             // No oversampling filters are used. ADCFLTR2 = 0;

/* Set up the trigger sources */
ADCTRGSNSbits.LVL7 = 0;
ADC1TRG2bits.TRGSRC7 = 1; // Set AN7 to trigger from software

/* Turn the ADC on */
ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;       // Enable the clock to analog bias

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7); // Wait until ADC7 is ready

/* Enable the ADC module */ ADCCON3bits.DIGEN7 = 1;       // Enable ADC7

while (1) {
/* Trigger a conversion */ ADCCON3bits.GSWTRG = 1;

/* Wait the conversions to complete */
```

```
while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result[0] = ADCDATA7;

/*
* Process results here
*
* Note 1: Loop time determines the sampling time since all inputs are Class 2.
* If the loop time happens is small and the next trigger happens before the
* completion of set sample time, the conversion will happen only after the
* sample time has elapsed.
*
* Note 2: Results are in fractional format
*
*/
}
return (1);
}
```

36.5.4 Selecting the Conversion Trigger Source

Class 2 inputs to the ADC module can be triggered for conversion either individually or as part of a scan sequence. Class 3 inputs can only be triggered as part of a scan sequence. Individual or scan triggers can originate from an event system (EVSYS), from external digital circuits connected to INT0, from external analog circuits connected to an analog comparator or through software by setting a trigger bit in an SFR.

Note: When conversion triggers for multiple Class 2 analog inputs occur simultaneously, they are prioritized according to a natural order priority scheme based on the analog input used. AN6 has the highest priority, AN7 has the next highest priority and so on.

36.5.4.1 Trigger Selection Class 2 Inputs

For each one of the Class 2 inputs, the user application can independently specify a conversion trigger source. The individual trigger source for an analog input x is specified by the TRGSRC[4:0] bits located in registers ADCTRG1 through ADCTRG2. For example, these trigger sources can include:

- **Event System (EVSYS)**: The event system is tied with many peripherals. The peripheral can be an event generator and ADC can be an event user to trigger the ADC conversion. See *User m* register from Related Links.
- **External INT0 Pin Trigger**: In this mode, the ADC module starts a conversion on an active transition on the INT0 pin. The INT0 pin may be programmed for either a rising edge input or a falling edge input to trigger the conversion process.
- **Global Software Trigger**: The ADC module can be configured for manually triggering a conversion for all inputs that have selected this trigger option. The user can manually trigger a conversion by setting the Global Software Trigger bit, GSWTRG (ADCCON3[6]).

Related Links

30.8.13. USERm

36.5.4.2 Conversion Trigger Sources and Control

The following are the possible sources for each trigger signal:

- External trigger selection through the TRGSRCx[4:0] bits in the ADCTRGx registers. This capability is supported only for Class 2 analog inputs. Typically, the user specifies a particular trigger source to initiate a conversion for specific input. All of the analog inputs may select the same trigger source if desired. In such an event, the result resembles a scanned conversion, which has its order of completion enforced by the priority of the inputs associated with the same trigger source. The first trigger selection is '00000' (no trigger), which amounts to temporarily disabling that particular trigger and, consequently, temporarily disabling that analog input from being converted. The next two selections for trigger source (GSWTRG and GLSWTRG) are software-generated trigger sources. The second software-generated trigger selection is the Global Software Trigger (GSWTRG). This trigger links to the GSWTRG bit in the ADCCON3 register,

which may be used to enable the user application to initiate a single conversion. GSWTRG is a self-clearing bit; therefore, it clears itself on the next ADC clock cycle after being set by the user application. The third software-generated trigger selection is the Global Level Software Trigger (GLSWTRG), which is linked to the GLSWTRG bit in the ADCCON3 register. This trigger may be used by the user application to initiate a burst of consecutive samples as the GLSWTRG bit is not self-clearing. The fourth trigger selection is a special selection, the Scan Trigger selection, which allows the Class 2 analog inputs to be included as members of a global scan of all inputs.

- Scanned trigger selection via the STRGSRC[4:0] bits in the ADCCON1 register and select bits in the ADCCSS1 registers. This mode is typically used to initiate the conversion of a group of analog inputs. This capability works for 2 and 3 analog inputs but is typically used for Class 3 inputs because they do not have individual associated TRGSRC bits. One of the trigger selections is the GSWTRG bit in the ADCCON3 register, which may be used to enable the user software to initiate a conversion.
- User-initiated trigger via the ADINSEL[5:0] bits and the RQCNVRT bit in the ADCCON3 register. This mode enables the user application to create an individual conversion trigger request for a specified analog input. Using this mode enables the user application to trigger the conversion of an input without changing the trigger source configuration of the ADC. This is useful in handling error situations where another software module wants ADC information without disrupting the normal operation of the ADC. This is also the preferred method to generate the initial trigger to start a digital filter sequence.
- User-controlled sampling of Class 2 and Class 3 inputs via the ADINSEL[5:0] bits and the SAMP bit in the ADCCON3 register. Setting the SAMP bit causes the Class 2 and Class 3 inputs to be in the Sampling mode while ignoring the selection of the SAMC[9:0] bits. This mode is also useful in software conversion of ADC with software-selectable sample time.

36.5.4.3 User-Requested Individual Conversion Trigger (Software ADC Conversion)

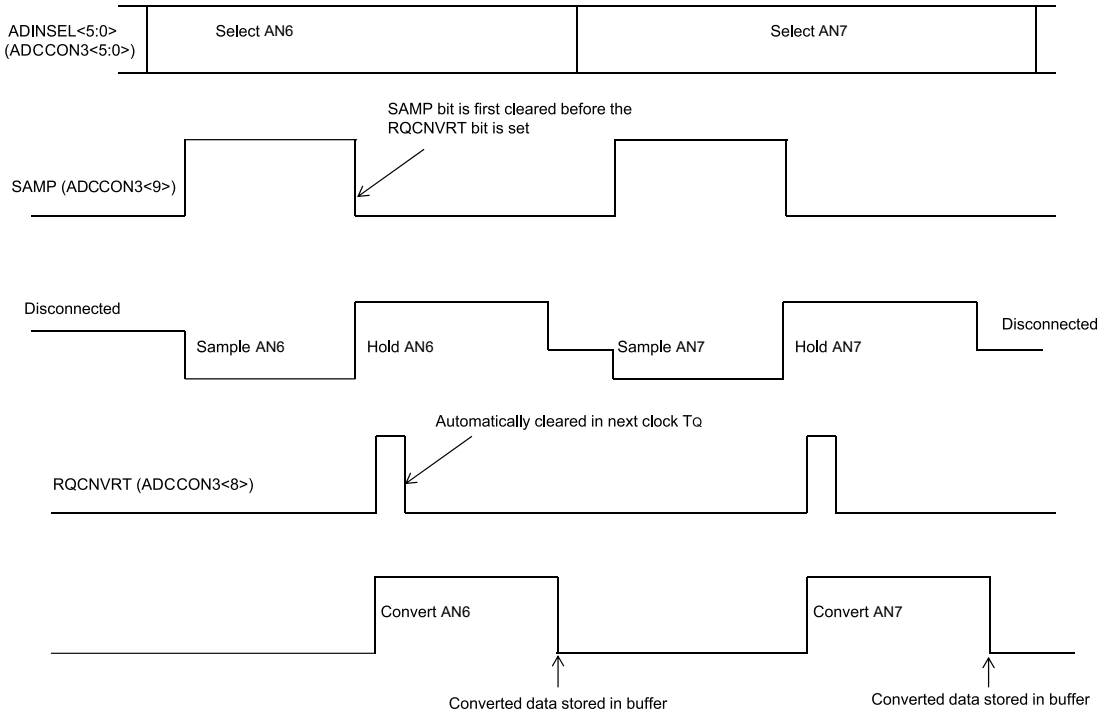
The user can explicitly request a single conversion (by software) of any selected analog input at any time during program execution without changing the trigger source configuration of the ADC.

The steps to be followed for conversion are as follows:

1. The analog input ID to be converted is specified by the ADC Input Select bits, ADINSEL[5:0] (ADCCON3[5:0]).
2. The sampling of analog input is started by setting the SAMP bit (ADCCON3[9]).
3. After the required sampling time (time delay), the SAMP bit is cleared.
4. The conversion of sampled signal is started by setting the RQCNVRT bit (ADCCON3[8]).
5. When the conversion is complete, the ARDYx bit of the ADCDSTATx register is set. The data can be read from the ADCDATAx register.

The following figure illustrates the conversion process in graphical form.

Figure 36-6. Individual Conversion Trigger Process



36.5.5 Selecting the Voltage Reference Source

The user application can select the voltage reference for the ADC module, which can be internal or external. The Voltage Reference Input Selection bits, VREFSEL[2:0] (ADCCON3[15:13]), select the voltage reference for ADC. The upper voltage reference (V_{REFH}) and the lower voltage reference (V_{REFL}) may be the internal AV_{DD} and AV_{SS} voltage rails or the band gap reference generator or the external V_{REFH}+ and V_{REF}- input pins. When the voltage reference and band gap reference are ready, the BGVRRDY (ADCCON2[31]) bit is set. If a fault occurs in the voltage reference (such as a brown-out), the REFFLT bit (ADCCON2[30]) is set. The BGVRRDY and REFFLT bits can also generate interrupts if the BGVRIEN bit (ADCCON2[15]) and REFFLTIEN bit (ADCCON2[14]), respectively, are set.

The voltages applied to the external reference pins must comply with certain specifications. See *Electrical Characteristics* from Related Links.

The Analog Input Charge Pump Enable bit, AICPMPEN (ADCCON1[12]), must be set when the difference between the selected reference voltages (V_{REFH} - V_{REFL}) is less than 0.65 * (AV_{DD} - AV_{SS}). Setting this bit does not increase the magnitude of the reference voltage; however, setting this bit reduces the series source resistance to the sampling capacitors. This maximizes the SNR for ADC using small reference voltage rails.

Related Links

43. Electrical Characteristics

36.5.6 Selecting the Scanned Inputs

All available analog inputs can be configured for scanning. Class 2 and Class 3 inputs are sampled using the shared ADC module. A single conversion trigger source is selected for all of the inputs selected for scanning using the STRGSRC[4:0] bits (ADCCON1[20:16]). On each conversion trigger, the ADC module starts converting (in the natural priority) all inputs specified in the user-specified

scan list (ADCCSS1). For Class 2 and Class 3 inputs, the trigger initiates a sequential sample/conversion process in the natural priority order.

An analog input belongs to the scan if it is:

- A Class 3 input. For Class 3 inputs, scan is the only mechanism for conversion.
- A Class 2 input that has the scan trigger selected as the trigger source by selecting the STRIG option in the TRGSRCx[4:0] bits located in the ADCTRG1 and ADCTRG2 registers

The trigger options available for scan are identical to those available for independent triggering of Class 2 inputs. Any Class 2 inputs that are part of the scan must have the STRIG option selected as their trigger source in the TRGSRCx[4:0] bits.

Note: The end-of-scan (EOS) is generated only if the last shared input conversion has completed. Until this condition is met, the scan sequence is still in effect. Therefore, the EOS Interrupt can be used for any scan sequence with any combination of input types.

The following code is an example for ADC scanning multiple inputs.

```
int main(int argc, char** argv) {
int result[3];

/* Configure ADCCON1 */
ADCCON1 = 0;              // No ADCCON1 features are enabled including: Stop-in-Idle, turbo,
// CVD mode, Fractional mode and scan trigger source. ADCCON1bits.SELRES = 3;   // ADC7
resolution is 12 bits
ADCCON1bits.STRGSRC = 1;  // Select scan trigger.

/* Configure ADCCON2 */
ADCCON2bits.SAMC = 5;     // ADC7 sampling time = 5 * TAD7
ADCCON2bits.ADCDIV = 1;   // ADC7 clock freq is half of control clock = TAD7

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5; // Wakeup exponent = 32 * TAD7

/* Clock setting */
ADCCON3bits.ADCSEL = 0;       // Select input clock source
ADCCON3bits.CONCLKDIV = 1;    // Control clock frequency is half of input clock
ADCCON3bits.VREFSEL = 0;      // Select AVDD and AVSS as reference source

ADC0TIMEbits.ADCDIV = 1;      // ADC0 clock frequency is half of control clock = TAD7
ADC0TIMEbits.SAMC = 5;        // ADC0 sampling time = 5 * TAD7
ADC0TIMEbits.SELRES = 3;      // ADC0 resolution is 12 bits

/* Select ADC input mode */
ADCIMCON1bits.SIGN0 = 0;      // unsigned data format
ADCIMCON1bits.DIFF0 = 0;      // Single ended mode
ADCIMCON1bits.SIGN7 = 0;      // unsigned data format
ADCIMCON1bits.DIFF7 = 0;      // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;               // No interrupts are used. ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                 // Clear all bits
ADCCSS1bits.CSS0 = 1;         // AN0 set for scan ADCCSS1bits.CSS7 = 1;         // AN7 (Class
2) set for scan

/* Configure ADCCMPCONx */
ADCCMPCON1 = 0;               // No digital comparators are used. Setting the ADCCMPCONx
ADCCMPCON2 = 0;               // register to '0' ensures that the comparator is disabled.

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                // No oversampling filters are used. ADCFLTR2 = 0;

/* Set up the trigger sources */
ADCTRG1bits.TRGSRC0 = 3;      // Set AN0 (Class 2) to trigger from scan source
ADCTRG2bits.TRGSRC7 = 3;      // Set AN7 (Class 2) to trigger from scan source

/* Turn the ADC on */ ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
```

```
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;       // Enable, ADC7

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY0);     // Wait until ADC0 is ready while(!
ADCANCONbits.WKRDY7);     // Wait until ADC7 is ready

/* Enable the ADC module */
ADCCON3bits.DIGEN7 = 1;          // Enable ADC7

while (1) {
/* Trigger a conversion */ ADCCON3bits.GSWTRG = 1;

/* Wait the conversions to complete */
while (ADCDSTAT1bits.ARDY0 == 0);
/* fetch the result */
result[0] = ADCDATA0;

while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result[1] = ADCDATA7;

/*
* Process results here
*
*
*/
}
return (1);
}
```

36.5.7 Selecting the Analog-to-Digital Conversion Clock Source and Prescaler

The ADC module can use the internal Fast RC (FRC) oscillator output, system clock (SYS_CLK), reference clock (REFO3) or peripheral bus clock (PB1_CLK) as the conversion clock source (T_Q). See *ADCCON3* register from Related Links.

When the ADCSEL[1:0] bits (ADCCON2[31:30]) = 01, the internal FRC oscillator is used as the ADC clock source. When using the internal FRC oscillator, the ADC module can continue to function in Sleep mode and Idle mode.

Note: It is recommended that applications that require precise timing of ADC acquisitions use SYS_CLK as the clock source for the ADC.

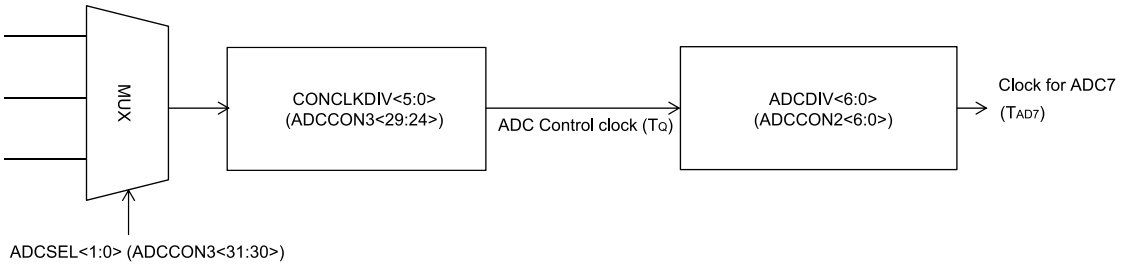
For correct ADC, the conversion clock limits must not be exceeded. Clock frequencies from 1 MHz to 32 MHz are supported by the ADC module.

The maximum rate that analog-to-digital conversions can be completed by the ADC module (effective conversion throughput) is 2 Msps. However, the maximum rate that a single input can be converted is dependent on the sampling time requirements. In addition, the sampling time depends on the output impedance of the analog signal source. For more information on sampling time, see *ADC Sampling Requirements* from Related Links.

The input clock source for the ADC is selected using the ADCSEL[1:0] bits (ADCCON3[31:30]). The input clock is further divided by the control clock divider CONCLKDIV[5:0] bits (ADCCON3[29:24]). The output clock is called the ADC control clock with a time period of T_Q.

The ADC control clock is divided before it is used for the shared ADC by the ADCDIV[6:0] bits (ADCCON2[6:0]). The time period for this clock is denoted as T_{AD7}.

Figure 36-7. Clock Derivation for Shared ADC Modules



Equation 36-1. Sample Time for the Shared ADC Module

$$t_{SAMC} = ADCCON2 < 25:16 > T_{AD7}$$

$$t_{conversion} = 2 + ADCCON2 < 22:21 > T_{AD7}$$

Equation 36-2. ADC Throughput Rate

$$\mathrm{FTP} = T_{AD7}/(T_{SAMC} + T_{CONV})$$

Where,

- T_{AD7} = The frequency of the individual ADC module

Related Links

36.13.3. ADCCON3

36.11. ADC Sampling Requirements

36.5.8 Turning ON the ADC

Turning ON the ADC module involves the following procedure.

When the ADC module enable bit, ON (ADCCON1[15]), is set to '1', the module is in the Active mode and is fully powered and functional. When the ON bit is '0', the ADC module is disabled. When disabled, the digital and analog portions of the ADC are turned OFF for maximum current savings. In addition to setting the ON bit, the analog and digital circuits of the ADC must be turned ON. See *Low-power Mode* from Related Links.

Note: Writing to the ADC control bits that control the ADC clock, input assignments, scanning, voltage reference selection, S&H circuit operating modes and interrupt configuration is not recommended while the ADC module is enabled.

Related Links

36.8.3. Low-Power Mode

36.5.9 ADC Status Bits

The ADC module includes the WKRDY7 status bit in the ADCANCON register, which indicates the current state of ADC Analog and bias circuit. The user application must not perform any ADC operations until this bit is set.

36.6 Additional ADC Functions

This section describes some additional features of the ADC module, which includes:

- Digital comparator
- Oversampling filter

36.6.1 Digital Comparator

The ADC module features digital comparators that can be used to monitor selected analog input conversion results and generate interrupts when a conversion result is within the user-specified limits. Conversion triggers are still required to initiate conversions. The comparison occurs automatically once the conversion is complete. This feature is enabled by setting the Digital Comparator Module Enable bit, ENDCMP (ADCCMPCONx[7]).

The user application makes use of an interrupt that is generated when the analog-to-digital conversion result is higher or lower than the specified high and low limit values in the ADCCMPx register. The high and low limit values are specified in the DCMPHI[15:0] bits (ADCCMPx[31:16]) and the DCMPLO[15:0] bits (ADCCMPx[15:0]).

The CMPEx bits (x = 0 through 11) in the ADCCMPENx registers are used to specify which analog inputs are monitored by the digital comparator (for 12 analog inputs, ANx, where x = 0 through 11). The ADCCMPCONx register specifies the comparison conditions that generates an interrupt, as follows:

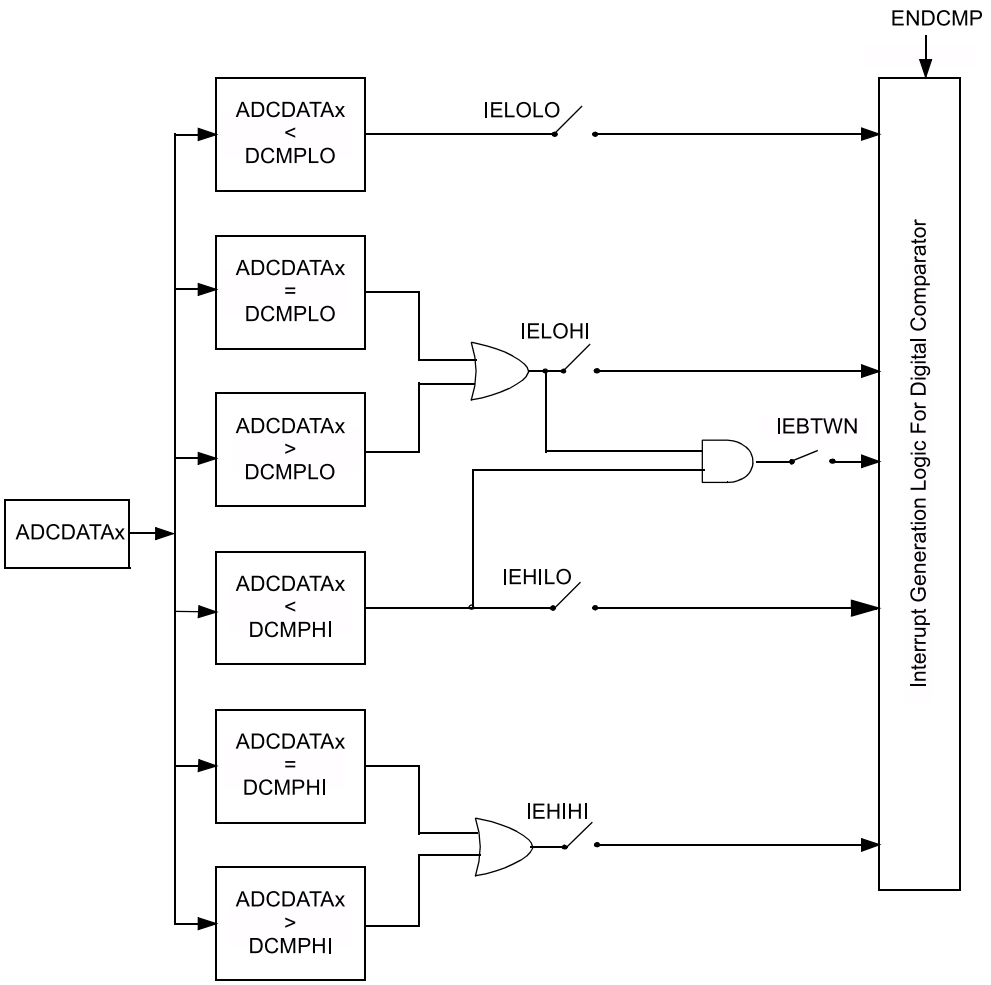
- When IEBTWN is '1', an interrupt is generated when DCMPLO ≤ ADCDATA < DCMPHI
- When IEHIHI is '1', an interrupt is generated when DCMPHI ≤ ADCDATA
- When IEHILO is '1', an interrupt is generated when ADCDATA < DCMPHI
- When IELOHI is '1', an interrupt is generated when DCMPLO ≤ ADCDATA
- When IELOLO is '1', an interrupt is generated when ADCDATA < DCMPLO

The comparator event generation is illustrated in the following figure. When the ADC module generates a conversion result, the conversion result is provided to the comparator. The comparator uses the DIFFx and SIGNx bits of the ADCIMCONx register (depending on the analog input used) to determine the data format used and to appropriately select whether the comparison must be signed or unsigned. The global ADC setting, which is specified by the FRACT bit (ADCCON1[23]), is also used to set the fractional or integer format. The digital comparator compares the ADC result with the high and low limit values (depending on the selected comparison criteria) in the ADCCMPx register.

Depending on the comparator results, a digital comparator interrupt event may be generated. If a comparator event occurs, the Digital Comparator Interrupt Event Detected status bit, DCMPED (ADCCMPCONx[5]), is set, and the Analog Input Identification (ID) bits, AINID[4:0] (ADCCMPCONx[12:8]), are automatically updated so that the user application knows which analog input generated the interrupt event.

Note: The user software must format the values contained in the ADCCMPx registers to match the converted data format as either signed or unsigned and fractional or integer.

Figure 36-8. Digital Comparator



The following code is an example for the ADC digital comparator.

```
int main(int argc, char** argv) {
int result = 0, eventFlag = 0;

/* Configure ADCCON1 */
ADCCON1 = 0;                  // No ADCCON1 features are enabled including: Stop-in-Idle,
// turbo, CVD mode, Fractional mode and scan trigger source. ADCCON1bits.SELRES = 3;       //
ADC resolution is 12 bits
ADCCON1bits.STRGSRC = 0;      // No scan trigger.

/* Configure ADCCON2 */
ADCCON2bits.SAMC = 5;         // ADC7 sampling time = 5 * TAD7
ADCCON2bits.ADCDIV = 1;       // ADC7 clock freq = TAD7

/* Initialize warm up time register */ ADCANCON = 0;


/* Select ADC input mode */
ADCIMCON1bits.SIGN7 = 0;      // unsigned data format
ADCIMCON1bits.DIFF7 = 0;      // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;               // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                  // No scanning is used
```

```
/* Configure ADCCMPCONx */
ADCCMP1 = 0;                  // Clear the register ADCCMP1bits.DCMPHI = 0xC00;  // High
limit is a 3072 result. ADCCMP1bits.DCMPLO = 0x500;  // Low limit is a 1280 result.
ADCCMPCON1bits.IEBTWN = 1;    // Create an event when the measured result is
// >= low limits and < high limit. ADCCMPEN1 = 0;                // Clear all enable bits
ADCCMPEN1bits.CMPE8 = 1;      // set the bit corresponding to AN8
ADCCMPCON1bits.ENDCMP = 1;    // enable comparator

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                // No oversampling filters are used. ADCFLTR2 = 0;
/* Set up the trigger sources */
ADCTRG2bits.TRGSRC7 = 3;         // Set AN7 (Class 2) to trigger from scan source

/* Turn the ADC on */ ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY);     // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);       // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;          // Enable the clock to analog bias

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7);     // Wait until ADC7 is ready

/* Enable the ADC module */
ADCCON3bits.DIGEN7 = 1;          // Enable ADC7

while (1) {
/* Trigger a conversion */ ADCCON3bits.GSWTRG = 1;

while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result = ADCDATA7;

/* Note: It is not necessary to fetch the result for the digital
* comparator to work. In this example we are triggering from
* software so we are using the ARDY8 to gate our loop. Reading the
* data clears the ARDY bit.
*/
/* See if we have a comparator event*/
if (ADCCMPCON1bits.DCMPED == 1) {
eventFlag = 1;
/*
* Process results here
*/
}
}
return (1);
}
```

36.6.2 Oversampling Digital Filter

The ADC module supports two oversampling digital filters. The oversampling digital filter consists of an accumulator and a decimator (down-sampler), which function together as a low-pass filter. By sampling an analog input at a higher-than-required sample rate, then, processing the data through the oversampling digital filter, the effective resolution of the ADC module can be increased at the expense of decreased conversion throughput.

To obtain x bits of extra resolution, the number of samples required (over and above the Nyquist rate) = $(2^x)^2$:

- 4x oversampling yields one extra bit of resolution (total 13 bits resolution)
- 16x oversampling yields two extra bits of resolution (total 14 bits resolution)
- 64x oversampling provides three extra bits of resolution (total 15 bits resolution)
- 256x oversampling provides four extra bits of resolution (total 16 bits resolution)

The digital filter also has an averaging mode, where it accumulates the samples and divides it by the number of samples.

Note:

1. Only Class 2 analog inputs can engage the digital filter. Therefore, the CHNLID[2:0] bits are 3 bits wide (0 to 7).
2. During the burst conversion process (repeated trigger until all required data for oversampling is obtained), if filtering Class 2 input using the shared ADC module, higher priority ADC inputs may still process conversions; lower priority ADC conversion requests are held waiting until the filter burst sequence is completed.
3. If higher priority requests occur during the digital filter sequence, they delay the completion of the filtering process. This delay may affect the accuracy of the result because the multiple samples cannot be contiguous. The user must arrange the initiation trigger for the oversampling filters to occur while there are no expected interruptions from higher priority ADC conversion requests.

The user application must configure the following bits to perform an oversampling conversion:

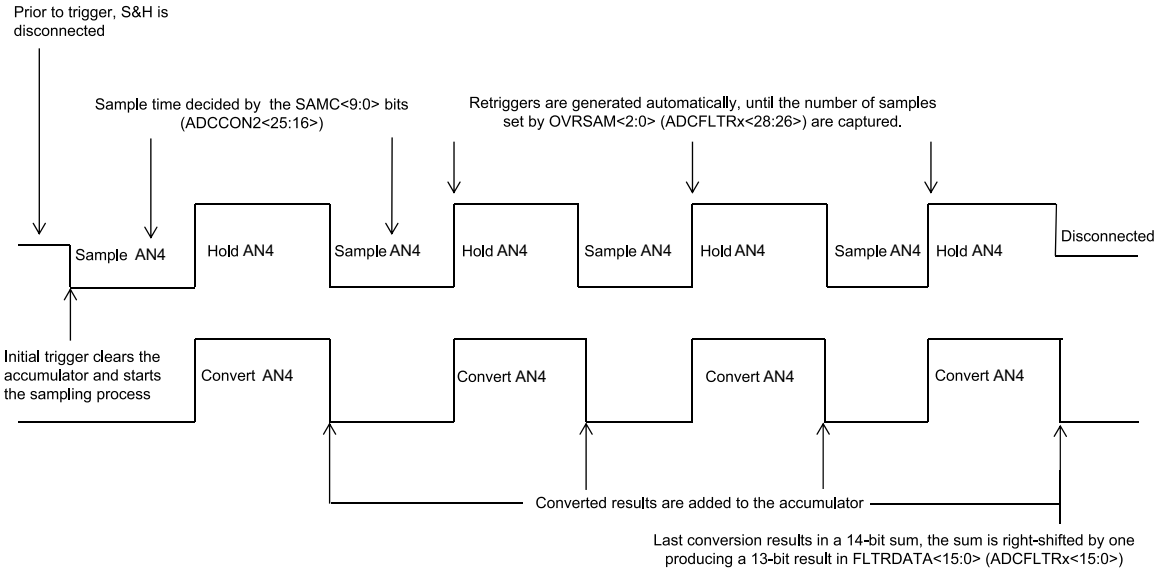
- Select the amount of oversampling through the Oversampling Filter Oversampling Ratio (OVRSAM[2:0]) bits in the ADC Filter register (ADCFLTRx[28:26]).
- Set the filter mode to either Oversampling mode or Averaging mode using the DFMODE bit(ADCFLTRx[29]).
- If the filter is set to Averaging mode and the data format is set to fractional (FRACT bit), set or clear the DATA16EN bit (ADCFLTRx[30]) to set the output resolution.
- Set the sample time for subsequent samples:
  - If using Class 2 inputs, select the sample time using the SAMC[9:0] bits (ADCCON2[25:16]).
- Select the specific analog input to be oversampled by configuring the Analog Input ID Selection bits, CHNLID[4:0] (ADCFLTRx[20:16]).
- If needed, include the oversampling filter interrupt event in the global ADC interrupt by setting the Accumulator Filter Global Interrupt Enable bit, AFGIEN (ADCFLTRx[25]).
- Enable the oversampling filter by setting the Oversampling Filter Accumulator Enable bit, AFEN (ADCFLTRx[31]).

When the digital filter module is configured, the filter's control logic waits for an external trigger to initiate the process. The trigger signal for the analog input to be oversampled causes the accumulator to be cleared and initiates the first conversion. The trigger also forces the trigger sensitivity into level mode and forces the trigger itself to 1 as long as the filter needs to acquire the user-specified number of samples via the OVRSAM[2:0] bits (ADCFLTRx[28:26]). The time delay between each acquired sample is decided by the set sample time in the SAMC[9:0] bits in the ADCCON2 register for Class 2 and the time for conversion. When the required number set by OVRSAM[2:0] are received and processed, the data stored in the FLTRDATA[15:0] bit (ADCFLTRx[15:0]) and the AFRDY bit (ADCFLTRx[24]) are set and the interrupt is generated (if enabled).

The following figure illustrates 4x oversampling using a Class 2 input. Triggering a Class 2 input initiates sampling for the length of time defined by the SAMC[9:0] bits. Retriggers generated by the oversampling logic use the SAMC[9:0] bits to set the sample time.

Class 2 inputs use the shared S&H; therefore, oversampling blocks lower priority Class 2 and Class 3 triggers. Higher priority Class 2 triggers completely disrupt the oversampling process; therefore, they must be avoided completely. The same priority rule applies to two Class 2 inputs that use two digital filters. In such a case, the higher priority input also uses the shared ADC module in Burst mode and prevents the lower priority input from using the shared ADC. Only after all required samples are obtained by the higher priority input can the lower priority input use the shared ADC to acquire samples for its own digital filtering.

Figure 36-9. 4x Oversampling of a Class 2 Input

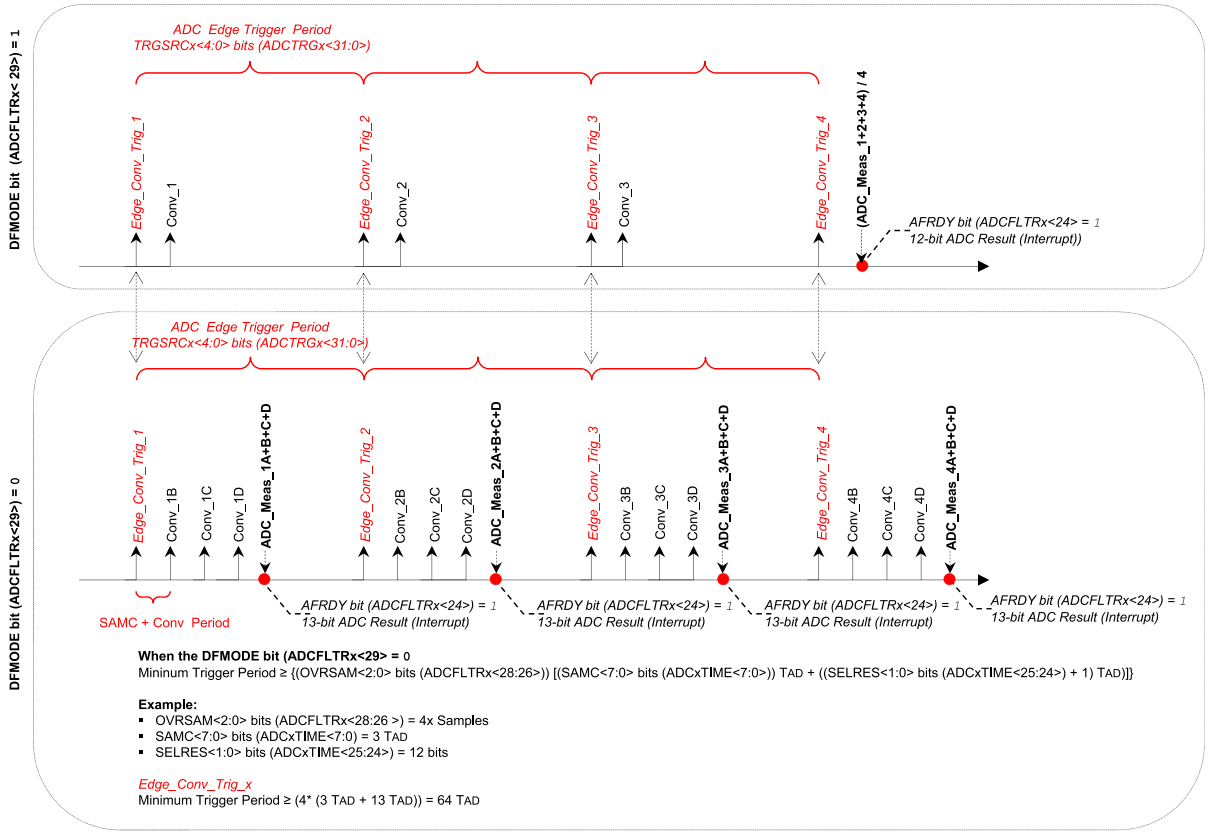


The following code is an example for ADC digital oversampling filter:

```
int main(int argc, char** argv) {
int result;

/* Configure ADCCON1 */
ADCCON1 = 0;          // No ADCCON1 features are enabled including: Stop-in-Idle, turbo,
// CVD mode, Fractional mode and scan trigger source.

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5;  // Wake-up exponent = 32 * TAD7

/* Clock setting */ ADCCON3 = 0;
ADCCON3bits.ADCSEL = 0;       // Select input clock source
ADCCON3bits.CONCLKDIV = 1;    // Control clock frequency is half of input clock
ADCCON3bits.VREFSEL = 0;      // Select AVDD and AVSS as reference source

ADCCON2bits.ADCDIV = 1;      // ADC7 clock frequency is half of control clock = TAD7
ADCCON2bits.SAMC = 5;        // ADC7 sampling time = 5 * TAD7
ADCCON1bits.SELRES = 3;      // ADC7 resolution is 12 bits


/* Select ADC input mode */
ADCIMCON1bits.SIGN0 = 0;      // unsigned data format
ADCIMCON1bits.DIFF0 = 0;      // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;               // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                  // No scanning is used
/* Configure ADCCMPCONx */
ADCCMPCON1 = 0;               // No digital comparators are used. Setting the ADCCMPCONx

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                 // Clear all bits ADCFLTR1bits.CHNLID = 0;      // Use AN0 as
the source ADCFLTR1bits.OVRSAM = 3;      // 16x oversampling ADCFLTR1bits.DFMODE = 0;      //
Oversampling mode ADCFLTR1bits.AFEN = 1;        // Enable filter 1
ADCFLTR2 = 0;                 // Clear all bits

/* Set up the trigger sources */ ADCTRGSNSbits.LVL0 = 0;        // Edge trigger
ADCTRG1bits.TRGSRC0 = 1;     // Set AN0 to trigger from software.

/* Turn the ADC on */
```

```
ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN0 = 1;       // Enable the clock to analog bias and digital control

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7); // Wait until ADC7 is ready

/* Enable the ADC module */ ADCCON3bits.DIGEN7 = 1;       // Enable ADC7

while (1) {
/* Trigger a conversion */ ADCCON3bits.GSWTRG = 1;

/* Wait for the oversampling process to complete */
while (ADCFLTR1bits.AFRDY == 0);
/* fetch the result */
result = ADCFLTR1bits.FLTRDATA;

/*
* Process result Here
*
* Note 1: Loop time determines the sampling time for the first sample.
* remaining samples sample time is determined by set sampling + conversion time.
*
* Note 2: The first 5 samples may have reduced accuracy.
*
*/
}
return (1);
}
```

Figure 36-10. ADC Filter Comparisons Example



36.7 Interrupts

The ADC module supports interrupts triggered from a variety of sources that can be processed individually or globally. An early interrupt feature is also available to compensate for interrupt servicing latency.

After an enabled interrupt is generated, the CPU jumps to the vector assigned to that interrupt. The CPU begins executing code at the vector address. The user software at this vector address must perform the required operations, such as processing the data results, clearing the interrupt flag, then exiting. See *Nested Vector Interrupt Controller (NVIC)* from Related Links for more information on interrupts and the vector address table details.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

36.7.1 Interrupt Sources

The ADC is capable of generating interrupts from the events listed in the following table.

Table 36-4. ADC Interrupt Sources

Interrupt Event	Description	Interrupt Enable Bit	Interrupt Status Bit
ANx Data Ready Event (ADC_GIRQ)	Interrupt is generated upon a completion of a conversion from an analog input source (ANx). Each of the ARDYx bits is capable of generating a unique interrupt when set using the ADCBASE register.	AGIENx of ADCGIRQEN1	ARDYx of ADCDSTAT1 register
Digital Comparator Event (ADC_DIRQ)	When a conversion's comparison criteria are met by a configured and enabled digital comparator. Each of the digital comparators is capable of generating a unique interrupt when its DCMPED bit is set.	DCMPGIEN of ADCCMPCONx register	DCMPED of ADCCMPCONx register
Oversampling Filter Data Ready Event (ADC_AIRQ)	When an oversampling filter completed the accumulation/decimation process and stored the result	AFGIEN of ADCFLTRx register	AFRDY of ADCFLTRx register
Both Band Gap Voltage and ADC Reference Voltage Ready Event (ADC_BGVR_RDY)	Interrupt is generated when both band gap voltage and ADC reference voltage are ready	BGVRIEN of ADCCON2 register	BGVRRDY of ADCCON2 register
Band Gap Fault/Reference Voltage Fault/AV_{DD} Brown-out Fault Event (ADC_FLT)	Interrupt is generated when Band Gap Fault/Reference Voltage Fault/AV_{DD} Brown-out occurs	REFFLTIEN of ADCCON2 register	REFFLT of ADCCON2 register
End of Scan Event (ADC_FCC)	Interrupt is generated when all the selected inputs completed scan	EOSIEN of ADCCON2 register	EOSRDY of ADCCON2 register
ADC Module Wake-up Event	Interrupt is generated when ADC wakes up after being enabled	WKIEN7 of ADCANCON register	WKRDY7 of ADCANCON register
Update Ready Event	Interrupt is generated when ADC SFRs are ready to be (and can be safely) updated with new values	UPDIEN of ADCCON3 register	UPDRDY of ADCCON3 register

36.7.2 ADC Base Register (ADCBASE) Usage

After conversion of ADC is complete, if the interrupt is vectored to a function that is common to all analog inputs, it takes some significant time to find the ADC input by evaluating the ARDYx bits in the ADCDSTATx. To avoid this time spent, the ADCBASE register is provided, which contains the base address of the user's ADC ISR jump table. When read, the ADCBASE register provides a sum of the contents of the ADCBASE register plus an encoding of the ARDYx bits set in the ADCDSTATx registers. This use of the ADCBASE register supports the creation of an interrupt vector address that can be used to improve the performance of an ISR.

The ARDYx bits are binary priority encoded with ARDY0 being the highest priority and ARDY11 being the lowest priority. The encoded priority result is, then, shifted left the amount specified by the number of bit positions specified by the IRQVS[2:0] bits in the ADCCON1 register, then, added to the contents of the ADCBASE register. If there are no ARDYx bits set, reading the ADCBASE register equals the value written into the ADCBASE register.

The ADCBASE register is typically loaded with the base address of a jump table that contains the address of the appropriate ISR. The k^{th} interrupt request is enabled via the AGIENx bit (0-11) in ADCGIRQEN1.

The following codes are examples for the ADCBASE register usage.

Case 1:

```
ADCBASE = 0x1234; // Set the address
ADCCON1bits.IRQVS = 2; // left shift by 2
ADCGIRQEN1bits.AGIEN0 = 1; // enable interrupt when AN0 completion is done.
```

When the ADC conversion for AN0 is complete, bit `0` of ADCDSTAT1 = ARDY0 is set.

Read value of ADCBASE = 0x1234 + (0 << 2) = 0x1234.

Therefore, the ISR must be placed at address `0x1234` for AN0.

Case 2:

```
ADCBASE = 0x1234; // Set the address
ADCCON1bits.IRQVS = 2; // left shift by 2
ADCGIRQEN1bits.AGIEN0 = 2; // enable interrupt when AN2 completion is done.
```

When the ADC conversion for AN2 is complete, bit `2` of ADCDSTAT1 = ARDY2 is set.

Read value of ADCBASE = 0x1234 + (2 << 2) = 0x123C.

Therefore, the ISR must be placed at address `0x123C` for AN2.

Note: The contents of the ADCBASE register are not altered. Summation is performed when the ADCBASE register is read and the summation result is the returned read value from the ADCBASE SFR.

36.7.3 Interrupt Enabling, Priority and Vectoring

Each of the ADC events previously mentioned generates an interrupt when its associated peripheral Interrupt bits are enabled.. Each of the ADC events previously listed also has an associated interrupt vector. See *Nested Vector Interrupt Controller (NVIC)* from Related Links for more information on the vector location and control/status bits associated with each individual interrupt.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

36.7.4 Individual and Global Interrupts

The use of the individual interrupts previously listed can significantly optimize the servicing of multiple ADC events by keeping each ISR focused on efficiently handling a specific event. In addition, different ISRs can be easily segregated according to the tasks performed, thereby making user software easier to implement and maintain. There may be cases where it is desirable to have a single ISR service multiple interrupt events. To facilitate this, each ADC event can be logically “ORed” to create a single global ADC interrupt. When an ADC event is enabled for a global interrupt, it vectors to a single interrupt routine. It is the responsibility of this single global ISR to determine the source of the interrupt through polling and process it accordingly.

Use of the Global Interrupt requires configuration of its own unique ISER, IPR0, INTFLAG and configuration of its interrupt vector as described in Interrupt Enabling, Priority and Vectoring. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Interrupts for the ADC can be configured as individual or global or utilized as both where some are processed individually and others in the global ISR.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

36.8 Power-Saving Modes of Operation

The Power-Saving mode, Standby Sleep mode and Idle mode are useful for reducing the conversion noise by minimizing the digital activity of the CPU, buses and other peripherals.

36.8.1 Standby Sleep Mode

When the device enters the Standby Sleep mode, the system clock (SYS_CLK) is halted. If an ADC module selects SYS_CLK as its clock source or selects REFO3 as its clock source (REFO3 is generated from SYS-CLK) and the Standby Sleep mode occurs during a conversion, the conversion is aborted. The converter cannot resume a partially completed conversion on exiting from the Sleep mode. The ADC register contents are not affected by the device entering or leaving the Sleep mode. The ADC module can operate during the Sleep mode if the ADC clock source is derived from a source other than SYS_CLK that is active during the Sleep mode. The FRC clock source is a logical choice for operation during Sleep; however, the REFO3 clock source can also be used, provided it has an input clock that is operational during the Sleep mode.

ADC operation during the Sleep mode reduces the digital switching noise from the conversion. When the conversion is completed, the ARDYx status bit for that analog input is set and the result is loaded into the corresponding ADC Result register (ADCDATAx).

If any of the ADC interrupts are enabled, the device is woken up from the Sleep mode when the ADC interrupt occurs. The program execution resumes at the ADC ISR if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction after the WFI instruction that placed the device in the Sleep mode.

To minimize the effects of digital noise on the ADC module operation, the user must select a conversion trigger source that ensures that the ADC take places in the Sleep mode. For example, the external interrupt pin (INT0) conversion trigger option (TRGSRC[4:0] = 00100) can be used for performing sampling and conversion while the device is in the Sleep mode.

Note: For the ADC module to operate in the Sleep mode, the ADC clock source must be set to Internal FRC (ADCSEL[1:0] bits (ADCCON2[31:30]) = 01). Alternately, the REFO3 source can be used; however, the clock source used for REFO3 must operate during Sleep mode. Any changes to the ADC clock configuration require that the ADC be disabled.

36.8.2 Operation During Idle Mode

For the ADC, the stop in Idle Mode bit, SIDL (ADCCON1[13]), specifies whether the ADC module stops on Idle or continues on Idle. If SIDL = 0, the ADC module continues normal operation when the device enters Idle mode. If any of the ADC interrupts are enabled, the device wakes up from Idle mode when the ADC interrupt occurs. The program execution resumes at the ADC ISR if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction after the WAIT instruction that placed the device in Idle mode.

If SIDL = 1, the ADC module stops in Idle mode. If the device enters Idle mode during a conversion, the conversion is aborted. The converter cannot resume a partially completed conversion on exiting from Idle mode.

36.8.3 Low-Power Mode

The ADC module can be placed in a low-power state by disabling the digital circuit for individual ADC modules that are not running. This is possible by clearing the DIGEN7 bit in the ADCCON3 register. (See *ADCCON3* register from Related Links.)

An even lower power state is possible by disabling the analog and bias circuit for ADC module that is not running. This is possible by clearing the ANEN7 bit in the ADCANCON register. (See *ADCANCON* register from Related Links.) Disabling the digital circuit to achieve Low-Power mode provides a significantly faster module restart compared to disabling and re-enabling the analog and bias circuit of the ADC module. This is because disabling and re-enabling the analog and bias circuit using the ANEN7 bit requires a wake-up time (typical minimum wake-up time of 20 µs) for the ADC module before it can be used. See *Electrical Characteristics* from Related Links for more information on the stabilization time.

When the analog and bias circuit for an ADC module is enabled, the wake-up must be polled (or through an interrupt) using the wake-up ready bits, WKRDY7, which must be equal to '1'.

Related Links

36.13.3. ADCCON3
36.13.20. ADCANCON
43. Electrical Characteristics

36.9 Effects of Reset

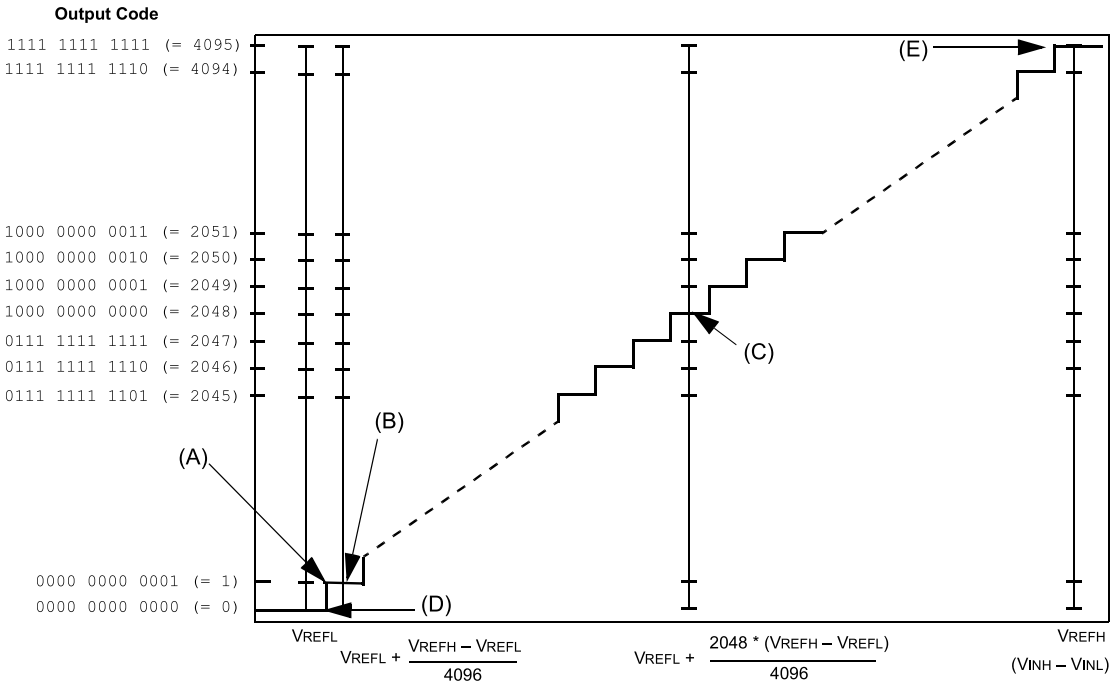
Following any Reset event, all the ADC control and status registers are reset to their default values with control bits in a non-active state. This disables the ADC module and sets the analog input pins to Analog Input mode. Any conversion that was in progress terminates, and the result cannot be written to the result buffer. The values in the ADCDATAx registers are initialized to '0x00000000' during a device Reset. The bias circuits are also turned OFF, so the ADC resuming operations wait for the bias circuits to stabilize by polling (or requesting to be interrupted by) the BGVRRDY bit (ADCCON2 register).

36.10 Transfer Function

A typical transfer function of the 12-bit ADC is illustrated in the following figure. The difference of the input voltages (V_{INH} - V_{INL}) is compared with the reference (V_{REFH} - V_{REFL}).

- The first code transition (A) occurs when the input voltage is (V_{REFH} - V_{REFL}/8192) or 0.5 LSb.
- The 0000 0000 0001 code is centered at (V_{REFH} - V_{REFL}/4096) or 1.0 LSb (B).
- The 1000 0000 0000 code is centered at (2048 * (V_{REFH} - V_{REFL})/4096) (C).
- An input voltage less than (1 * (V_{REFH} - V_{REFL})/8192) converts as 0000 0000 0000 (D).
- An input greater than (8192 * (V_{REFH} - V_{REFL})/8192) converts as 1111 1111 1111 (E).

Figure 36-11. Analog-to-Digital Transfer Function



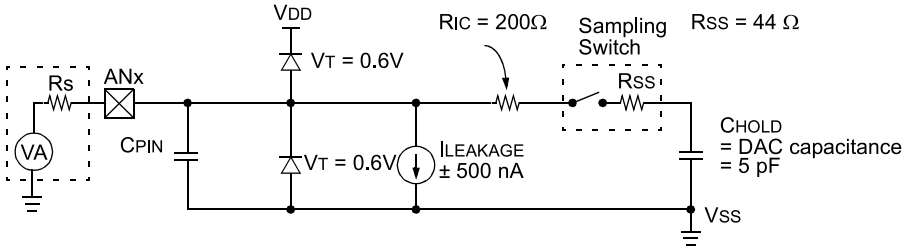
36.11 ADC Sampling Requirements

The analog input model of the 12-bit ADC is illustrated in the following figure. The total acquisition time for the analog-to-digital conversion is a function of the internal circuit settling time and the holding capacitor charge time.

For the ADC module to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the voltage level on the analog input pin. The analog output source impedance (R_S), the interconnect impedance (R_{IC}) and the internal sampling switch (R_{SS}) impedance combine to directly affect the time required to charge the C_{HOLD}. The combined impedance of the analog sources must, therefore, be small enough to fully charge (to within one-fourth LSB of the desired voltage) the holding capacitor within the selected sample time. The internal holding capacitor is in the discharged state prior to each sample operation.

At least 1 T_{AD7} time period must be allowed between conversions for the acquisition time. See *Electrical Characteristics* from Related Links.

Figure 36-12. 12-bit ADC Analog Input Model



Note: The C_{PIN} value depends on the device package and is not tested. The effect of the C_{PIN} is negligible if Rs 5 k.

Legend:

- C_{PIN} = Input capacitance
- R_{SS} = Sampling switch resistance
- R_S = Source resistance
- $I_{LEAKAGE}$ = Leakage current at the pin due to various junctions
- V_T = Threshold voltage
- R_{IC} = Interconnect resistance
- C_{HOLD} = Sample/hold capacitance

Related Links

43. Electrical Characteristics

36.11.1 Connection Considerations

Because the analog inputs employ Electrostatic Discharge (ESD) protection, they have diodes to V_{DD} and V_{SS}; therefore, the analog input must be between V_{DD} and V_{SS}. If the input voltage exceeds this range by greater than 0.3V (either direction), one of the diodes becomes forward biased, and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for antialiasing of the input signal. The R (resistive) component must be selected to ensure that the acquisition time is met. Any external components connected (through high-impedance) to an analog input pin (capacitor, Zener diode and so on) must have very little leakage current at the pin.

36.12 Register Summary

The register offsets shown below are with respect to Base Address = 0x4400_0000.

The PIC32CX-BZ3 12-bit High Speed SAR ADC module has the following Special Function Registers (SFRs):

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 ... 0x13FF	Reserved									
0x1400	ADCCON1	7:0		IRQVS[2:0]			STRGLVL			
		15:8	ON	FRZ	SIDL	AIPMPEN	CVD_EN		FSYUPB	SCANEN
		23:16	FRACT	SELRES[1:0]				STRGSRC[4:0]		
		31:24								
0x1404 ... 0x140F	Reserved									
0x1410	ADCCON2	7:0		ADCDIV[6:0]						
		15:8	BGVRIEN	REFFLTIEN	EOSIEN					
		23:16	SAMC[7:0]							
		31:24	BGVRRDY	REFFLT	EOSRDY	CVD_CPL[2:0]			SAMC[9:8]	
0x1414 ... 0x141F	Reserved									
0x1420	ADCCON3	7:0	GLSWTRG	GSWTRG	ADINSEL[5:0]					
		15:8	VREFSEL[2:0]			TRGSUSP	UPDIEN	UPDRDY	SAMP	RQCNVRT
		23:16	DIGEN7							
		31:24	ADCSEL[1:0]		CONCLKDIV[5:0]					
0x1424 ... 0x143F	Reserved									
0x1440	ADCIMCON1	7:0	DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
		15:8	DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
		23:16	DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
		31:24								
0x1444 ... 0x147F	Reserved									
0x1480	ADCGIRQEN1	7:0	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0
		15:8					AGIEN11	AGIEN10	AGIEN9	AGIEN8
		23:16								
		31:24								
0x1484 ... 0x149F	Reserved									
0x14A0	ADCCSS1	7:0	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
		15:8					CSS11	CSS10	CSS9	CSS8
		23:16								
		31:24								
0x14A4 ... 0x14BF	Reserved									
0x14C0	ADCDSTAT1	7:0	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0
		15:8					ARDY11	ARDY10	ARDY9	ARDY8
		23:16								
		31:24								
0x14C4 ... 0x14DF	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x14E0	ADCCMPEN1	7:0	CMPEx[7:0]							
		15:8					CMPEx[11:8]			
		23:16								
		31:24								
0x14E4 ... 0x14EF	Reserved									
0x14F0	ADCCMP1	7:0	DCMPLO[7:0]							
		15:8	DCMPLO[15:8]							
		23:16	DCMPHI[7:0]							
		31:24	DCMPHI[15:8]							
0x14F4 ... 0x14FF	Reserved									
0x1500	ADCCMPEN2	7:0	CMPEx[7:0]							
		15:8					CMPEx[11:8]			
		23:16								
		31:24								
0x1504 ... 0x150F	Reserved									
0x1510	ADCCMP2	7:0	DCMPLO[7:0]							
		15:8	DCMPLO[15:8]							
		23:16	DCMPHI[7:0]							
		31:24	DCMPHI[15:8]							
0x1514 ... 0x159F	Reserved									
0x15A0	ADCFLTR1	7:0	FLTRDATA[7:0]							
		15:8	FLTRDATA[15:8]							
		23:16				CHNLID[4:0]				
		31:24	AFEN	DATA16EN	DFMODE	OVRSAM[2:0]			AFGIEN	AFRDY
0x15A4 ... 0x15AF	Reserved									
0x15B0	ADCFLTR2	7:0	FLTRDATA[7:0]							
		15:8	FLTRDATA[15:8]							
		23:16				CHNLID[4:0]				
		31:24	AFEN	DATA16EN	DFMODE	OVRSAM[2:0]			AFGIEN	AFRDY
0x15B4 ... 0x15FF	Reserved									
0x1600	ADCTRG1	7:0				TRGSRC0[4:0]				
		15:8				TRGSRC1[4:0]				
		23:16				TRGSRC2[4:0]				
		31:24				TRGSRC3[4:0]				
0x1604 ... 0x160F	Reserved									
0x1610	ADCTRG2	7:0				TRGSRC4[4:0]				
		15:8				TRGSRC5[4:0]				
		23:16				TRGSRC6[4:0]				
		31:24				TRGSRC7[4:0]				
0x1614 ... 0x167F	Reserved									
0x1680	ADCCMPCON1	7:0	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
		15:8				AINID[4:0]				
		23:16	CVD_DATA[7:0]							
		31:24	CVD_DATA[15:8]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1684 ... 0x168F	Reserved									
0x1690	ADCCMPCON2	7:0	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
		15:8				AINID[4:0]				
		23:16								
		31:24								
0x1694 ... 0x16FF	Reserved									
0x1700	ADCBASE	7:0	ADCBASE[7:0]							
		15:8	ADCBASE[15:8]							
		23:16								
		31:24								
0x1704 ... 0x173F	Reserved									
0x1740	ADCTRGSNS	7:0	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
		15:8								
		23:16								
		31:24								
0x1744 ... 0x17FF	Reserved									
0x1800	ADCANCON	7:0	ANEN7							
		15:8	WKRDY7							
		23:16	WKIEN7							
		31:24					WKUPCLKCNT[3:0]			
0x1804 ... 0x1AFF	Reserved									
0x1B00	ADCSYSCFG0	7:0	AN[7:0]							
		15:8								
		23:16								
		31:24								
0x1B04 ... 0x1DFF	Reserved									
0x1E00	ADCDATA0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
...										
0x1E2C	ADCDATA11	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Related Links

6.4.1.9. CLR, SET and INV Registers

36.13 Register Description

The following is 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
- HS = Hardware Set
- HC = Hardware Cleared

36.13.1 ADCCON1 – ADC Control Register 1

Name: ADCCON1
Offset: 0x1400
Reset: 0x00601000
Property: -

This register controls the basic operation of the ADC module, including behavior in the Sleep and Idle modes and data formatting. This register also specifies the vector shift amounts for the Interrupt Controller.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FRACT	SELRES[1:0]		STRGSRC[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ON	FRZ	SIDL	AIPMPEN	CVD_EN		FSYUPB	SCANEN
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	6	5	4	3	2	1	0
		IRQVS[2:0]			STRGLVL			
Access		R/W	R/W	R/W	R/W			
Reset		0	0	0	0			

Bit 23 – FRACT Fractional Data Output Format bit

Value	Description
0	Integer
1	Fractional

Bits 22:21 – SELRES[1:0] Shared ADC Resolution bits

Note: Changing the resolution of the ADC does not shift the result in the corresponding ADCDATAx register. The result occupies 12 bits with the corresponding lower unused bits set to '0'. For example, a resolution of 6 bits results in ADCDATAx[5:0] being set to '0' and ADCDATAx[11:6] holding the result.

Value	Description
11	12 bits (default)
10	10 bits
01	8 bits
00	6 bits

Bits 20:16 – STRGSRC[4:0] ScanTrigger Source Select bits

Value	Description
10001 - 11111	Reserved
10000	EVSYS_51
01111	EVSYS_50
01110	EVSYS_49
01101	EVSYS_48
01100	EVSYS_47
01011	EVSYS_46

Value	Description
01010	EVSYS_45
01001	EVSYS_44
01000	EVSYS_43
00111	EVSYS_42
00110	EVSYS_41
00101	EVSYS_40
00100	INT0 External interrupt
00011	Reserved
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

Bit 15 – ON ADC Module Enable bit

Note: The ON bit must be set only after the ADC module is configured.

Value	Description
0	ADC module is disabled
1	ADC module is enabled

Bit 14 – FRZ Freeze in Debug Mode

Value	Description
0	Do not freeze in the Debug mode
1	Freeze in the Debug mode

Bit 13 – SIDL Stop in Idle Mode bit

Value	Description
0	Continue module operation in the Idle mode
1	Discontinue module operation when device enters the Idle mode

Bit 12 – AIPMPEN Analog Input charge Pump Enable

Note: The power-up/reset default value is 1'b0.

Bit 11 – CVD_EN CVD Enable (Capacitive Voltage Division Enable) is the bit that enables and starts the CVD operation.

Notes:

1. The software must ensure that prior to enable CVD_EN, the shared ADC core is enabled and ready for conversions; in other words, ADCANCON.ANENx = 1'b1 and ADCANCON.WKRDYx = 1'b1 and ADCCON3.CHN_EN_SHR = 1'b1.
2. The software must disable all external triggers for all second class channels by setting the corresponding ADCTRGx.TRGSRCx[4:0] = 5'h00 and, also, the third class scan trigger by setting ADCCONx.STRGSRC[4:0] = 5'h00.
3. The software must enable the ADCCSS1.CSSx channel scan select bits of all the channels to be included in the CVD scan.
4. The software must set up the Digital Comparator 1 with the necessary comparison values in ADCCMP1 and the required setup in ADCCMPCON1 (enabling the comparator itself as well as its interrupt if desired). The register ADCCMPEN1 is irrelevant for the CVD operation. The Digital Comparator 1 updates its status field ADCCMPCON1.AINID[5:0] with the channel ID just finished for CVD only upon issuing an ADCCMPCON1.DCMPED interrupt signifying a detected touch event. When the CVD accomplishes its purpose (usually after an interrupt request from the Digital Comparator 1 signaling a touch event), the software must clear first the ADCCON3.DIGEN7 bit before clearing CVD_EN. After that, the software may set again ADCCON3.DIGEN7 and start normal A/D conversions on the shared ADC core for all second and third class channels.

Bit 9 – FSYUPB Fast Synchronous UPB Clock bit

Note: ADCCON1.FSYUPB must be '0' when ADCCON1.ADCSEL[1:0] != 0.

Value	Description
0	Fast synchronous UPB clock is disabled
1	Fast synchronous UPB clock is enabled

Bit 8 – SCANEN SCAN Enable bit

Value	Description
0	When this bit is cleared, the scan cycle is re-triggerable, which means that if an edge-sensitive scan trigger arrives in the middle of a current scan cycle, the current conversion completes but the rest of the current scan is aborted, the EOSRDY status bit is asserted and the scan cycle starts again from the beginning with the first channel included in the scan cycle.
1	By setting this bit, the user disallows the scan trigger to re-start the current scan cycle and the current scan cycle completes with its last included channel before getting re-started.

Bits 6:4 – IRQVS[2:0] Interrupt Vector Shift bits

To determine the interrupt vector address, this bit specifies the amount of left-shift done to the ARDYx status bits in the ADCDSTAT1 and ADCDSTAT2 registers prior to adding with the ADCBASE register.

Interrupt Vector Address = Read Value of ADCBASE and Read Value of ADCBASE = Value written to ADCBASE + x << IRQVS[2:0], where 'x' is the smallest active input ID from the ADCDSTAT1 or ADCDSTAT2 registers (which has highest priority).

Value	Description
111	Shift x left 7 bit positions
110	Shift x left 6 bit positions
101	Shift x left 5 bit positions
100	Shift x left 4 bit positions
011	Shift x left 3 bit positions
010	Shift x left 2 bit positions
001	Shift x left 1 bit position
000	Shift x left 0 bit positions

Bit 3 – STRGLVL ScanTrigger High Level/Positive Edge Sensitivity bit

Value	Description
0	Scan trigger is positive edge sensitive. When the STRIG mode is selected (TRGSRCx[4:0] in the ADCTRGx register), only a single scan trigger is generated, which completes the scan of all selected analog inputs.
1	Scan trigger is high level sensitive. When the STRIG mode is selected (TRGSRCx[4:0] in the ADCTRGx register), the scan trigger continues for all selected analog inputs, until the STRIG option is removed.

36.13.2 ADCCON2 – ADC Control Register 2

Name: ADCCON2
Offset: 0x1410
Reset: 0x00000000
Property: -

This register controls the reference selection for the ADC module, the sample time for the shared ADC module, interrupt enable for reference, early interrupt selection and clock division selection for the shared ADC.

Bit	31	30	29	28	27	26	25	24
	BGVRRDY	REFFLT	EOSRDY	CVD_CPL[2:0]			SAMC[9:8]	
Access	R/HS/HC	R/HS/HC	R/HS/HC	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
	SAMC[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	15	14	13	12	11	10	9	8
	BGVRIEN	REFFLTIEN	EOSIEN					
Access	R/W	R/W	R/W					
Reset	0	0	0					

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

Bit 31 – BGVRRDY Band Gap Voltage/ADC Reference Voltage Status bit
Data processing is valid only after BGVRRDY is set by hardware, so the application code must check that the BGVRRDY bit is set to ensure data validity. This bit set to '0' when ON (ADCCON1[15]) = 0.

Value	Description
0	Either or both band gap voltage and ADC reference voltages (V_{REF}) are not ready
1	Both band gap voltage and ADC reference voltages (V_{REF}) are ready

Bit 30 – REFFLT Band Gap/V_{REF}/A_{VDD} BOR Fault Status bit
This bit is cleared when the ON bit (ADCCON1[15]) = 0 and the BGVRRDY bit = 1.

Value	Description
0	Band gap and V_{REF} voltage are working properly
1	Fault in band gap or the V_{REF} voltage while the ON bit (ADCCON1[15]) was set. Most likely a band gap or V_{REF} fault is caused by a BOR of the analog V_{DD} supply.

Bit 29 – EOSRDY End of Scan Interrupt Status bit
This bit is cleared when ADCCON2[31:24] are read in software.

Value	Description
0	Scanning is not complete
1	All analog inputs are considered for scanning through the scan trigger (all analog inputs specified in the ADCCSS1 register) completed scanning

Bits 28:26 – CVD_CPL[2:0] CVD Partly Line Capacitor Setting; Cpline = CVD_CPL[2:0] * 2.5 pF = 0 to 17.5 pF

Bits 25:16 – SAMC[9:0] SampleTime for the Shared ADC bits
Where T_{AD7} = Period of the ADC conversion clock for the Shared ADC controlled by the ADCCON2.ADCDIV[6:0] bits

Value	Description
1111111111	1025 T_{AD7}
...	—
0000000001	3 T_{AD7}
0000000000	2 T_{AD7}

Bit 15 – BGVRIEN Band Gap/V_{REF} Voltage Ready Interrupt Enable bit

Value	Description
0	No interrupt is generated when the BGVRRDY bit is set
1	Interrupt is generated when the BGVRDDY bit is set

Bit 14 – REFFLTIEN Band Gap/V_{REF} Voltage Fault Interrupt Enable bit

Value	Description
0	No interrupt is generated when the REFFLT bit is set
1	Interrupt is generated when the REFFLT bit is set

Bit 13 – EOSIEN End of Scan Interrupt Enable bit

Value	Description
0	No interrupt is generated when the EOSRDY bit is set
1	Interrupt is generated when the EOSRDY bit is set

Bits 6:0 – ADCDIV[6:0] Division Ratio for the Shared SAR ADC Core Clock bits
The ADCDIV[6:0] bits divide the ADC control clock (T_Q) to generate the clock for the shared SAR ADC.

Value	Description
1111111	254 * T_Q = T_{AD7}
...	—
0000011	6 * T_Q = T_{AD7}
0000010	4 * T_Q = T_{AD7}
0000001	2 * T_Q = T_{AD7}
0000000	Reserved

36.13.3 ADCCON3 – ADC Control Register 3

Name: ADCCON3
Offset: 0x1420
Reset: 0x00000000
Property: -

This register enables ADC clock selection, enables/disables the digital feature for the shared ADC module and controls the manual (software) sampling and conversion.

Note:

1. The SAMP bit has the highest priority, and setting this bit keeps the S&H circuit in Sample mode until the bit is cleared. Also, usage of the SAMP bit causes settings of SAMC[9:0] bits (ADCCON2[25:16]) to be ignored.
2. The SAMP bit only connects Class 2 and Class 3 analog inputs to the shared ADC.
3. The SAMP bit is not a self-clearing bit and it is the responsibility of application software to first clear this bit and, only after setting the RQCNVRT bit, to start the ADC.
4. Normally, when the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx[4:0] bits and STRGSRC[4:0] bits must be set to '00000' to disable all external hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software-controlled trigger RQCNVRT.

Bit	31	30	29	28	27	26	25	24
	ADCSEL[1:0]		CONCLKDIV[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	23	22	21	20	19	18	17	16
	DIGEN7							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	VREFSEL[2:0]			TRGSUSP	UPDIEN	UPDRDY	SAMP	RQCNVRT
Access	R/W	R/W	R/W	R/W	R/W	R/HS/HC	R/W	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GLSWTRG	GSWTRG	ADINSEL[5:0]					
Access	R/W	R/W, HC	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – ADCSEL[1:0] Analog-to-Digital Clock Source (T_{CLK}) bits

Value	Description
00	Peripheral Bus Clock (PB1_CLK)
01	FRC Clock
10	REFO3 Clock Output
11	System Clock (SYS_CLK)

Bits 29:24 – CONCLKDIV[5:0] Analog-to-Digital Control Clock (T_Q) Divider bits

Value	Description
111111	64 * T_{CLK}= T_Q
...	—
000011	4 * T_{CLK}= T_Q
000010	3 * T_{CLK}= T_Q

Value	Description
000001	2 * T_{CLK}= T_Q
000000	T_{CLK}= T_Q

Bit 23 – DIGEN7 Shared ADC Digital Enable bit

Value	Description
1	ADC is digital enabled
0	ADC is digital disabled

Bits 15:13 – VREFSEL[2:0] Voltage Reference (V_{REF}) Input Selection bits

Table 36-5.

VREFSEL[2:0]	AD_{REF+}	AD_{REF-}
000	AV_{DD}	AV_{SS}
001–111	Reserved	

Bit 12 – TRGSUSP Trigger Suspend bit

Value	Description
1	Triggers are blocked from starting a new analog-to-digital conversion, but the ADC module is not disabled
0	Triggers are not blocked

Bit 11 – UPDIEN Update Ready Interrupt Enable bit

Value	Description
1	Interrupt is generated when the UPDRDY bit is set by hardware
0	No interrupt is generated

Bit 10 – UPDRDY ADC Update Ready Status bit

Note: This bit is only active while the TRGSUSP bit is set and there are no more running conversions of any ADC modules.

Value	Description
1	ADC SFRs can be updated
0	ADC SFRs cannot be updated

Bit 9 – SAMP Class 2 and Class 3 Analog Input Sampling Enable bit(1,2,3,4)

Value	Description
1	The ADC S&H amplifier is sampling
0	The ADC S&H amplifier is holding

Bit 8 – RQCNVRT Individual ADC Input Conversion Request bit

This bit and its associated ADINSEL[5:0] bits enable the user to individually request an ADC of an analog input through software.

Note: This bit is automatically cleared in the next ADC clock cycle.

Value	Description
1	Trigger the conversion of the selected ADC input as specified by the ADINSEL[5:0] bits
0	Do not trigger the conversion

Bit 7 – GLSWTRG Global Level Software Trigger bit

Value	Description
1	Trigger conversion for ADC inputs that have selected the GLSWTRG bit as the trigger signal, either through the associated TRGSRC[4:0] bits in the ADCTRGx registers or through the STRGSRC[4:0] bits in the ADCCON1 register
0	Do not trigger an ADC

Bit 6 – GSWTRG Global Software Trigger bit

This bit is automatically cleared in the next ADC clock cycle.

Value	Description
0	Trigger conversion for ADC inputs that have selected the GSWTRG bit as the trigger signal, either through the associated TRGSRC[4:0] bits in the ADCTRGx registers or through the STRGSRC[4:0] bits in the ADCCON1 register
1	Do not trigger an ADC

Bits 5:0 – ADINSEL[5:0] Analog Input Select bits

These bits select the analog input to be converted when the RQCNVRT bit is set.

Value	Description
111111	Reserved
...	—
001011	VDD33/2(AN11) (Internal)
001010	VDD 1.2V(VDD_1V2(AN10)) (Internal)
001001	ADC Charge-pump 1.2V(CP_1V2(AN9)) (Internal)
001000	BandGap Reference (BG_VREF (AN8)) (Internal)
000111	AN7 is being monitored
...	—
000001	AN1 is being monitored
000000	AN0 is being monitored

36.13.4 ADCIMCON1 – ADC Input Mode Control Register 1

Name: ADCIMCON1
Offset: 0x1440
Reset: 0x00000000
Property: -

This register enables the user to select between single-ended and differential operation as well as select between signed and unsigned data format.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
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
	DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
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
	DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
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 – DIFF11 AN11 Mode bit

Value	Description
1	AN11 is using Differential mode
0	AN11 is using Single-ended mode

Bit 22 – SIGN11 AN11 Signed Data Mode bit

Value	Description
1	AN11 is using Signed Data mode
0	AN11 is using Unsigned Data mode

Bit 21 – DIFF10 AN10 Mode bit

Value	Description
1	AN10 is using Differential mode
0	AN10 is using Single-ended mode

Bit 20 – SIGN10 AN10 Signed Data Mode bit

Value	Description
1	AN10 is using Signed Data mode
0	AN10 is using Unsigned Data mode

Bit 19 – DIFF9 AN9 Mode bit

Value	Description
1	AN9 is using Differential mode
0	AN9 is using Single-ended mode

Bit 18 – SIGN9 AN9 Signed Data Mode bit

Value	Description
1	AN9 is using Signed Data mode
0	AN9 is using Unsigned Data mode

Bit 17 – DIFF8 AN8 Mode bit

Value	Description
1	AN8 is using Differential mode
0	AN8 is using Single-ended mode

Bit 16 – SIGN8 AN8 Signed Data Mode bit

Value	Description
1	AN8 is using Signed Data mode
0	AN8 is using Unsigned Data mode

Bit 15 – DIFF7 AN7 Mode bit

Value	Description
1	AN7 is using Differential mode
0	AN7 is using Single-ended mode

Bit 14 – SIGN7 AN7 Signed Data Mode bit

Value	Description
1	AN7 is using Signed Data mode
0	AN7 is using Unsigned Data mode

Bit 13 – DIFF6 AN6 Mode bit

Value	Description
1	AN6 is using Differential mode
0	AN6 is using Single-ended mode

Bit 12 – SIGN6 AN6 Signed Data Mode bit

Value	Description
1	AN6 is using Signed Data mode
0	AN6 is using Unsigned Data mode

Bit 11 – DIFF5 AN5 Mode bit

Value	Description
1	AN5 is using Differential mode
0	AN5 is using Single-ended mode

Bit 10 – SIGN5 AN5 Signed Data Mode bit

Value	Description
1	AN5 is using Signed Data mode
0	AN5 is using Unsigned Data mode

Bit 9 – DIFF4 AN4 Mode bit

Value	Description
1	AN4 is using Differential mode
0	AN4 is using Single-ended mode

Bit 8 – SIGN4 AN4 Signed Data Mode bit

Value	Description
1	AN4 is using Signed Data mode
0	AN4 is using Unsigned Data mode

Bit 7 – DIFF3 AN3 Mode bit

Value	Description
1	AN3 is using Differential mode

Value	Description
0	AN3 is using Single-ended mode

Bit 6 – SIGN3 AN3 Signed Data Mode bit

Value	Description
1	AN3 is using Signed Data mode
0	AN3 is using Unsigned Data mode

Bit 5 – DIFF2 AN2 Mode bit

Value	Description
1	AN2 is using Differential mode
0	AN2 is using Single-ended mode

Bit 4 – SIGN2 AN2 Signed Data Mode bit

Value	Description
1	AN2 is using Signed Data mode
0	AN2 is using Unsigned Data mode

Bit 3 – DIFF1 AN1 Mode bit

Value	Description
1	AN1 is using Differential mode
0	AN1 is using Single-ended mode

Bit 2 – SIGN1 AN1 Signed Data Mode bit

Value	Description
1	AN1 is using Signed Data mode
0	AN1 is using Unsigned Data mode

Bit 1 – DIFF0 AN0 Mode bit

Value	Description
1	AN0 is using Differential mode
0	AN0 is using Single-ended mode

Bit 0 – SIGN0 AN0 Signed Data Mode bit

Value	Description
1	AN0 is using Signed Data mode
0	AN0 is using Unsigned Data mode

36.13.5 ADCGIRQEN1 – ADC Global Interrupt Enable Register 1

Name: ADCGIRQEN1
Offset: 0x1480
Reset: 0x00000000
Property: -

This register specifies which of the individual input conversion interrupts can generate the global ADC interrupt.

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
					AGIEN11	AGIEN10	AGIEN9	AGIEN8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0
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 – AGIEN ADC Global Interrupt Enable bits

Value	Description
1	Interrupts are enabled for the selected analog input. The interrupt is generated after the converted data is ready (indicated by the ARDYx bit ('x' = 8-1) of the ADCDSTAT1 register)
0	Interrupts are disabled

36.13.6 ADCCSS1 – ADC Common Scan Select Register 1

Name: ADCCSS1
Offset: 0x14A0
Reset: 0x00000000
Property: -

This register specifies the analog inputs to be scanned by the common scan trigger.

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
					CSS11	CSS10	CSS9	CSS8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
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 – CSS Analog Common Scan Select bits

Notes:

1. In addition to setting the appropriate bits in this register, Class 2 analog inputs must select the STRIG input as the trigger source if they are to be scanned through the CSSx bits. Refer to the bit descriptions in the ADCTRGx registers for selecting the STRIG option.
2. If a Class 2 input is included in the scan by setting the CSSx bit to '1' and by setting the TRGSRCx[4:0] bits to STRIG mode (0b11), the user application must ensure that no other triggers are generated for that input using the RQCNVRT bit in the ADCCON3 register or the hardware input or any digital filter. Otherwise, the scan behavior is unpredictable.

Value	Description
1	Select ANx for input scan
0	Skip ANx for input scan

36.13.7 ADCDSTAT1 – ADC Data Ready Status Register 1

Name: ADCDSTAT1
Offset: 0x14C0
Reset: 0x00000000
Property: -

This register contains the interrupt status of the individual analog input conversions. Each bit represents the data-ready status for its associated conversion result.

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
					ARDY11	ARDY10	ARDY9	ARDY8
Access					R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0
Access	R/HS/HC	R/HS/HC	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	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – ARDY Conversion Data Ready for Corresponding Analog Input Ready bits

Value	Description
1	This bit is set when converted data is ready in the data register
0	This bit is cleared when the associated data register is read

36.13.8 ADCCMPEN1 – ADC Digital Comparator 1 Enable Register

Name: ADCCMPEN1
Offset: 0x14E0
Reset: 0x00000000
Property: -

These registers select which analog input conversion result is processed by the digital comparator.

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
					CMPEx[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMPEx[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 11:0 – CMPEx[11:0] ADC Digital Comparator x Enable bits
Note: CMPEx = where x stands for bit value from 0 to 11

These bits enable conversion results corresponding to the analog input to be processed by the digital comparator. CMPE0 enables AN0, CMPE1 enables AN1 and so on.
Notes:

1. CMPEx = ANx, where x = 0-11 (Digital Comparator inputs are limited to AN0 through AN11)
2. Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

36.13.9 ADCCMPEN2 – ADC Digital Comparator 2 Enable Register

Name: ADCCMPEN2
Offset: 0x1500
Reset: 0x00000000
Property: -

These registers select which analog input conversion result is processed by the digital comparator.

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
					CMPEx[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMPEx[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 11:0 – CMPEx[11:0] ADC Digital Comparator x Enable bits
Note: CMPEx = where x stands for the bit value from 0 to 11

These bits enable conversion results corresponding to the analog input to be processed by the digital comparator. CMPE0 enables AN0, CMPE1 enables AN1 and so on.
Notes:

1. CMPEx = ANx, where x = 0-11 (Digital Comparator inputs are limited to AN0 through AN11)
2. Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

36.13.10 ADCCMP1 – ADC Digital Comparator 1 Limit Value Register

Name: ADCCMP1
Offset: 0x14F0
Reset: 0x00000000
Property: -

These registers contain the high and low digital comparison values for use by the digital comparator.
Notes:

1. Changing theses bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.
2. The format of the limit values must match the format of the ADC converted value in terms of sign and fractional settings.
3. For Digital Comparator 1 used in the CVD mode, the DCMPHI[15:0] and DCMPLO[15:0] bits must always be specified in signed format as the CVD output data is differential and is always signed.

Bit	31	30	29	28	27	26	25	24
	DCMPHI[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	23	22	21	20	19	18	17	16
	DCMPHI[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	15	14	13	12	11	10	9	8
	DCMPLO[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
	DCMPLO[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 31:16 – DCMPHI[15:0] Digital Comparator x High Limit Value bits
These bits store the high limit value, which is used by the digital comparator for comparisons with ADC converted data.

Bits 15:0 – DCMPLO[15:0] Digital Comparator x Low Limit Value bits
These bits store the low limit value, which is used by digital comparator for comparisons with ADC converted data.

36.13.11 ADCCMP2 – ADC Digital Comparator 2 Limit Value Register

Name: ADCCMP2
Offset: 0x1510
Reset: 0x00000000
Property: -

These registers contain the high and low digital comparison values for use by the digital comparator.
Notes:

1. Changing theses bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.
2. The format of the limit values must match the format of the ADC converted value in terms of sign and fractional settings.

Bit	31	30	29	28	27	26	25	24
	DCMPHI[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	23	22	21	20	19	18	17	16
	DCMPHI[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	15	14	13	12	11	10	9	8
	DCMPLO[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
	DCMPLO[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 31:16 – DCMPHI[15:0] Digital Comparator x High Limit Value bits
These bits store the high limit value, which is used by digital comparator for comparisons with ADC converted data.

Bits 15:0 – DCMPLO[15:0] Digital Comparator x Low Limit Value bits
These bits store the low limit value, which is used by digital comparator for comparisons with ADC converted data.

36.13.12 ADCFLTR1 – ADC Digital Filter 1 Register

Name: ADCFLTR1
Offset: 0x15A0
Reset: 0x00000000
Property: -

These registers provide control and status bits for the oversampling filter accumulator and also include the 16-bit filter output data.

Bit	31	30	29	28	27	26	25	24
	AFEN	DATA16EN	DFMODE	OVRSAM[2:0]			AFGIEN	AFRDY
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
				CHNLID[4:0]				
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
	FLTRDATA[15:8]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit	7	6	5	4	3	2	1	0
	FLTRDATA[7:0]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit 31 – AFEN Digital Filter x Enable bit

Value	Description
1	Digital filter is enabled
0	Digital filter is disabled and the AFRDY status bit is cleared

Bit 30 – DATA16EN Filter Significant Data Length bit

Note: This bit is significant only if DFMODE = 1 (Averaging mode) and FRACT (ADCCON1[23]) = 1 (Fractional Output mode).

Value	Description
1	All 16 bits of the filter output data are significant
0	Only the first 12 bits are significant, followed by four zeros

Bit 29 – DFMODE ADC Filter Mode bit

Value	Description
1	Filter x works in Averaging mode
0	Filter x works in Oversampling Filter mode (default)

Bits 28:26 – OVRSAM[2:0] Oversampling Filter Ratio bits

Value	Description
	If DFMODE is '0'
111	128 samples (shift sum 3 bits to right, output data is in 15.1 format)
110	32 samples (shift sum 2 bits to right, output data is in 14.1 format)
101	8 samples (shift sum 1 bit to right, output data is in 13.1 format)
100	2 samples (shift sum 0 bits to right, output data is in 12.1 format)
011	256 samples (shift sum 4 bits to right, output data is 16 bits)

Value	Description
010	64 samples (shift sum 3 bits to right, output data is 15 bits)
001	16 samples (shift sum 2 bits to right, output data is 14 bits)
000	4 samples (shift sum 1 bit to right, output data is 13 bits)
	If DFMODE is '1'
111	256 samples (256 samples to be averaged)
110	128 samples (128 samples to be averaged)
101	64 samples (64 samples to be averaged)
100	32 samples (32 samples to be averaged)
011	16 samples (16 samples to be averaged)
010	8 samples (8 samples to be averaged)
001	4 samples (4 samples to be averaged)
000	2 samples (2 samples to be averaged)

Bit 25 – AFGIEN Digital Filter x Interrupt Enable bit

Value	Description
1	Digital filter interrupt is enabled and is generated by the AFRDY status bit
0	Digital filter is disabled

Bit 24 – AFRDY Digital Filter x Data Ready Status bit

Note: This bit is cleared by reading the FLTRDATA[15:0] bits or by disabling the Digital Filter module (by setting AFEN to '0').

Value	Description
1	Data is ready in the FLTRDATA[15:0] bits
0	Data is not ready

Bits 20:16 – CHNLID[4:0] Digital Filter Analog Input Selection bits

Note: Only the first 8 analog inputs, Class 2 (AN0 -AN7), can use a digital filter.

These bits specify the analog input to be used as the oversampling filter data source.

Value	Description
11111	Reserved
...	—
...	—
...	—
00111	AN7
...	—
...	—
...	—
00010	AN2
00001	AN1
00000	AN0

Bits 15:0 – FLTRDATA[15:0] Digital Filter x Data Output Value bits

The filter output data is as per the fractional format set in the FRACT bit (ADCCON1[23]). The FRACT bit must not be changed while the filter is enabled. Changing the state of the FRACT bit after the operation of the filter ended must not update the value of the FLTRDATA[15:0] bits to reflect the new format.

36.13.13 ADCFLTR2 – ADC Digital Filter 2 Register

Name: ADCFLTR2
Offset: 0x15B0
Reset: 0x00000000
Property: -

These registers provide control and status bits for the oversampling filter accumulator and also include the 16-bit filter output data.

Bit	31	30	29	28	27	26	25	24
	AFEN	DATA16EN	DFMODE	OVRSAM[2:0]			AFGIEN	AFRDY
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
				CHNLID[4:0]				
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
	FLTRDATA[15:8]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit	7	6	5	4	3	2	1	0
	FLTRDATA[7:0]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit 31 – AFEN Digital Filter x Enable bit

Value	Description
1	Digital filter is enabled
0	Digital filter is disabled and the AFRDY status bit is cleared

Bit 30 – DATA16EN Filter Significant Data Length bit

Note: This bit is significant only if DFMODE = 1 (Averaging mode) and FRACT (ADCCON1[23]) = 1 (Fractional Output mode).

Value	Description
1	All 16 bits of the filter output data are significant
0	Only the first 12 bits are significant, followed by four '0'

Bit 29 – DFMODE ADC Filter Mode bit

Value	Description
1	Filter x works in Averaging mode
0	Filter x works in Oversampling Filter mode (default)

Bits 28:26 – OVRSAM[2:0] Oversampling Filter Ratio bits

Value	Description
	If DFMODE is '0'
111	128 samples (shift sum 3 bits to right, output data is in 15.1 format)
110	32 samples (shift sum 2 bits to right, output data is in 14.1 format)
101	8 samples (shift sum 1 bit to right, output data is in 13.1 format)
100	2 samples (shift sum 0 bits to right, output data is in 12.1 format)
011	256 samples (shift sum 4 bits to right, output data is 16 bits)

Value	Description
010	64 samples (shift sum 3 bits to right, output data is 15 bits)
001	16 samples (shift sum 2 bits to right, output data is 14 bits)
000	4 samples (shift sum 1 bit to right, output data is 13 bits)
	If DFMODE is '1'
111	256 samples (256 samples to be averaged)
110	128 samples (128 samples to be averaged)
101	64 samples (64 samples to be averaged)
100	32 samples (32 samples to be averaged)
011	16 samples (16 samples to be averaged)
010	8 samples (8 samples to be averaged)
001	4 samples (4 samples to be averaged)
000	2 samples (2 samples to be averaged)

Bit 25 – AFGIEN Digital Filter x Interrupt Enable bit

Value	Description
1	Digital filter interrupt is enabled and is generated by the AFRDY status bit
0	Digital filter is disabled

Bit 24 – AFRDY Digital Filter x Data Ready Status bit

Note: This bit is cleared by reading the FLTRDATA[15:0] bits or by disabling the Digital Filter module (by setting AFEN to '0').

Value	Description
1	Data is ready in the FLTRDATA[15:0] bits
0	Data is not ready

Bits 20:16 – CHNLID[4:0] Digital Filter Analog Input Selection bits

Note: Only the first 8 analog inputs, Class 2 (AN0-AN7), can use a digital filter.

These bits specify the analog input to be used as the oversampling filter data source.

Value	Description
11111	Reserved
...	—
...	—
...	—
01100	Reserved
0111	AN7
...	—
...	—
...	—
00010	AN2
00001	AN1
00000	AN0

Bits 15:0 – FLTRDATA[15:0] Digital Filter x Data Output Value bits

The filter output data is as per the fractional format set in the FRACT bit (ADCCON1[23]). The FRACT bit must not be changed while the filter is enabled. Changing the state of the FRACT bit after the operation of the filter ended must not update the value of the FLTRDATA[15:0] bits to reflect the new format.

36.13.14 ADCTRG1 – ADC Trigger Source 1 Register

Name: ADCTRG1
Offset: 0x1600
Reset: 0x00000000
Property: -

This register controls the trigger source selection for AN0 through AN3 analog inputs.

Bit	31	30	29	28	27	26	25	24
				TRGSRC3[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TRGSRC2[4:0]				
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
				TRGSRC1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TRGSRC0[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 28:24 – TRGSRC3[4:0] Trigger Source for Conversion of Analog Input AN3 Select bits
Note: For STRIG, in addition to setting the trigger, it also requires programming of the STRGSRC[4:0] bits (ADCCON1[20:16]) to select the trigger source and requires the appropriate CSS bits to be set in the ADCCSSx registers.

Value	Description
10001 – 11111	Reserved
10000	EVSYS_51
01111	EVSYS_50
01110	EVSYS_49
01101	EVSYS_48
01100	EVSYS_47
01011	EVSYS_46
01010	EVSYS_45
01001	EVSYS_44
01000	EVSYS_43
00111	EVSYS_42
00110	EVSYS_41
00101	EVSYS_40
00100	INT0 External interrupt
00011	STRIG
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

Bits 20:16 – TRGSRC2[4:0] Trigger Source for Conversion of Analog Input AN2 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 12:8 – TRGSRC1[4:0] Trigger Source for Conversion of Analog Input AN1 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 4:0 – TRGSRC0[4:0] Trigger Source for Conversion of Analog Input AN0 Select bits
Note: See Bits 28-24 for bit value definitions.

36.13.15 ADCTRG2 – ADC Trigger Source 2 Register

Name: ADCTRG2
Offset: 0x1610
Reset: 0x00000000
Property: -

This register controls the trigger source selection for AN4 through AN7 analog inputs.

Bit	31	30	29	28	27	26	25	24
				TRGSRC7[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TRGSRC6[4:0]				
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
				TRGSRC5[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TRGSRC4[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 28:24 – TRGSRC7[4:0] Trigger Source for Conversion of Analog Input AN7 Select bits
Note: For STRIG, in addition to setting the trigger, it also requires programming of the STRGSRC[4:0] bits (ADCCON1[20:16]) to select the trigger source and requires the appropriate CSS bits to be set in the ADCCSSx registers.

Value	Description
10001 – 11111	Reserved
10000	EVSYS_51
01111	EVSYS_50
01110	EVSYS_49
01101	EVSYS_48
01100	EVSYS_47
01011	EVSYS_46
01010	EVSYS_45
01001	EVSYS_44
01000	EVSYS_43
00111	EVSYS_42
00110	EVSYS_41
00101	EVSYS_40
00100	INT0 External interrupt
00011	STRIG
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

Bits 20:16 – TRGSRC6[4:0] Trigger Source for Conversion of Analog Input AN6 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 12:8 – TRGSRC5[4:0] Trigger Source for Conversion of Analog Input AN5 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 4:0 – TRGSRC4[4:0] Trigger Source for Conversion of Analog Input AN4 Select bits
Note: See Bits 28-24 for bit value definitions.

36.13.16 ADCCMPCON1 – ADC Digital Comparator 1 Control Register

Name: ADCCMPCON1
Offset: 0x1680
Reset: 0x00000000
Property: -

This register controls the operation of Digital Comparator 1, including the generation of interrupts, comparison criteria to be used and provides status when a comparator event occurs.

Bit	31	30	29	28	27	26	25	24
	CVD_DATA[15:8]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit	23	22	21	20	19	18	17	16
	CVD_DATA[7:0]							
Access	R/HS/HC	R/HS/HC	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	0	0

Bit	15	14	13	12	11	10	9	8
				AINID[4:0]				
Access				R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
Access	R/W	R/W	R/HS/HC	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – CVD_DATA[15:0] CVD Differential Output Data
In the CVD mode, this 16-bit field gets the CVD differential output data whenever a DCMPED interrupt is generated. The value in this field is ADCCON1.FRACT-compliant and always signed because it is the result of the subtraction between the CVD positive and negative measurements.

Bits 12:8 – AINID[4:0] Digital Comparator 1 Analog Input Identification (ID) bits
When a digital comparator event occurs (DCMPED = 1), these bits identify the analog input being monitored by Digital Comparator 1.

Value	Description
11111	Reserved
...	—
...	—
...	—
01011	AN11 is being monitored
...	—
01000	AN8 is being monitored
00111	AN7 is being monitored
...	—
00001	AN1 is being monitored
00000	AN0 is being monitored

Bit 7 – ENDCMP Digital Comparator 1 Enable bit

Value	Description
1	Digital Comparator 1 is enabled
0	Digital Comparator 1 is not enabled, and the DCMPED status bit (ADCCMPCON[5]) is cleared

Bit 6 – DCMPGIEN Digital Comparator 1 Global Interrupt Enable bit

Value	Description
1	A Digital Comparator 1 interrupt is generated when the DCMPED status bit (ADCCMPCON[5]) is set
0	A Digital Comparator 1 interrupt is disabled

Bit 5 – DCMPED Digital Comparator 1 Output True Event Status bit

The logical conditions where the digital comparator becomes True are defined by the IEBTWN, IEHIHI, IEHILO, IELOHI and IELOLO bits.

Note: This bit is cleared by reading the AINID[4:0] bits or by disabling the Digital Comparator module (by setting ENDCMP to '0').

Value	Description
1	Digital Comparator 1 output true event has occurred (output of comparator is '1')
0	Digital Comparator 1 output is false (output of comparator is '0')

Bit 4 – IEBTWN Between Low/High Digital Comparator 1 Event bit

Value	Description
1	Generate a digital comparator event when DATA[31:0] is less than DCMPHI[15:0] and greater than DCMPLO[15:0]
0	Do not generate a digital comparator event

Bit 3 – IEHIHI High/High Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DCMPHI[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 2 – IEHILO High/Low Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DATA[31:0] bits are less than DCMPHI[15:0] bits
0	Do not generate an event

Bit 1 – IELOHI Low/High Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DCMPLO[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 0 – IELOLO Low/Low Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DATA[31:0] bits are less than DCMPLO[15:0] bits
0	Do not generate an event

36.13.17 ADCCMPCON2 – ADC Digital Comparator 2 Control Register

Name: ADCCMPCON2
Offset: 0x1690
Reset: 0x00000000
Property: -

These registers control the operation of Digital Comparator 2, including the generation of interrupts and the comparison criteria to be used. This register also provides the status when a comparator event occurs.

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
				AINID[4:0]				
Access				R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
Access	R/W	R/W	R/HS/HC	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:8 – AINID[4:0] Digital Comparator 2 Analog Input Identification (ID) bits
When a digital comparator event occurs (DCMPED = 1), these bits identify the analog input being monitored by the digital comparator.

Value	Description
11111	Reserved
01011	AN11 is being monitored
...	—
...	—
...	—
00111	AN7 is being monitored
...	—
00001	AN1 is being monitored
00000	AN0 is being monitored

Bit 7 – ENDCMP Digital Comparator 2 Enable bit

Value	Description
1	Digital Comparator 2 is enabled
0	Digital Comparator 2 is not enabled, and the DCMPED status bit is cleared

Bit 6 – DCMPGIEN Digital Comparator 2 Global Interrupt Enable bit

Value	Description
1	Digital Comparator 2 interrupt is generated when the DCMPED status bit is set
0	Digital Comparator 2 interrupt is disabled

Bit 5 – DCMPED Digital Comparator 2 Output True Event Status bit

The logical conditions where the digital comparator gets True are defined by the IEBTWN, IEHIHI, IEHILO, IELOHI and IELOLO bits.

Note: This bit is cleared by reading the AINID[4:0] bits or by disabling the Digital Comparator module (by setting ENDCMP to '0').

Value	Description
1	Digital Comparator 2 output true event has occurred (output of comparator is '1')
0	Digital Comparator 2 output is false (output of comparator is '0')

Bit 4 – IEBTWN Between Low/High Digital Comparator 2 Event bit

Value	Description
1	Generate a digital comparator event when DCMPLO[15:0] bits DATA[31:0] bits [DCMPHI[15:0] bits
0	Do not generate a digital comparator event

Bit 3 – IEHIHI High/High Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DCMPHI[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 2 – IEHILO High/Low Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DATA[31:0] bits are less than DCMPHI[15:0] bits
0	Do not generate an event

Bit 1 – IELOHI Low/High Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DCMPLO[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 0 – IELOLO Low/Low Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DATA[31:0] bits are less than DCMPLO[15:0] bits
0	Do not generate an event

36.13.18 ADCBASE – ADC Base Register

Name: ADCBASE
Offset: 0x1700
Reset: 0x00000000
Property: -

This register specifies the base address of the user ADC Interrupt Service Routine (ISR) jump table.

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
	ADCBASE[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
	ADCBASE[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 – ADCBASE[15:0] ADCISR Base Address bits

This register, when read, contains the base address of the user's ADC ISR jump table. The interrupt vector address is determined by the IRQVS[2:0] bits of the ADCCON1 register specifying the amount of left shift done to the ARDYx status bits in the ADCDSTAT1 register, prior to adding with ADCBASE register.

Interrupt vector address = Read value of ADCBASE

Read value of ADCBASE = Value written to ADCBASE + x << ADCCON1.IRQVS[2:0], where x is the smallest active analog input ID from the ADCDSTAT1 register (which has the highest priority).

36.13.19 ADCTRGSNS – ADC Trigger Level/Edge Sensitivity Register

Name: ADCTRGSNS
Offset: 0x1740
Reset: 0x00000000
Property: -

This register contains the setting for trigger level for each ADC analog input.

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
	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
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 – LVL Trigger Level and Edge Sensitivity bits

Notes:

1. This register specifies the trigger level for analog inputs 0 to 7.
2. The higher analog input ID belongs to Class 3, and, therefore, is only scan triggered. All Class 3 analog inputs use the scan trigger, for which the level/edge is defined by the STRGLVL bit (ADCCON1[3]).

Value	Description
1	Analog input is sensitive to the high level of its trigger (level sensitivity implies retriggering as long as the trigger signal remains high)
0	Analog input is sensitive to the positive edge of its trigger (this is the value after a reset)

36.13.20 ADCANCON – ADC Analog Warm-up Control Register

Name: ADCANCON
Offset: 0x1800
Reset: 0x00000000
Property: -

This register contains the warm-up control settings for the analog and bias circuit of the ADC module.

Bit	31	30	29	28	27	26	25	24
					WKUPCLKCNT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WKIEN7							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	WKRDY7							
Access	R/HS/HC							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	ANEN7							
Access	R/W							
Reset	0							

Bits 27:24 – WKUPCLKCNT[3:0] Wake-up Clock Count bits
These bits represent the number of ADC clocks required to warm-up the ADC module before it can perform conversion. Although the clocks are specific to each ADC, the WKUPCLKCNT bit is common to all ADC modules.

Value	Description
1111	2^{15}= 32,768 clocks
...	—
...	—
...	—
0110	2^{6}= 64 clocks
0101	2^{5}= 32 clocks
0100	2^{4}= 16 clocks
0011	2^{4}= 16 clocks
0010	2^{4}= 16 clocks
0001	2^{4}= 16 clocks
0000	2^{4}= 16 clocks

Bit 23 – WKIEN7 Shared ADC Wake-up Interrupt Enable bit

Value	Description
1	Enable interrupt and generate interrupt when the WKRDY7 status bit is set
0	Disable interrupt

Bit 15 – WKRDY7 Shared ADC Wake-up Status bit
Note: This bit is cleared by hardware when the ANEN7 bit is cleared.

Value	Description
1	ADC Analog and bias circuitry ready after the wake-up count number $2^{WKUPEXP}$ clocks after setting ANEN2 to '1'
0	ADC Analog and bias circuitry is not ready

Bit 7 – ANEN7 Shared ADC Analog and Bias Circuitry Enable bit

Value	Description
1	Analog and bias circuitry enabled. When the analog and bias circuit is enabled, the ADC module needs a warm-up time, as defined by the ADCANCON.WKUPCLKCNT[3:0] bits.
0	Analog and bias circuitry disabled

36.13.21 ADCSYSCFG0 – ADC System Configuration Register 0

Name: ADCSYSCFG0
Offset: 0x1B00
Reset: 0x00000000
Property: -

This register contains read-only bits corresponding to the analog input.

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
	AN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – AN[7:0] ADC Analog Input bits
These bits reflect the system configuration and are updated during boot-up time. By reading these read-only bits, the user application can determine whether or not an analog input in the device is available.

36.13.22 ADCDATAx – ADC Output Data Register ('x' = 0 to 11)

Name: ADCDATAx
Offset: 0x1E00 + x*0x04 [x=0..11]
Reset: 0x00000000
Property: -

These registers are the analog-to-digital conversion output data registers. The ADCDATAx register is associated with each external analog input, 0-7, plus internal analog inputs 8-11.

Bit	31	30	29	28	27	26	25	24
	DATA[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
	DATA[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
	DATA[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
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] ADC Converted Data Output bits
Note: Reading the ADCDATAx register value after changing the ADCCON1.FRACT bit converts the data into the format specified by the ADCCON1.FRACT bit.

37. Capacitive Voltage Divider (CVD) Controller for Touch Sensing

37.1 Overview

The CVD peripheral adds touch sensing capability to the PIC32CX-BZ3 device family. The enhanced CVD peripheral available on these family of devices supports both self- and mutual-capacitance measurement. The enhanced CVD controller offloads the CPU by performing touch scans with programmable phase timing and oversampling.

37.2 Features

- Self Measurement and Mutual Measurement for Touch Detection
  Note: The CVD Mutual only mode is not supported.
- Supports Up to Eight RX/TX channels for Touch Measurements
  - Supports up to eight channels in Self-measurement
  - Supports up to 16 channels in Mutual cap measurement
- AddCap Control to Optimize Sensitivity and Noise Performance
- Support for Lumping of Multiple RX Inputs and/or TX Outputs
- Oversampling of Measurements to Increase the Signal-to-Noise Ratio
- Driven Shield+ for Superior Noise Immunity and Moisture Tolerance
  - Any RX/TX line can be used for the driven shield
  - All enabled sensors are driven at the same potential as the sensor scanned
- Programmable Threshold to Detect Touch in Low-Power Measurements
- Supported in Standby Sleep and Idle Mode

37.3 Configuration

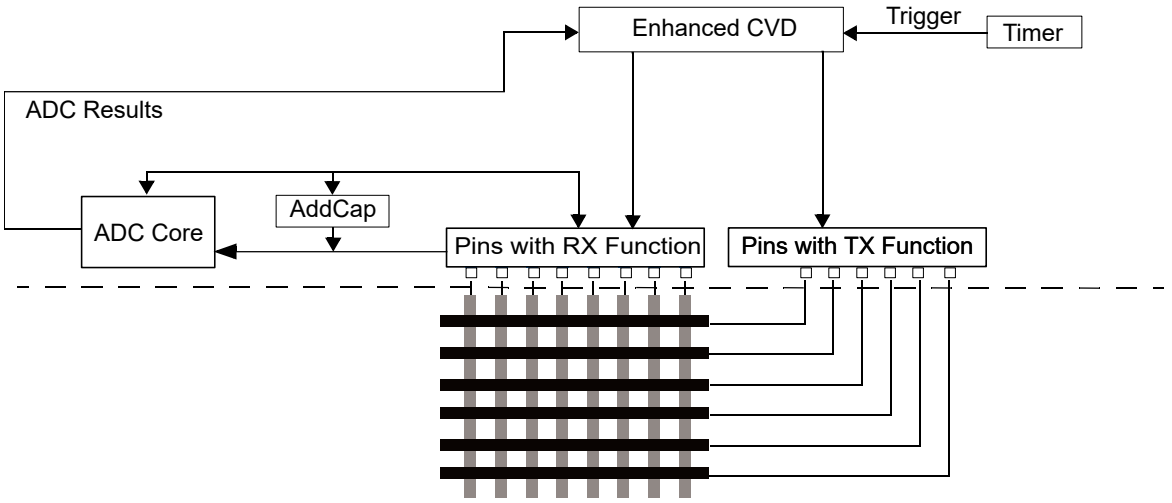
Touch library support is available to configure the touch sensors like Button/Slider/Wheel/Surface. The touch library is configured using Microchip Code Configurator (MCC). For more details, refer to *https://microchipdeveloper.com/touch:generate-touch-project-with-harmony*.

For more details on the example project, refer to https://discover.microchip.com/.

For more details on designing the touch sensor electrode, refer to *Capacitive Touch Sensor Design Guide*.

37.4 Block Diagram

Figure 37-1. Capacitive Voltage Divider (CVD) Controller Block Diagram



37.5 CVD Module Operation

The enhanced CVD uses the shared ADC SARCORE for its operations; see *Analog-to-Digital Converter (ADC)* from Related Links. The enhanced CVD controller controls the shared ADC SARCORE in a simplified mode that supports only the needs of CVD, performing the full sequence of an oversampled CVD touch scan.

The controller also controls pin functions needed for the CVD operations. Some of these pins provide RX functionality by having an analog. Some pins may have both RX and TX functions. The RX and TX pins connect to a matrix of button electrodes, a touch screen or touch pad electrode grid with horizontal and vertical structures. To determine a touch or an approach, measure the capacitance of these electrodes.

An AddCap module connects to the analog input net to augment the internal capacitance of the chip to match the base capacitance of the touch sensor. Measurements are triggered either by a CPU or a timer through EVSYS.

Related Links

36. Analog-to-Digital Converter (ADC)

37.5.1 Theory of Operation

This section describes the full sequence of an oversampled CVD touch scan for self and mutual mode.

37.5.1.1 CVD Self Sensing

A CVD operation begins with the internal Sample-and-Hold capacitor (S&H Cap) being disconnected from the path that connects it to the external capacitive sensor node. While disconnected, S&H Cap is precharged to V_{DD} or discharged to V_{SS}. The sensor node is either discharged or charged to V_{SS} or V_{DD}, respectively, to the opposite level of S&H Cap. When the precharge phase is complete, the V_{DD}/V_{SS} bias paths for the two nodes are disconnected and the paths between S&H Cap and the external sensor node is reconnected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged S&H Cap and sensor nodes, which results in a final voltage level setting on S&H Cap, which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on S&H Cap. This process is, then, repeated with the selected precharge levels inverted

for both the S&H Cap and the sensor nodes. The waveform for two CVD measurements, which is known as the differential CVD measurement, is illustrated in the following figure.

Figure 37-2. CVD Self Sensing Waveform

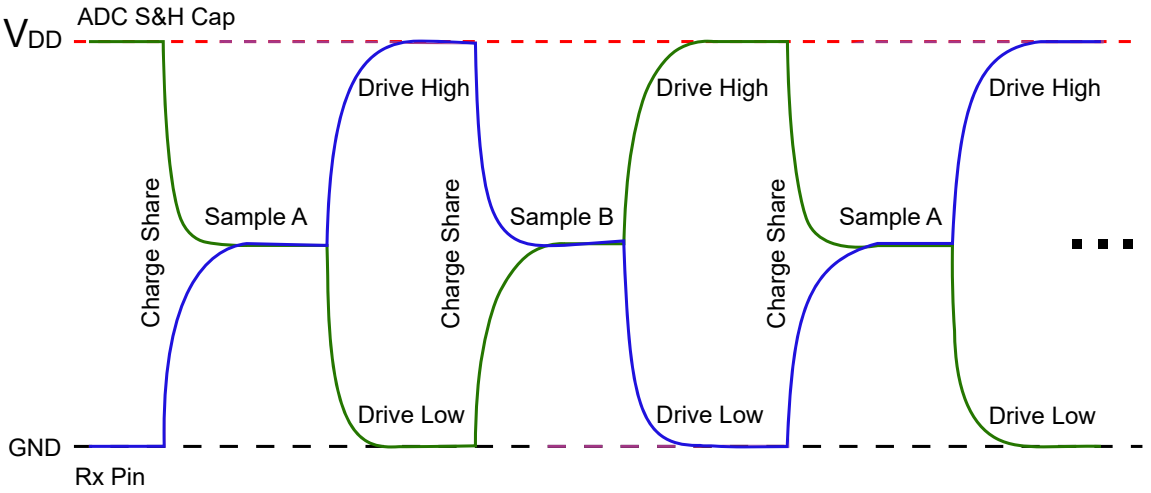
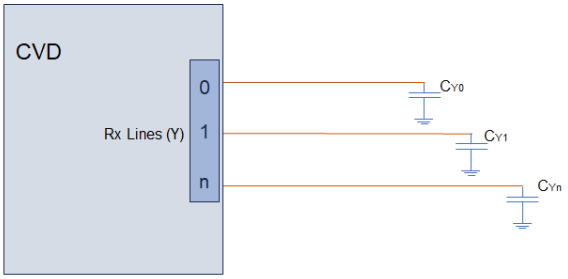


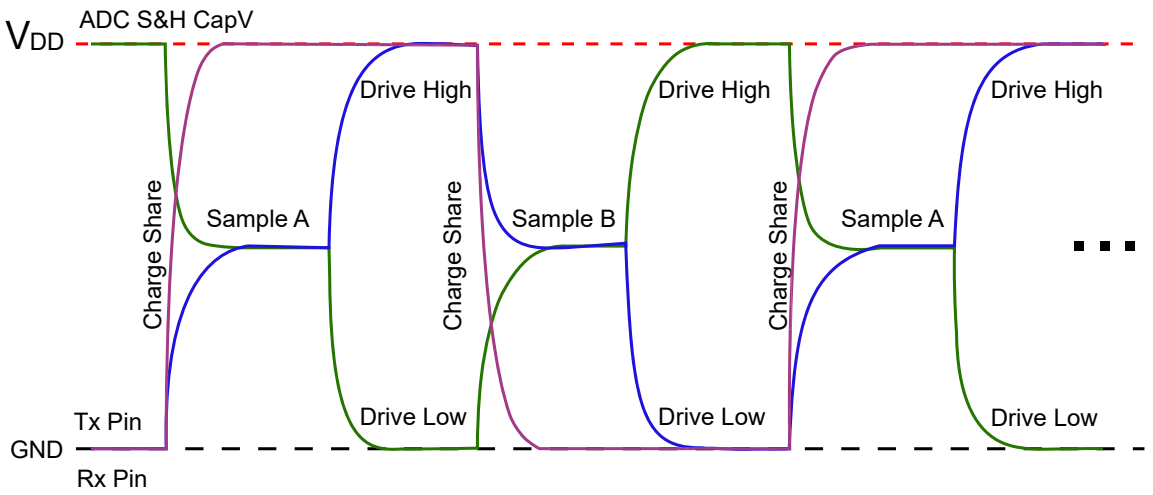
Figure 37-3. Typical Self Cap Sensor Arrangements



37.5.1.2 CVD Self Sensing with Driven Shield

The driven shield signal is a square wave that is driven high during all of Sample A and low during all of Sample B. This signal aligns in-phase with the external sensor's starting voltage for each sample. The following figure illustrates the waveform of CVD self and driven shield waveform.

Figure 37-4. CVD Self Sensing with Driven Shield



37.5.1.3 CVD Mutual Sensing

The mutual capacitance measurement is performed between two lines, namely transmit (TX) and receive (RX). The mutual TX line is driven in sync with the charge-sharing phase of self-cap measurement resulting in charge contribution from the TX pin. The resulting charge distribution between S&H Cap and the RX line is a combination of charge accumulated on S&H Cap and charge contributed by driving the TX line.

Figure 37-5. CVD Mutual Sensing Waveform

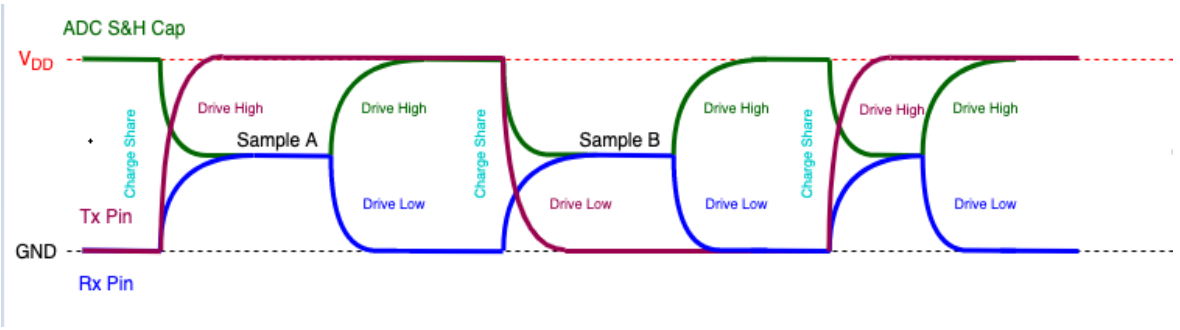
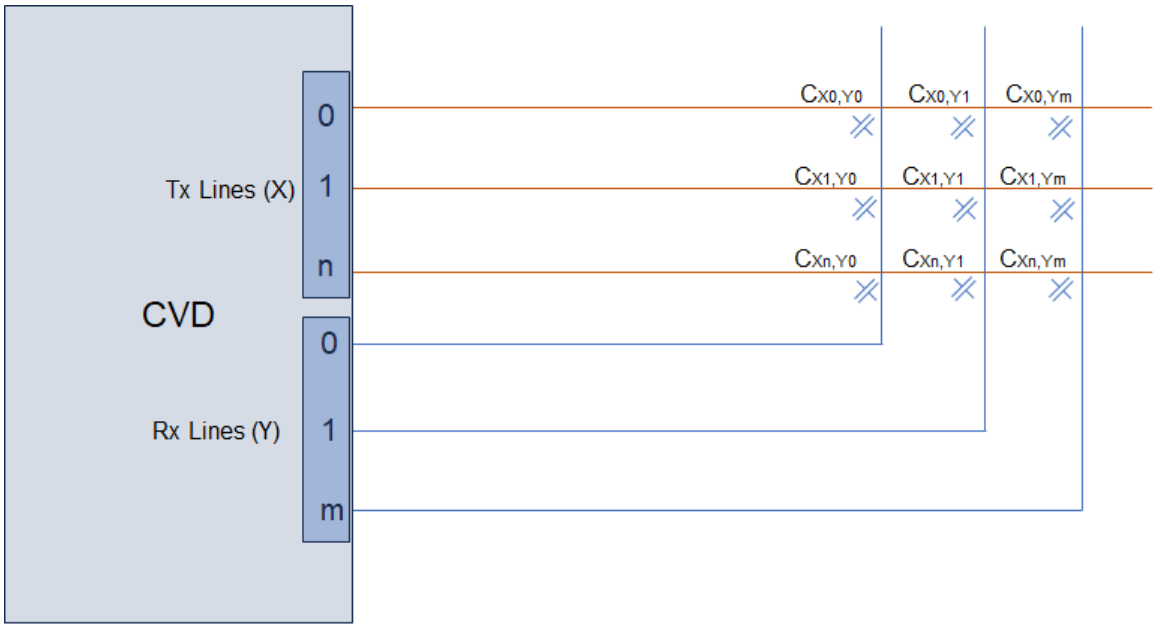


Figure 37-6. Typical Mutual Cap Sensor Arrangements



37.5.2 Channel Sets

The enhanced CVD module can control up to eight RX inputs and eight TX inputs. It enables the user to map up to eight specific CVDRx pins sequential RX indexes and up to eight specific CVDTx pins to sequential TX indexes via Special Function Registers (SFRs). The actual pins used can, therefore, be in any order and with any gaps required. This enables the PCB designer to map I/O pins to panel RX/TX functions with greater ease.

37.5.2.1 Lumping/Channel Grouping/STRIDE

Each scan can perform touch measurement on multiple RX/TX pins via the STRIDE (also known as Lump and Channel Grouping) setting, enabling the user to scan multiple RX inputs in parallel and/or drive multiple TX outputs in parallel.

37.5.2.2 Scan Descriptors

The enhanced CVD module includes four scan descriptors to off-load the intervention requirements and timing dependencies of the CPU. Each scan descriptor includes the RX and TX indexes to be scanned and the strides for each, as well as the time for each measurement phase, the amount of oversampling and a threshold at which to act on the data.

Each descriptor indicates:

- RX indexes to be scanned
- TX indexes to be driven
- Number of RX indexes to scan at one time
- Number of TX indexes to scan at one time
- Independent enable for Self (RX-drive) and combination of Self and Mutual modes
- Channel timing control
- Oversampling or threshold support
- Provision to enable an interrupt when complete or when threshold exceeds
- The Enable mode settings to enable the single, continuous or continuous until threshold scanning

37.5.3 Oversampling

A scan descriptor can program each measurement for any amount of oversampling from 0-127 additional samples. This accumulates all the samples into a 19-bit result. Do not perform division or averaging to prevent data loss and to enable non-power-of-two oversampling.

37.5.4 Phase Timing

The user can tune the timing of the main six phases of the CVD measurement as per application needs.

The scan descriptor holds time values for each of the following phases of the measurement cycle:

- Charge
- Acquire
- Conversion
- Spacing between polarity measurements
- Spacing between oversampling of same point(s)
- Spacing between channels

37.5.5 Thresholds

Each scan descriptor specifies a threshold value. If the application wishes to receive all the accumulated oversample data, it can set the threshold to zero. Otherwise, the scan descriptor can be set to only store data that exceed the requested threshold or to store all data and assert an interrupt only when the threshold is met.

37.5.6 Interrupts

Configure the enhanced CVD to cause an interrupt when the FIFO exceeds a programmed threshold and/or when a scan descriptor exceeds a threshold and/or completes its function.
Note: FIFO threshold interrupt is not supported in the Standby Sleep mode.

37.5.7 Triggers

The enhanced CVD can be set to start the enabled Scan Descriptors upon two possible trigger events. One is a software trigger SFR bit and another trigger is from any of EVSYS generators. See *Event System (EVSYS)* from Related Links.

If a scan descriptor is set to run in the continuous mode, it runs when it receives a trigger and stays enabled. The CVD module then waits for the next trigger to run the next enabled scan descriptor. Alternatively, the scan descriptor may instruct the CVD module to continue scanning the one descriptor on every trigger until a threshold is met. These modes allow the CVD module to run autonomously, being triggered by a system timer while the CPU sleeps (Idle mode or Standby Sleep mode), and interrupt the CPU to wake it if a tough threshold is reached. The first mode is useful for scanning various portions of the panel at regular intervals. The second mode is useful for scanning a low-resolution scan on every trigger, then, moving to a higher resolution scan descriptor if a threshold is met.

Related Links

30. Event System (EVSYS)

37.5.8 FIFO

Results are stored in a FIFO with a depth of 8 word. The FIFO contains the accumulated result data and the TX and RX indexes for that data. The user can configure the CVD module to store all accumulated results into the FIFO or only results that exceed the threshold specified in the scan descriptor. In either case, the user can generate an interrupt when the FIFO exceeds a watermark value.

The application has the choice of reading out:

- Only a single 32-bit word that contains the RX and TX indexes and the delta of the positive and negative accumulated CVD measurements
- A total of three 32-bit words to read the actual positive and negative CVD measurement along with delta

The FIFO uses following registers:

- 37.7.4. CVDRESH – CVD Results POS FIFO Read Register
- 37.7.5. CVDRESL – CVD Results NEG FIFO Read Register
- 37.7.6. CVDRESD – CVD Results Descriptor FIFO Read Register

37.6 Register Summary

See the *CVD* 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	CVDCON	7:0			CLKSEL[1:0]		TRIGSEL[3:0]			
		15:8	FIFOTH[7:0]							
		23:16	THSTR				CVDIEN	FIFOIEN	FIFOTH[9:8]	
		31:24	ON	FRZ	SIDL	ORDER	SDWREN		ABORT	SWTRIG
0x04	CVDADC	7:0					DIGEN	DIFFPEN	SELRES[1:0]	
		15:8								
		23:16								
		31:24								
0x08	CVDSTAT	7:0		SD2INT	SD2DONE	SD2BUSY		SD1INT	SD1DONE	SD1BUSY
		15:8		SD4INT	SD4DONE	SD4BUSY		SD3INT	SD3DONE	SD3BUSY
		23:16	FIFOCNT[7:0]							
		31:24	FIFOFULL	FIFOWM	FIFOMT				FIFOCNT[9:8]	
0x0C ... 0x0F	Reserved									
0x10	CVDRESH	7:0	POS[7:0]							
		15:8	POS[15:8]							
		23:16						POS[18:16]		
		31:24								
0x14	CVDRESL	7:0	NEG[7:0]							
		15:8	NEG[15:8]							
		23:16						NEG[18:16]		
		31:24								
0x18	CVDRESD	7:0	DELTA[7:0]							
		15:8	DELTA[15:8]							
		23:16	RXINDEX[4:0]						DELTA[17:16]	
		31:24	TXINDEX[4:0]						SDNUM[1:0]	
0x1C ... 0x7F	Reserved									
0x80	CVDRX0	7:0	RXAN0[5:0]							
		15:8	RXAN1[5:0]							
		23:16	RXAN2[5:0]							
		31:24	RXAN3[5:0]							
0x84	CVDRX1	7:0	RXAN4[5:0]							
		15:8	RXAN5[5:0]							
		23:16	RXAN6[5:0]							
		31:24	RXAN7[5:0]							
0x88 ... 0xBF	Reserved									
0xC0	CVDTX0	7:0	TXAN0[5:0]							
		15:8	TXAN1[5:0]							
		23:16	TXAN2[5:0]							
		31:24	TXAN3[5:0]							
0xC4	CVDTX1	7:0	TXAN4[5:0]							
		15:8	TXAN5[5:0]							
		23:16	TXAN6[5:0]							
		31:24	TXAN7[5:0]							
0xC8 ... 0xFF	Reserved									
0x0100	CVDSD0C1	7:0		SD0OVRSAMP[6:0]						
		15:8	SD0TH[7:0]							
		23:16	SD0TH[15:8]							
		31:24	SD0TH[23:16]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0104	CVDSD0C2	7:0	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXBEG[5:0]					
		15:8	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXEND[5:0]					
		23:16	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXBEG[5:0]					
		31:24	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXEND[5:0]					
0x0108	CVDSD0C3	7:0		SD0CHGTIME[6:0]						
		15:8		SD0ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD0EN[1:0]				SD0BUF	SD0INTEN	SD0SELF	SD0MUT
0x010C	CVDSD0T2	7:0		SD0CONTIME[6:0]						
		15:8		SD0POLTIME[6:0]						
		23:16		SD0OVRTIME[6:0]						
		31:24		SD0CHNTIME[6:0]						
0x0110	CVDSD1C1	7:0		SD1OVRSAMP[6:0]						
		15:8	SD1TH[7:0]							
		23:16	SD1TH[15:8]							
		31:24	SD1TH[23:16]							
0x0114	CVDSD1C2	7:0	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXBEG[5:0]					
		15:8	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXEND[5:0]					
		23:16	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXBEG[5:0]					
		31:24	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXEND[5:0]					
0x0118	CVDSD1C3	7:0		SD1CHGTIME[6:0]						
		15:8		SD1ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD1EN[1:0]				SD1BUF	SD1INTEN	SD1SELF	SD1MUT
0x011C	CVDSD1T2	7:0		SD1CONTIME[6:0]						
		15:8		SD1POLTIME[6:0]						
		23:16		SD1OVRTIME[6:0]						
		31:24		SD1CHNTIME[6:0]						
0x0120	CVDSD2C1	7:0		SD2OVRSAMP[6:0]						
		15:8	SD2TH[7:0]							
		23:16	SD2TH[15:8]							
		31:24	SD2TH[23:16]							
0x0124	CVDSD2C2	7:0	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXBEG[5:0]					
		15:8	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXEND[5:0]					
		23:16	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXBEG[5:0]					
		31:24	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXEND[5:0]					
0x0128	CVDSD2C3	7:0		SD2CHGTIME[6:0]						
		15:8		SD2ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD2EN[1:0]				SD2BUF	SD2INTEN	SD2SELF	SD2MUT
0x012C	CVDSD2T2	7:0		SD2CONTIME[6:0]						
		15:8		SD2POLTIME[6:0]						
		23:16		SD2OVRTIME[6:0]						
		31:24		SD2CHNTIME[6:0]						
0x0130	CVDSD3C1	7:0		SD3OVRSAMP[6:0]						
		15:8	SD3TH[7:0]							
		23:16	SD3TH[15:8]							
		31:24	SD3TH[23:16]							
0x0134	CVDSD3C2	7:0	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXBEG[5:0]					
		15:8	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXEND[5:0]					
		23:16	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXBEG[5:0]					
		31:24	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXEND[5:0]					
0x0138	CVDSD3C3	7:0		SD3CHGTIME[6:0]						
		15:8		SD3ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD3EN[1:0]				SD3BUF	SD3INTEN	SD3SELF	SD3MUT
0x013C	CVDSD3T2	7:0		SD3CONTIME[6:0]						
		15:8		SD3POLTIME[6:0]						
		23:16		SD3OVRTIME[6:0]						
		31:24		SD3CHNTIME[6:0]						

Related Links

8. Product Memory Mapping Overview

37.7 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.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description. See *Peripheral Access Controller (PAC)* 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.

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
- HS = Hardware Set
- HC = Hardware Cleared

Related Links

24. Peripheral Access Controller (PAC)

37.7.1 CVDCON - CVD Control Register

Name: CVDCON
Offset: 0x00
Reset: 0x00020000
Property: -

Bit	31	30	29	28	27	26	25	24
	ON	FRZ	SIDL	ORDER	SDWREN		ABORT	SWTRIG
Access	R/W	R/W	R/W	R/W	R/W		W/HC	W/HC
Reset	0	0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	THSTR				CVDIEN	FIFOIEN	FIFOTH[9:8]	
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	1	0

Bit	15	14	13	12	11	10	9	8
	FIFOTH[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
			CLKSEL[1:0]		TRIGSEL[3:0]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 31 – ON Enables the State Machine to scan the enabled Scan Descriptors upon next trigger
Before turning ON bit from 1'b1 to 1'b0, the Scan Enable bits of all descriptors must be cleared and the CVD controller must either be allowed to finish any scan in progress or must be instructed to abort the scan with the ABORT bit.

Bit 30 – FRZ Freeze Mode

Value	Description
1	CVD controller stops in the Debugger mode
0	CVD controller runs in the Debugger mode

Bit 29 – SIDL Stop in Idle Mode bit

Value	Description
1	CVD controller stops when device enters the Idle mode
0	CVD controller continues running in the Idle mode

Bit 28 – ORDER RX/TX Loop Order

Value	Description
1	Scans all the requested TX indexes, then increments RX index and continues operation
0	Scans all the requested RX indexes, then increments TX index and continues operation

Bit 27 – SDWREN Scan Descriptor Write Enable

Value	Description
1	Enables writes to the scan descriptors
0	Prevents writes to the scan descriptors

Bit 25 – ABORT Abort Current Scan
Note: The controller moves on to the next enabled Scan Descriptor if there is one; otherwise, it goes to idle state. Hardware clears this bit.

Value	Description
1	Aborts the current scan
0	CVD controller continues with the current scan

Bit 24 – SWTRIG Software Trigger control. Starts scan manually

Note: Hardware clears this bit.

Value	Description
1	Starts scan manually
0	Continues without the scan

Bit 23 – THSTR Threshold Store Mode

Value	Description
1	Stores only the results which exceed the programmed threshold for the Scan Descriptor
0	Stores all the results in FIFO

Bit 19 – CVDIEN Global Interrupt Enable

Value	Description
1	Enables the FIFO and scan descriptor interrupts
0	Disables the FIFO and scan descriptor interrupts

Bit 18 – FIFOIEN FIFO Threshold Interrupt Enable

Value	Description
1	Controller asserts an interrupt when the FIFO threshold is met
0	Controller does not assert an interrupt when the FIFO threshold is met

Bits 17:8 – FIFOTH[9:0] Threshold for the results FIFO

These bits contain threshold for the results FIFO that causes an interrupt and watermark FIFOWM status bit assertion.

Bits 5:4 – CLKSEL[1:0] Clock Select for CVD

Value	Description
00	PB1_CLK
01	POSC
02	LPRC
03	Reserved

Bits 3:0 – TRIGSEL[3:0] Trigger select for starting the scan

Value	Description
0000	SFR controlled software trigger
0001	EVSYS event. See *Event System (EVSYS)* from Related Links.
...	
...	
...	
1111	Reserved

Related Links

30. Event System (EVSYS)

37.7.2 CVD ADC Configuration Register

Name: CVDADC
Offset: 0x04
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
					DIGEN	DIFFPEN	SELRES[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – DIGEN
Shared ADC Digital Enable bit (Differential Mode Select from ADC controller)

Value	Description
1	Shared ADC is digital enabled
0	Shared ADC is digital disabled

Bit 2 – DIFFPEN
Controls differential mode operation of ANN0.

Value	Description
1	ANN0 (Differential) enabled
0	ANN0 (Differential) disabled

Bits 1:0 – SELRES[1:0] Shared ADC Resolution bits
Read as '0'.
Note: Changing the resolution of the ADC does not shift the result in the corresponding ADCDATAx register. The result will still occupy 12 bits, with the corresponding lower unused bits set to '0'. For example, a resolution of 6 bits will result in ADCDATAx[5:0] being set to '0' and ADCDATAx[11:6] holding the result.

Value	Description
00	6 bits
01	8 bits
10	10 bits
11	12 bits (default)

37.7.3 CVD Status Register

Name: CVDSTAT
Offset: 0x08
Reset: 0x20000000
Property: -

Bit	31	30	29	28	27	26	25	24
	FIFOFULL	FIFOWM	FIFOMT				FIFOCNT[9:8]	
Access	R	R	R				R	R
Reset	0	0	1				0	0

Bit	23	22	21	20	19	18	17	16
	FIFOCNT[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
		SD4INT	SD4DONE	SD4BUSY		SD3INT	SD3DONE	SD3BUSY
Access		R/W/HS	R	R		R/W/HS	R	R
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		SD2INT	SD2DONE	SD2BUSY		SD1INT	SD1DONE	SD1BUSY
Access		R/W/HS	R	R		R/W/HS	R	R
Reset		0	0	0		0	0	0

Bit 31 – FIFOFULL Results FIFO is Full

Value	Description
1	FIFO is full
0	Not full

Bit 30 – FIFOWM

Value	Description
1	FIFO reached the programmed FIFOTHRESH threshold
0	FIFO did not reach the programmed FIFOTHRESH threshold

Bit 29 – FIFOMT

Value	Description
1	FIFO is empty
0	FIFO is not empty

Bits 25:16 – FIFOCNT[9:0]
These bits indicate the number of words in the Results FIFO.

Bit 14 – SD4INT

Value	Description
1	Scan Descriptor 4 caused an interrupt
0	Scan Descriptor 4 did not cause an interrupt

Bit 13 – SD4DONE
Note: The hardware clears this bit upon receiving next trigger for Scan Descriptor 4.

Value	Description
1	Scan Descriptor 4 completed at least once

Value	Description
0	Scan Descriptor 4 did not complete

Bit 12 – SD4BUSY

Value	Description
1	Scan Descriptor 4 is in progress
0	Scan Descriptor 4 is not in progress

Bit 10 – SD3INT

Value	Description
1	Scan Descriptor 3 caused an interrupt
0	Scan Descriptor 3 did not cause an interrupt

Bit 9 – SD3DONE

The controller sets this bit if Scan Descriptor 3 completes at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 3.

Bit 8 – SD3BUSY

Value	Description
1	Scan Descriptor 3 is in progress
0	Scan Descriptor 3 is not in progress

Bit 6 – SD2INT

Value	Description
1	Scan Descriptor 2 caused an interrupt
0	Scan Descriptor 2 did not cause an interrupt

Bit 5 – SD2DONE The controller sets this bit if Scan Descriptor 2 completes at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 2.

Bit 4 – SD2BUSY

Value	Description
1	Scan Descriptor 2 caused an interrupt
0	Scan Descriptor 2 did not cause an interrupt

Bit 2 – SD1INT Scan Descriptor 1 caused an interrupt

Value	Description
1	Scan Descriptor 1 caused an interrupt
0	Scan Descriptor 1 did not cause an interrupt

Bit 1 – SD1DONE The controller sets this bit if Scan Descriptor 1 completed at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 1.

Bit 0 – SD1BUSY Scan Descriptor 1 is in progress

Value	Description
1	Scan Descriptor 1 is in progress
0	Scan Descriptor 1 is not in progress

37.7.4 CVD Results POS FIFO Read Register

Name: CVDRESH
Offset: 0x010
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						POS[18:16]		
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	POS[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
	POS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 18:0 – POS[18:0]

The accumulated result of the positive-side measurements. The controller supports up to 128x oversampling; therefore, each polarity can accumulate up to 19 bits when using 12-bit ADC (actual number of bits = ADCBITS + 7). The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the software will need to account for the left-shift of the result returned.

37.7.5 CVD Results NEG FIFO Read Register

Name: CVDRESL
Offset: 0x014
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						NEG[18:16]		
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	NEG[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
	NEG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 18:0 – NEG[18:0]

The accumulated result of the negative-side measurements. The controller supports up to 128x oversampling; therefore, each polarity can accumulate up to 18 bits when using 12-bit ADC (actual number of bits = ADCBITS + 7). The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the software will need to account for the left-shift of the result returned.

37.7.6 CVD Results Descriptor FIFO Read Register

Name: CVDRESD
Offset: 0x018
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	TXINDEX[4:0]						SDNUM[1:0]	
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	RXINDEX[4:0]						DELTA[17:16]	
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	DELTA[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
	DELTA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:27 – TXINDEX[4:0] Transmit Index of the result
If the Stride of the Scan Descriptor is more than one, the Transmit Index indicates the first one of the group.

Bits 25:24 – SDNUM[1:0] Scan Descriptor Number
These bits has the Scan Descriptor Number that generated the result.

Bits 23:19 – RXINDEX[4:0] Receive Index of the result
If the Stride of the Scan Descriptor is more than one, the Receive Index indicates the first one of the group.

Bits 17:0 – DELTA[17:0] Delta of the accumulated results of the negative-side and positive-side measurements
The controller supports up to 128x oversampling; therefore, each polarity can accumulate up to 19 bits when using 12-bit ADC. The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the software will need to account for the results returned. The DELTA presented is the 18 MSBs (including sign bit) of a signed subtraction of the two accumulators. The data will be in 2's complement form if the delta is negative.
Width Needed: ADCBITS + 7 + 1 (sign)
Width available: 18
delta_pre[ADCBITS+7:0] = signed'({1'b0, POS[ADCBITS+7-1:0]} - {1'b0, NEG[ADCBITS+7-1:0]})
DELTA[17:0] = delta_pre[ADC_BITS+7:ADC_BITS+7-17]
Note: Reading this register increments the FIFO read pointer, destroying the data in the previous two registers for NEG and POS absolute values. If the NEG and POS values are desired, those registers must be read before this register. If the absolute values are not required, bandwidth can be saved by reading only this descriptor register.

37.7.7 CVD Receive Index 0 Configuration

Name: CVDRX0
Offset: 0x080
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN3[5:0]					
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
			RXAN2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN1[5:0]					
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
			RXAN0[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN3[5:0] ANx/CVDR channel to use for RX Index 3

Bits 21:16 – RXAN2[5:0] ANx/CVDR channel to use for RX Index 2

Bits 13:8 – RXAN1[5:0] ANx/CVDR channel to use for RX Index 1

Bits 5:0 – RXAN0[5:0] ANx/CVDR channel to use for RX Index 0

37.7.8 CVD Receive Index 1 Configuration

Name: CVDRX1
Offset: 0x084
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN7[5:0]					
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
			RXAN6[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN5[5:0]					
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
			RXAN4[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN7[5:0] ANx/CVDR channel to use for RX Index 7

Bits 21:16 – RXAN6[5:0] ANx/CVDR channel to use for RX Index 6

Bits 13:8 – RXAN5[5:0] ANx/CVDR channel to use for RX Index 5

Bits 5:0 – RXAN4[5:0] ANx/CVDR channel to use for RX Index 4

37.7.9 CVD Transmit Index 0 Configuration

Name: CVDTX0
Offset: 0x0C0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXAN3[5:0]					
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
			TXAN2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXAN1[5:0]					
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
			TXAN0[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXAN3[5:0] ANx/CVDR channel to use for TX Index 3

Bits 21:16 – TXAN2[5:0] ANx/CVDR channel to use for TX Index 2

Bits 13:8 – TXAN1[5:0] ANx/CVDR channel to use for TX Index 1

Bits 5:0 – TXAN0[5:0] ANx/CVDR channel to use for TX Index 0

37.7.10 CVD Transmit Index 1 Configuration

Name: CVDTX1
Offset: 0x0C4
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXAN7[5:0]					
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
			TXAN6[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXAN5[5:0]					
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
			TXAN4[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXAN7[5:0] ANx/CVDR channel to use for TX Index 7

Bits 21:16 – TXAN6[5:0] ANx/CVDR channel to use for TX Index 6

Bits 13:8 – TXAN5[5:0] ANx/CVDR channel to use for TX Index 5

Bits 5:0 – TXAN4[5:0] ANx/CVDR channel to use for TX Index 4

37.7.11 CVD Scan Descriptor 0 Control 1

Name: CVDSD0C1
Offset: 0x100
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0TH[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	23	22	21	20	19	18	17	16
	SD0TH[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	15	14	13	12	11	10	9	8
	SD0TH[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
		SD0OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD0TH[23:0] Scan Descriptor Threshold

The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD0OVRSAMP[6:0] Scan Descriptor Over Sampling

Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

37.7.12 CVD Scan Descriptor 0 Control 2

Name: CVDSD0C2
Offset: 0x104
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXEND[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	23	22	21	20	19	18	17	16
	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXBEG[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	15	14	13	12	11	10	9	8
	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXEND[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
	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXBEG[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 29:24 – SD0TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD0TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value, then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD0TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX Indexes included in a single measurement.

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD0TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan.

Bits 13:8 – SD0RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD0RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value, then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD0RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement.

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD0RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

37.7.13 CVD Scan Descriptor 0 Control 3

Name: CVDSD0C3
Offset: 0x108
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0EN[1:0]				SD0BUF	SD0INTEN	SD0SELF	SD0MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD0ACQTIME[6:0]						
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	6	5	4	3	2	1	0
		SD0CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD0EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor
01	Executes the Scan Descriptor one time only, then clears the enable
10	Executes the Scan Descriptor and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to the next enabled descriptors.

Bit 27 – SD0BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC input
0	Does not use CVD buffer output as shared ADC input

Bit 26 – SD0INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan descriptor creates an interrupt if the accumulator threshold is met
0	Scan descriptor does not create an interrupt

Bit 25 – SD0SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements

Bit 24 – SD0MUT Scan Descriptor Mutual Mode

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD0ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for the ADC voltage to settle.

Bits 6:0 – SD0CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs

37.7.14 CVD Scan Descriptor 0 Time 2

Name: CVDSD0T2
Offset: 0x10C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD0CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD0OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD0POLTIME[6:0]						
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	6	5	4	3	2	1	0
		SD0CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD0CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair

Bits 22:16 – SD0OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair

Bits 14:8 – SD0POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair

Bits 6:0 – SD0CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. User must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires

37.7.15 CVD Scan Descriptor 1 Control 1

Name: CVDSD1C1
Offset: 0x110
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1TH[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	23	22	21	20	19	18	17	16
	SD1TH[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	15	14	13	12	11	10	9	8
	SD1TH[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
		SD1OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD1TH[23:0] Scan Descriptor Threshold
The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD1OVRSAMP[6:0] Scan Descriptor Over Sampling
Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

37.7.16 CVD Scan Descriptor 1 Control 2

Name: CVDSD1C2
Offset: 0x114
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXEND[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	23	22	21	20	19	18	17	16
	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXBEG[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	15	14	13	12	11	10	9	8
	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXEND[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
	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXBEG[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 29:24 – SD1TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD1TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD1TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX indices included in a single measurement.

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD1TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD1RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD1RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD1RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD1RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

37.7.17 CVD Scan Descriptor 1 Control 3

Name: CVDSD1C3
Offset: 0x118
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1EN[1:0]				SD1BUF	SD1INTEN	SD1SELF	SD1MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD1ACQTIME[6:0]						
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	6	5	4	3	2	1	0
		SD1CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD1EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD1BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD1INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD1SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD1MUT Scan Descriptor Mutual Mode

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD1ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle.

Bits 6:0 – SD1CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs.

37.7.18 CVD Scan Descriptor 1 Time 2

Name: CVDSD1T2
Offset: 0x11C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD1CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD1OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD1POLTIME[6:0]						
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	6	5	4	3	2	1	0
		SD1CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD1CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair.

Bits 22:16 – SD1OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair.

Bits 14:8 – SD1POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair.

Bits 6:0 – SD1CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. User must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

37.7.19 CVD Scan Descriptor 2 Control 1

Name: CVDSD2C1
Offset: 0x120
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2TH[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	23	22	21	20	19	18	17	16
	SD2TH[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	15	14	13	12	11	10	9	8
	SD2TH[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
		SD2OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD2TH[23:0] Scan Descriptor Threshold

The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD2OVRSAMP[6:0] Scan Descriptor Over Sampling

Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

37.7.20 CVD Scan Descriptor 2 Control 2

Name: CVDSD2C2
Offset: 0x124
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXEND[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	23	22	21	20	19	18	17	16
	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXBEG[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	15	14	13	12	11	10	9	8
	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXEND[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
	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXBEG[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 29:24 – SD2TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD2TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD2TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX indices included in a single measurement.

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD2TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD2RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD2RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD2RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD2RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

37.7.21 CVD Scan Descriptor 2 Control 3

Name: CVDSD2C3
Offset: 0x128
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2EN[1:0]				SD2BUF	SD2INTEN	SD2SELF	SD2MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD2ACQTIME[6:0]						
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	6	5	4	3	2	1	0
		SD2CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD2EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD2BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD2INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD2SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD2MUT Scan Descriptor Mutual Mode

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD2ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle

Bits 6:0 – SD2CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs

37.7.22 CVD Scan Descriptor 2 Time 2

Name: CVDSD2T2
Offset: 0x12C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD2CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD2OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD2POLTIME[6:0]						
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	6	5	4	3	2	1	0
		SD2CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD2CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair.

Bits 22:16 – SD2OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair.

Bits 14:8 – SD2POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair.

Bits 6:0 – SD2CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. The user must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

37.7.23 CVD Scan Descriptor 3 Control 1

Name: CVDSD3C1
Offset: 0x130
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3TH[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	23	22	21	20	19	18	17	16
	SD3TH[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	15	14	13	12	11	10	9	8
	SD3TH[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
		SD3OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD3TH[23:0] Scan Descriptor Threshold

The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD3OVRSAMP[6:0] Scan Descriptor Over Sampling

Determines the amount of oversampling done on each measurement

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

37.7.24 CVD Scan Descriptor 3 Control 2

Name: CVDSD3C2
Offset: 0x134
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXEND[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	23	22	21	20	19	18	17	16
	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXBEG[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	15	14	13	12	11	10	9	8
	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXEND[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
	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXBEG[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 29:24 – SD3TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD3TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD3TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX Indexes included in a single measurement

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD3TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD3RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD3RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD3RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indexes included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD3RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

37.7.25 CVD Scan Descriptor 3 Control 3

Name: CVDSD3C3
Offset: 0x138
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3EN[1:0]				SD3BUF	SD3INTEN	SD3SELF	SD3MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD3ACQTIME[6:0]						
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	6	5	4	3	2	1	0
		SD3CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD3EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then, clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD3BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD3INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD3SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD3MUT Scan Descriptor Mutual Mode

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD3ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle

Bits 6:0 – SD3CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the Charging state for internal or external capacitors and for TX outputs

37.7.26 CVD Scan Descriptor 3 Time 2

Name: CVDSD3T2
Offset: 0x13C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD3CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD3OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD3POLTIME[6:0]						
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	6	5	4	3	2	1	0
		SD3CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD3CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair

Bits 22:16 – SD3OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair

Bits 14:8 – SD3POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair

Bits 6:0 – SD3CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. The user must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

38. Analog Comparators (AC)

38.1 Overview

The Analog Comparator (AC) supports two individual comparators:

- One shared (with built-in supply voltage monitor (MVREF)) AC_CMP1
- One dedicated AC_CMP0

Each comparator (COMP) compares the voltage levels on two inputs and provides a digital output based on the comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input changes.

The input selection includes up to four shared analog port pins and several internal signals. Each Comparator Output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one pair of comparators, Comparator 0 (AC_COMP0) and Comparator 1 (AC_COMP1), as a pair.

They have identical behaviors but separate Control registers. Each pair can be set in the Window mode to compare a signal to a voltage range instead of a single voltage level.

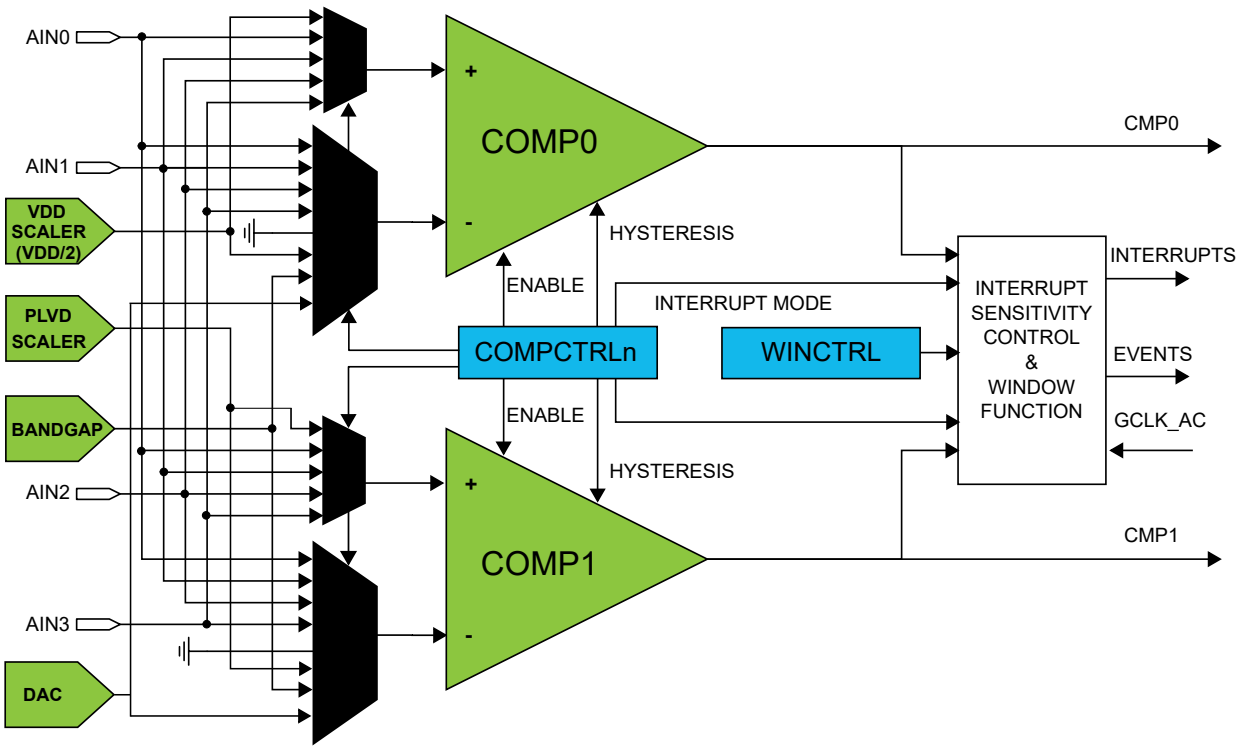
Note: This is not available in the 32-pin variant PIC32CX5109BZ31032 device.

38.2 Features

- Two Individual Comparators (Single Pair Configuration)
- Analog Comparator Outputs Available on Pins
  - Asynchronous or Synchronous
- Flexible Input Selection:
  - Up to four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - Programmable VDD scaler for AC_CMP1 (Shared with Programmable Low Voltage Detector (PLVD)) and fixed VDD/2 for AC_CMP0
  - DAC output
- Interrupt Generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window Function Interrupt Generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event Generation on:
  - Comparator output
  - Window function inside/outside window
- Optional Digital Filter on Comparator Output

38.3 Block Diagram

Figure 38-1. Analog Comparator Block Diagram



38.4 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

38.4.1 I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured as analog pins for AC_AINx inputs. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Table 38-1. I/O LINES

Signal	Peripheral Function
AC_AIN0	Comparator input
AC_AIN1	Comparator input
AC_AIN2	Comparator input
AC_AIN3	Comparator input
AC_CMP0	Comparator output
AC_CMP1	Comparator output

Note:

1. To get the analog comparator output on the I/O line, CFGCON1.CMP0_OE/CFGCON1.CMP1_OE needs to be set/enabled. See *System Configuration and Register Locking (CFG)* from Related Links.

Related Links

6. I/O Ports and Peripheral Pin Select (PPS)
22. System Configuration and Register Locking (CFG)

38.4.2 Power Management

The AC continues to operate in any Sleep mode (Idle, Standby Sleep) where the selected source clock is running. The AC's interrupts can be used to wake up the device from the sleep modes. Events connected to the event system can trigger other operations in the system without exiting the Standby Sleep mode.

38.4.3 Clocks

The AC bus clock (PB2_CLK) can be enabled and disabled in the Main Clock module, CRU (see *Clock and Reset Unit (CRU)* from Related Links), and the Analog Comparator module can be enabled or disabled via the PMD1 register. See *Peripheral Module Disable Register (PMD)* from Related Links.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC.

This generic clock is asynchronous to the bus clock (PB2_CLK). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. See *Synchronization* from Related Links.

Related Links

17. Clock and Reset Unit (CRU)
23. Peripheral Module Disable (PMD)
38.5.14. Synchronization

38.4.4 DMA

Not applicable.

38.4.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

38.4.6 Events

The events are connected to the Event System. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

Related Links

30. Event System (EVSYS)

38.4.7 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any ongoing comparison is completed. The AC can be forced to continue normal operation during debugging. See *DBGCTRL* register from Related Links. If the AC 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.

Related Links

38.7.9. DBGCTRL

38.4.8 Register Access Protection

All registers with write access can be write protected optionally by the PAC, except for the following registers:

- Control B register (CTRLB)

- Interrupt Flag register (INTFLAG)

Optional write protection by the 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.

38.4.9 Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. The pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, DAC must be configured and enabled prior to its use as a comparator input.

38.5 Functional Description

38.5.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive and '0' otherwise.

The individual comparators can be used independently (Normal mode) or paired to form a window comparison (Window mode).

38.5.2 Overview

The Analog Comparator (AC) supports two individual comparators:

- One shared (with built-in supply voltage monitor (MVREF)) AC_CMP1
- One dedicated AC_CMP0

Each comparator (COMP) compares the voltage levels on two inputs and provides a digital output based on the comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input changes.

The input selection includes up to four shared analog port pins and several internal signals. Each Comparator Output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one pair of comparators, Comparator 0 (AC_COMP0) and Comparator 1 (AC_COMP1), as a pair.

They have identical behaviors but separate Control registers. Each pair can be set in the Window mode to compare a signal to a voltage range instead of a single voltage level.

Note: This is not available in the 32-pin variant PIC32CX5109BZ31032 device.

38.5.3 Basic Operation

38.5.3.1 Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable protection is denoted by the Enable-Protected property in each individual register description.

38.5.3.2 Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled by writing a '0' to CTRLA.ENABLE. In addition to this AC module enable configuration, AC must also be enabled in the PMD module.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC are reset to their initial state, and the AC is disabled. See *CTRLA* register from Related Links.

Related Links

38.7.1. CTRLA

38.5.3.3 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See *Starting a Comparison* from Related Links.
- Fixed hysteresis
- Fixed speed of operation
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See *Selecting Comparator Inputs* from Related Links.
- Select the filtering option with COMPCTRLx.FLEN.
- Select the standby operation with the Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators but will not clear their COMPCTRLx.ENABLE bits.

Related Links

38.5.3.4. Starting a Comparison
38.5.4. Selecting Comparator Inputs

38.5.3.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, which is determined by the COMPCTRLx.SINGLE bit:

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in the Electrical Specifications. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the Single-Shot mode to chain further events in the system, regardless of the state of the comparator outputs. The Interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state regardless of whether the interrupt is enabled or not.

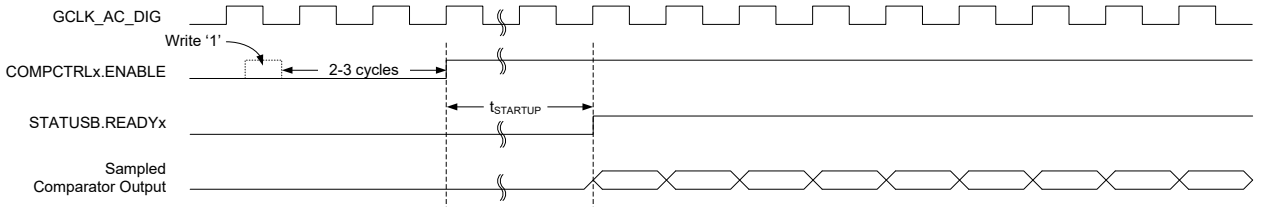
38.5.3.4.1 Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In the continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the following figure.

Figure 38-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep mode without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the CRU will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is, then, disabled again automatically unless configured to wake up the system from sleep.

38.5.3.4.2 Single-Shot

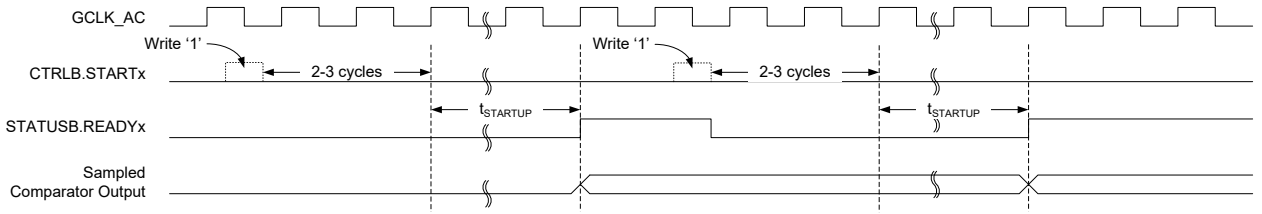
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled and, after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and will not clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in the following figure.

Figure 38-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the CRU will start GCLK_AC. The comparator is enabled and, after the start-up time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are, then, disabled again automatically unless configured to wake up the system from sleep.

38.5.4 Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one of the internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS).
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG).

In the case of using an external I/O pin, the selected pin must be configured for analog use in the GPIO by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL module (see *Configurable Custom Logic (CCL)* from Related Links), COMPCTRLx.OUT must be `0x1` or `0x2`.

Related Links

28. Configurable Custom Logic (CCL)

38.5.5 Window Operation

The comparator pair can be configured to work together in Window mode. In this mode, a voltage range is defined and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

Notes:

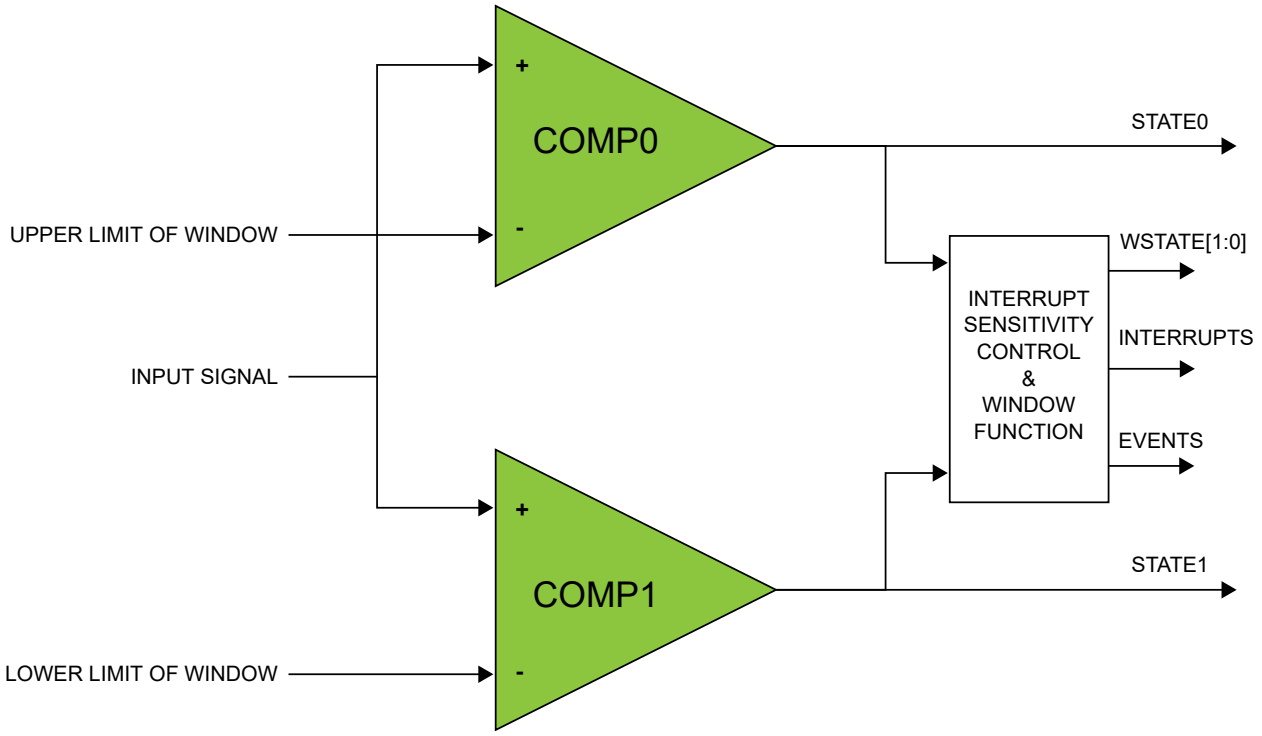
1. Both comparators must be enabled (COMPCTRLx.ENABLE = 1) before the Window mode is enabled (WINCTRL.WEN0 = 1).
2. Window mode must be first disabled (WINCTRL.WEN0 = 0) before both comparators are disabled (COMPCTRLx.ENABLE = 0).

To physically configure the comparators for Window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In the figure below, COMP0 defines the upper limit and COMP1 defines the lower limit of the window as shown, but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during Window mode.

When the comparators are configured for Window mode and Single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise, either peripheral event can start a measurement. The STATUSA.WSTATE bit field must be read only after STATUSB.READY bits of both the related comparators are set.

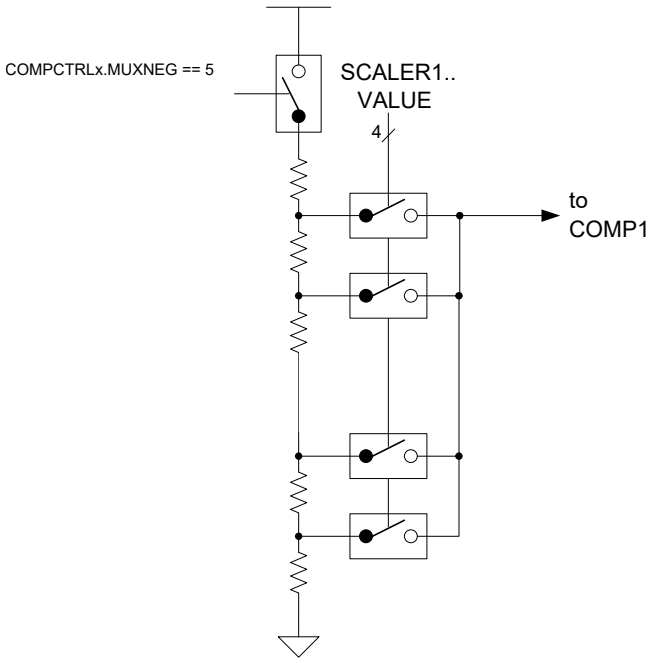
Figure 38-4. Comparators in Window Mode



38.5.6 VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage with 16 levels. The programmable VDD scaler (PLVD Scaler) is available for AC_CMP1 only. AC_CMP0 uses a fixed VDD/2 reference. The scaler of a comparator is enabled when the Negative Input Mux bit field or the Positive Input Mux in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to 0x5 and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the SCALER1 registers (SCALER1.VALUE) using the VDD resistor ladder.

Figure 38-5. PLVD Scaler



38.5.7 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For Continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 38-6. For Single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 38-7.

Figure 38-6. Continuous Mode Filtering

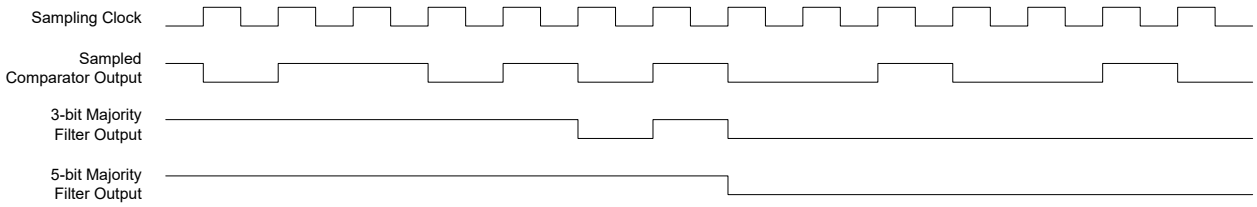
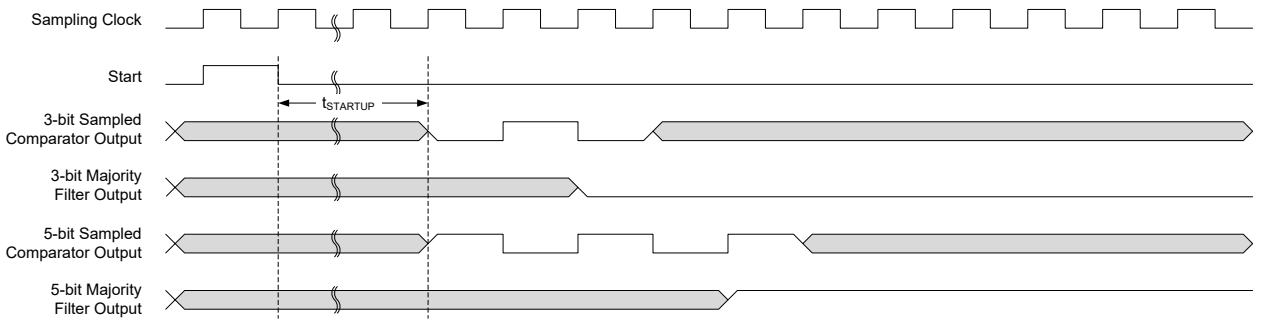


Figure 38-7. Single-Shot Filtering



During Sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during Sleep modes, or the resulting interrupt/event may be generated incorrectly.

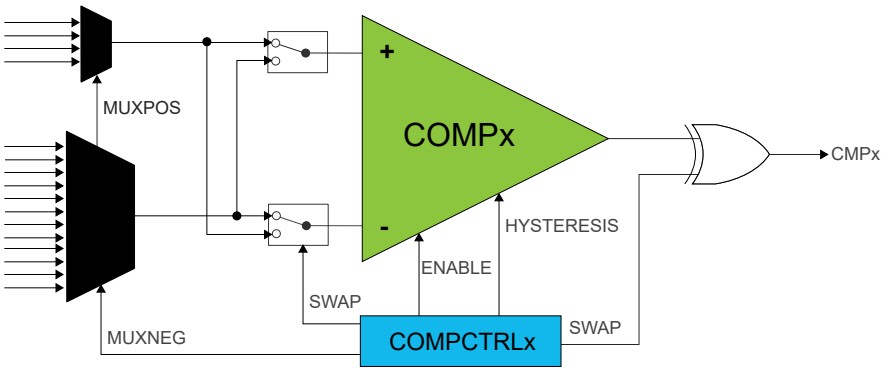
38.5.8 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). To get the analog comparator output on the I/O line, CFGCON1.CMP0_OE/CFGCON1.CMP1_OE also needs to be set or enabled. This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the GCLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding AC_CMPx pin. The AC_CMP1 can be output on an alternate pin by configuring the CFGCON0.ACCMP1_ALTEN configuration.

38.5.9 Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 38-8. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 38-8. Input Swapping for Offset Compensation



38.5.10 DMA Operation

Not applicable.

38.5.11 Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status
- Window (WIN0): Indicates a change in the window status

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

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 one to the corresponding bit in the Interrupt Enable Set (INTENSET) register and disabled by writing a one 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 interrupt flag is cleared, the interrupt is disabled or the AC is Reset. See *INTFLAG* register from Related Links for details on how to clear interrupt flags. All interrupt requests from the peripheral are OR'ed together on the 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. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)
38.7.6. INTFLAG

38.5.12 Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on an input event. Writing a zero to this bit disables the corresponding action on an input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

Related Links

30. Event System (EVSYS)

38.5.13 Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY

is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 38-2.

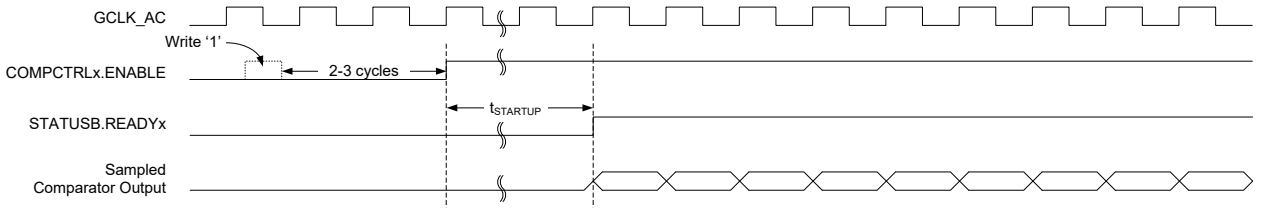
Table 38-2. Sleep Mode Operation

COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

38.5.13.1 Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

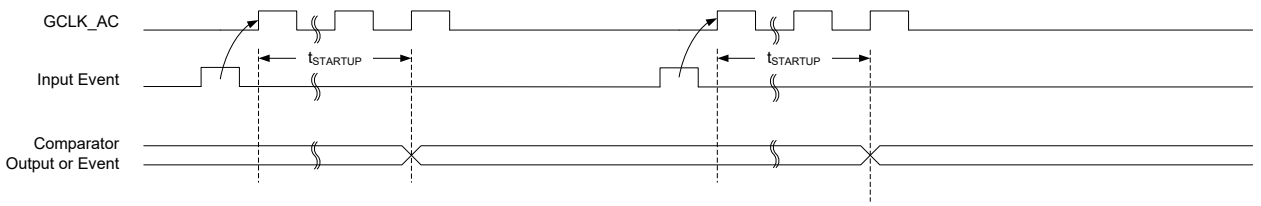
Figure 38-9. Continuous Mode SleepWalking



38.5.13.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the CRU will start GCLK_AC. The comparator is enabled and, after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated as shown in the following figure. The comparator and GCLK_AC are, then, disabled again automatically unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 38-10. Single-Shot SleepWalking



38.5.14 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 the Control register (CTRLA.SWRST)
- Enable bit in the Control register (CTRLA.ENABLE)
- Enable bit in the Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write synchronization is denoted by the Write-Synchronized property in the register description.

38.6 Register Summary

See the *AC* 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	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							START1	START0
0x02	EVCTRL	7:0				WINEO0			COMPEO1	COMPEO0
		15:8			INVEI1	INVEI0			COMPEI1	COMPEI0
0x04	INTENCLR	7:0				WIN0			COMP1	COMP0
0x05	INTENSET	7:0				WIN0			COMP1	COMP0
0x06	INTFLAG	7:0				WIN0			COMP1	COMP0
0x07	STATUSA	7:0			WSTATE0[1:0]				STATE1	STATE0
0x08	STATUSB	7:0							READY1	READY0
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B ... 0x0C	Reserved									
0x0D	SCALER1	7:0					VALUE[3:0]			
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP		MUXPOS[2:0]				MUXNEG[2:0]	
		23:16								
		31:24			OUT[1:0]				FLEN[2:0]	
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP		MUXPOS[2:0]				MUXNEG[2:0]	
		23:16								
		31:24			OUT[1:0]				FLEN[2:0]	
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
		15:8								
		23:16								
		31:24								

Related Links

8. Product Memory Mapping Overview

38.7 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 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 synchronized when read and/or written. Synchronization is denoted by the Write-Synchronized or the Read-Synchronized property in each individual register description. See *Synchronization* 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

38.4.8. Register Access Protection
38.5.14. Synchronization

38.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	W
Reset							0	0

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately, and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.
Note: To avoid spurious interrupts from multiple enable/disable cycles of AC, use the SWRST bit to reset the comparator module.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

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 AC to their initial state and the AC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

38.7.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – STARTx Comparator x Start Comparison
Writing a '0' to this field has no effect.
Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If Comparator x is not implemented or if it is not enabled in Single-shot mode, writing a '1' has no effect.
This bit always reads as '0'.

38.7.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEI1	INVEI0			COMPEI1	COMPEI0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEO1	COMPEO0
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 12, 13 – INVEIx Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 8, 9 – COMPEIx Comparator x Event Input

Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.

These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 4 – WINEO0 Window 0 Event Output Enable

These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 0, 1 – COMPEOx Comparator x Event Output Enable

These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

38.7.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
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 register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

38.7.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

38.7.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and generates an interrupt if INTENCLR/SET.WINx is also '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 0, 1 – COMPx Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as '0'.
This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and generates an interrupt if INTENCLR/SET.COMPx is also '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator x interrupt flag.

38.7.7 Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATE1	STATE0
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the Window 0 mode is enabled.
These values may change during start-up and measurement cycles. When polling for sample completion, use both the STATUSB.READYx bits to signal completion.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3	—	Reserved

Bits 0, 1 – STATEx Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when the STATUSB.READYx bit is one.

38.7.8 Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							READY1	READY0
Access							R	R
Reset							0	0

Bits 0, 1 – READYx Comparator x Ready

Value	Description
0	This bit is cleared when the Comparator x output is not ready.
1	This bit is set when the Comparator x output is ready.

38.7.9 Debug Control

Name: DBGCTRL
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

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 bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison completes before AC is halted.
1	The AC continues normal operation when the CPU is halted by an external debugger.

38.7.10 Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0] Window 0 Interrupt Selection
These bits configure the Interrupt mode for the comparator Window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0 Window 0 Mode Enable

Value	Description
0	Window mode is disabled.
1	Window mode is enabled.

38.7.11 Scaler 1

Name: SCALER1
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

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

Bits 3:0 – VALUE[3:0] Scaler Value
These bits define the scaling factor for channel 1 of the VDD voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{\mathrm{SCALE}} = V_{DD} \times \left(\frac{R\_Bottom}{R\_Total}\right)$$

Where, R_Total = 900.
Refer to the following table for R_Bottom for different VALUE[3:0]
For example, V_{SCALE} for VALUE[3:0] = 0x02 at V_{DD} = 3.3V

$$V_{\mathrm{SCALE}} = 3.3 \times \left(\frac{598.5}{900}\right) = 2.1945V$$

Table 38-3. Scaler Value

Value[3:0]	R_Bottom
0x0	Reserved
0x1	Reserved
0x2	598.5
0x3	634.5
0x4	306
0x5	324
0x6	328.5
0x7	360
0x8	387
0x9	400.5
0xA	432
0xB	450
0xC	468
0xD	490.5
0xE	481.5
0xF	External Reference on LVDIN pin

38.7.12 Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[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	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0] Output

These bits configure the output selection for Comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is '0'.

Note: For internal use of the comparison results by the CCL, this must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0] Filter Length

These bits configure the filtering for Comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is '0'.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bit 15 – SWAP Swap Inputs and Invert

This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is '0'.

These bits are not synchronized.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0] Positive Input Mux Selection

These bits select which input is connected to the positive input of Comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	AC_AIN0
0x1	PIN1	AC_AIN1
0x2	PIN2	AC_AIN2
0x3	PIN3	AC_AIN3
0x4	VSCALE	VDD scaler
0x5–0x7	—	Reserved

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection

These bits select which input is connected to the negative input of Comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler
0x6	BANDGAP	Internal bandgap voltage
0x7	DAC	DAC output

Bit 6 – RUNSTDBY Run in Standby

This bit controls the behavior of the comparator during Standby Sleep mode.
This bit is not synchronized

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0] Interrupt Selection

These bits select the condition for Comparator n to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (Single-shot mode only)

Bit 2 – SINGLE Single-Shot Mode

This bit determines the operation of Comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous Measurement mode.
1	Comparator n operates in Single-shot mode.

Bit 1 – ENABLE Enable

Writing a '0' to this bit disables Comparator n.
Writing a '1' to this bit enables Comparator n.
Due to synchronization, there is a delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE reads back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.
Writing a '1' to COMPCTRLn.ENABLE prevents further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to '0' and the write is synchronized.

38.7.13 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
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
				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 3, 4 – COMPCTRLx COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.
This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

Bit 2 – WINCTRL WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.
This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

39. Digital-to-Analog Converter (DAC)

39.1 Overview

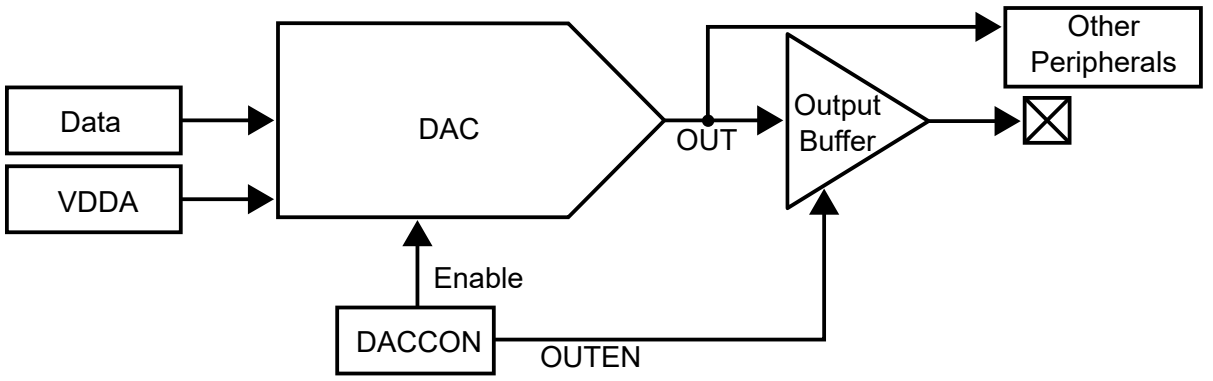
The PIC32CX-BZ3 7-bit Digital-to-Analog Converter (DAC), converts a digital value written to the Data (DACCODE.DATA) register to an analog voltage. The conversion range is between GND and AVDD. The DAC has one continuous time output with high drive capabilities. The application can start the DAC conversion by writing to the Data (DACCODE.DATA) register and enabling the DAC Module DACCON.EN bit to '1'.

39.2 Features

- Reference Voltage is Fixed to VDDA
- Seven Bits of Programmable Granularity Provided for Output Voltage with Six-Bit Accuracy
- Output is High Impedance in the Disabled State

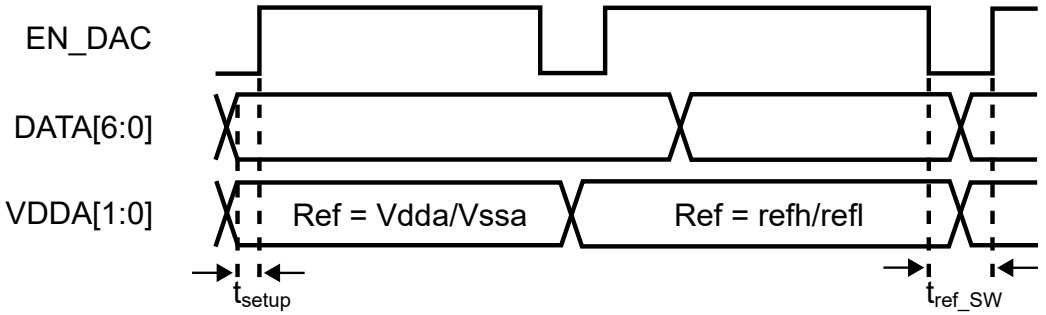
39.3 Block Diagram

Figure 39-1. Block Diagram



39.4 DAC Timing Diagram

Figure 39-2. DAC Timing Diagram



39.5 Functional Description

39.5.1 Initialization

The following steps are necessary to operate the DAC:

- The reference voltage fixed to VDDA.

- Configure the usage of the DAC output:
  - Configure an internal peripheral to use the DAC output. Refer to the documentation of the respective peripherals.
  - Enable the output to a pin by writing a '1' to the Output Buffer Enable (OUTEN) bit. The Port peripheral needs to disable the input for the DAC pin.
- Write an initial digital value to the Data (DACCODE.DATA) register.
- Enable the DAC by writing a '1' to the ENABLE bit in the Control A (DACCON.EN) register.

39.5.2 Operation

39.5.2.1 Enabling, Disabling and Resetting

The user enables the DAC by writing a '1' to the ENABLE bit in the DACCON (DACCON.EN) register and disables it by writing a '0' to this bit.

39.5.2.2 Starting a Conversion

When the ENABLE bit in the DACCON (DACCON.EN) register is written to '1', a conversion starts as soon as the Data (DACCODE.DATA) register is written.

When the ENABLE bit in DACCON.EN is written to '0', writing to the Data register does not trigger a conversion. Instead, the conversion starts when the ENABLE bit in DACCON.EN is written to '1'.

39.5.2.3 DAC as a Source for Internal Peripheral

The analog output of the DAC can be internally connected to other peripherals when the ENABLE bit in the DACCON (DACCON.EN) register is written to '1'. When the DAC analog output is only being used internally, the Output Buffer Enable (DACCON.OUTEN) bit in DACCON can be '0'.

39.5.2.4 DAC Output on Pin

In the DACCON (DACCON.OUTEN) register, the user can connect the analog output of the DAC to a pin by writing a '1' to the Output Buffer Enable (OUTEN) bit. The pin used by the DAC must have the input disabled from the Port peripheral. There is an output buffer between the DAC output and the pin, which ensures the analog value does not depend on the load of the pin. The output buffer can only source current; it has very limited sinking capability.

39.5.2.5 Low Power Mode Operation

If DAC conversion stops in Low Power mode, DAC is also disabled to reduce power consumption. When exiting Low Power mode, DAC is enabled again; therefore, a certain start-up time is required before starting a new conversion.

- When LPRCEN = 1, the DACCON2 register is not used.
- To set a clock request for LPRC, write a '1' to the DACCON.LPRCEN bit in the DACCON register.
- SnH mode must be used with OUTEN. This mode does not support an internal peripheral.

39.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DACCON	7:0	SNH	OUTEN	LPRCEN(1)					
		15:8	EN							
		23:16								
		31:24								
0x04	DACCODE	7:0		DATA[6:0]						
		15:8								
		23:16								
		31:24								
0x08	DACCON2	7:0	WIDTH[7:0]							
		15:8							WIDTH[9:8]	
		23:16	PERIOD[7:0]							
		31:24	PRESCALAR[2:0]						PERIOD[9:8]	

39.6.1 DACCON – Control Register

Name: DACCON
Offset: 0x00
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
	EN							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	SNH	OUTEN	LPRCEN(1)					
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bit 15 – EN Enable

Value	Description
1	The peripheral is enabled
0	The peripheral is disabled

Bit 7 – SNH Sample and Hold Circuitry

To set the clock request for LPRC, write a '1' to this bit when the LPRCEN bit is set.

Value	Description
1	DAC operation in Low Power mode
0	DAC operation in Normal mode

Bit 6 – OUTEN Will Enable the Output Buffer

Value	Description
1	Output buffer enabled
0	Output buffer disabled

Bit 5 – LPRCEN(1) LPRC Clock Enabled for SnH Mode

Value	Description
1	LPRC clock enabled
0	LPRC clock not enabled

Note:

1. SnH clock to DAC can

39.6.2 DACCODE – DAC Code Register

Name: DACCODE
Offset: 0x04
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
		DATA[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:0 – DATA[6:0] DAC Data
Digital value to be converted to analog voltage. 7-bit data that needs conversion is written here.

39.6.3 DACCON2 – DAC Config Register(1)

Name: DACCON2
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	PRESCALAR[2:0]						PERIOD[9:8]	
Access	R/W	R/W	R/W				R/W	R/W
Reset	0	0	0				0	0

Bit	23	22	21	20	19	18	17	16
	PERIOD[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	15	14	13	12	11	10	9	8
							WIDTH[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	WIDTH[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 31:29 – PRESCALAR[2:0] Prescaling Factor for SnH Clock
0x0 - Sampling clock is SnH clock directly
0x1 - Sampling clock is SnH clock/2
0x2 - Sampling clock is SnH clock/4
0x3 - Sampling clock is SnH clock/8
0x4 - Sampling clock is SnH clock/16
0x5 - Sampling clock is SnH clock/32
0x6 - Sampling clock is SnH clock/64
0x7 - Sampling clock is SnH clock/128

Bits 25:16 – PERIOD[9:0] SnH Clock Period
Tperiod = prescaler_clk_period * (DACCON2[PERIOD] + 1)

Bits 9:0 – WIDTH[9:0] SnH Clock Width
Twidth = prescaler_clk_period * (DACCON2[WIDTH] + 1)

Note:

1. The DACCON2 register is applicable when SnH clock is pb_clk. It is not applicable when SnH = 0 or when LPRCEN = 0.

40. Timer/Counter (TC)

40.1 Overview

There are up to eight TC peripheral instances.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter, together with the compare/capture channels, can be configured to time stamp input events or I/O pin edges, allowing for capturing of frequency and/or pulse width.

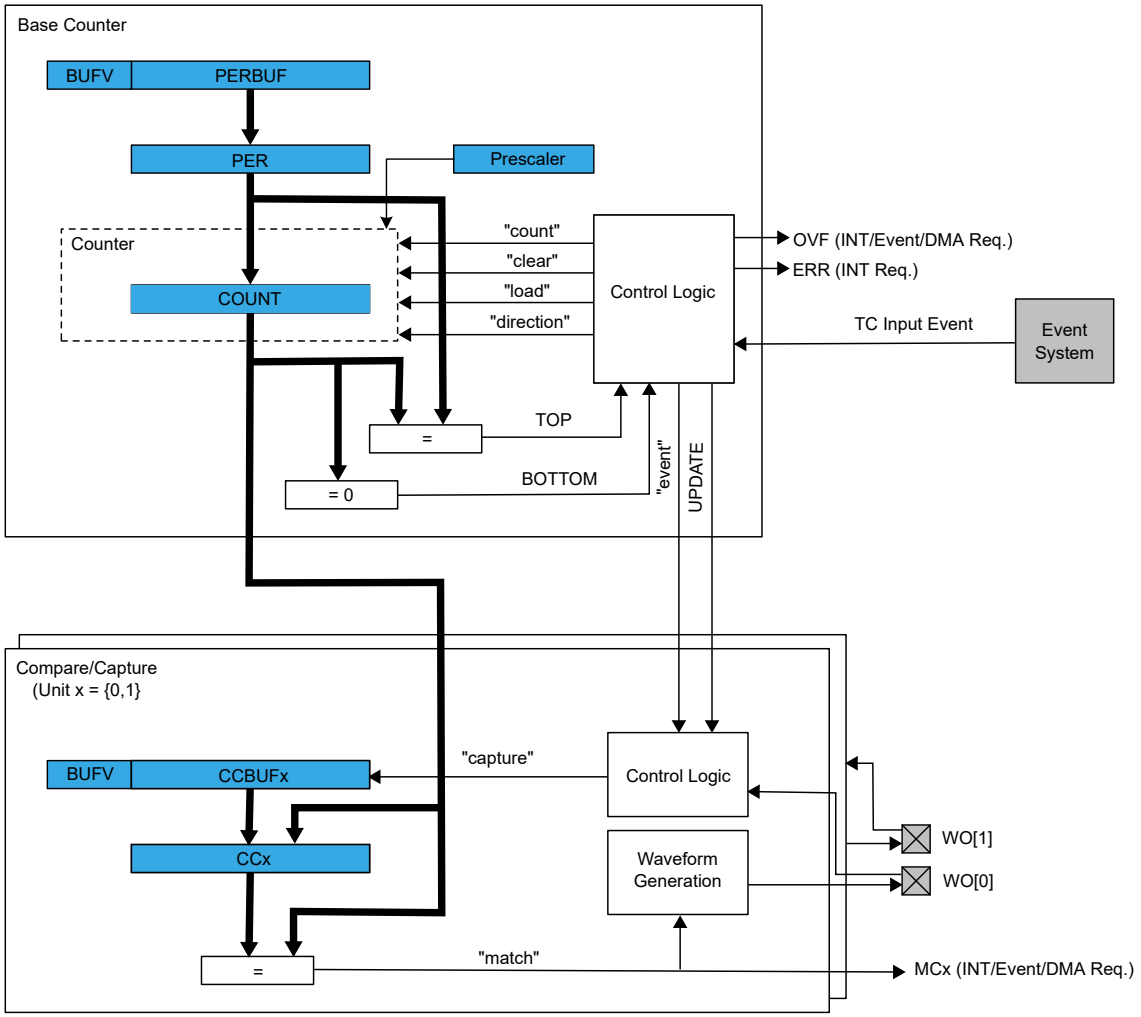
A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

40.2 Features

- Selectable Configuration
  - 8-, 16- or 32-bit TC operation with compare/capture channels
- Two Compare/Capture Channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform Generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input Capture
  - Event / IO pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
  - Minimum and maximum capture
- One Input Event
- Interrupts/Output Events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal Prescaler
- DMA support

40.3 Block Diagram

Figure 40-1. Timer/Counter Block Diagram



40.4 Signal Description

Table 40-1. Signal Description for TC

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

For details on the pin mapping for this peripheral, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links. One signal can be mapped on several pins.

Related Links

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

40.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

40.5.1 I/O Lines

To use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT). See *Port Register Summary* from Related Links.

Related Links

6.10. Port Register Summary

40.5.2 Power Management

This peripheral can continue to operate in any sleep mode (Idle, Standby Sleep) where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

40.5.3 Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Clock and Reset Unit (CRU).

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains.

Note: Two instances of the TC can share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. See *Clock and Reset Unit (CRU)* from Related Links.

Related Links

17. Clock and Reset Unit (CRU)

40.5.4 DMA

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

Related Links

26. Direct Memory Access Controller (DMAC)

40.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

40.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

30. Event System (EVSYS)

40.5.7 Debug Operation

When the CPU is halted in Debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging. For more details, see *DBGCTRL* from Related Links.

Related Links
40.8.11. DBGCTRL

40.5.8 Register Access Protection

Registers with write access can be optionally write protected by the PAC, except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

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.

40.5.9 Analog Connections

Not applicable.

40.6 Functional Description

40.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 40-2. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
ZERO	The counter is ZERO when it contains all zeros.
MAX	The counter reaches MAX when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g., counting external events).
CC	For compare operations, the CC are referred to as “compare channels.” For capture operations, the CC are referred to as “capture channels.”

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

For optimized timing, the CCx and CCBUFx registers share a common resource. When writing into CCBUFx, lock the access to the corresponding CCx register (SYNCBUSY.CCX = 1) until the CCBUFx register value is not loaded into the CCx register (BUFVx == 1). Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

The Counter register (COUNT) and the Compare and Capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with corresponding MAX values. Mode settings (CTRLA.MODE) determine the maximum range of the Counter register.

In 8-bit mode, a Period Value (PER) register and its Period Buffer Value (PERBUF) register are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match, the TC can request DMA transactions, or generate interrupts or events for the Event System.

In a compare operation, the counter value is continuously compared to the values in the CCx registers. In the case of a match, the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

Related Links

40.6.2.6.1. Waveform Output Operations

40.6.2 Basic Operation

40.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE = 0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Writing to enable-protected bits and setting the CTRLA.ENABLE bit can be performed in a single 32-bit access of the CTRLA register. Writing to enable-protected bits and clearing the CTRLA.ENABLE bit cannot be performed in a single 32-bit access.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock if not already enabled by default (PB1_CLK).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   - If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).
5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction down (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or I/O pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

40.6.2.2 Enabling, Disabling and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disabled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. See *CTRLA* from Related Links.

The TC must be disabled before the TC is reset to avoid undefined behavior.

Related Links

40.8.1. CTRLA

40.6.2.3 Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

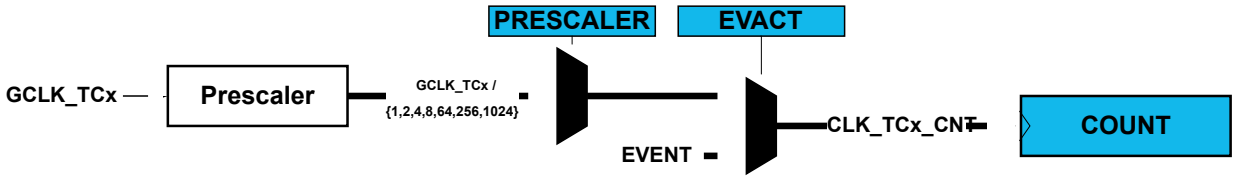
If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCx_CNT.

Figure 40-2. Prescaler



40.6.2.4 Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8 – The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16 – 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32 – 32-bit mode is achieved by pairing two 16-bit TC peripherals. TC0 is paired with TC1, and TC2 is paired with TC3. TC4 is paired with TC5 and TC6 with TC7 for the COUNT32 mode. When paired, the TC peripherals are configured using the registers of the even-numbered TC (TC0, TC2, TC4 or TC6, respectively).

  The TC bus clocks (PB1_CLK) for both host and client TCs need to be enabled.

  The odd-numbered partner (TC1, TC3, TC5 or TC7 respectively) acts as a client TC, and the Client bit in the Status register (STATUS.CLIENT) is set. The register values of a client TC will not reflect the registers of the 32-bit counter. Writing to any of the client registers will not affect the 32-bit counter. Normal access to the client COUNT and CCx registers is not allowed.

40.6.2.5 Counter Operations

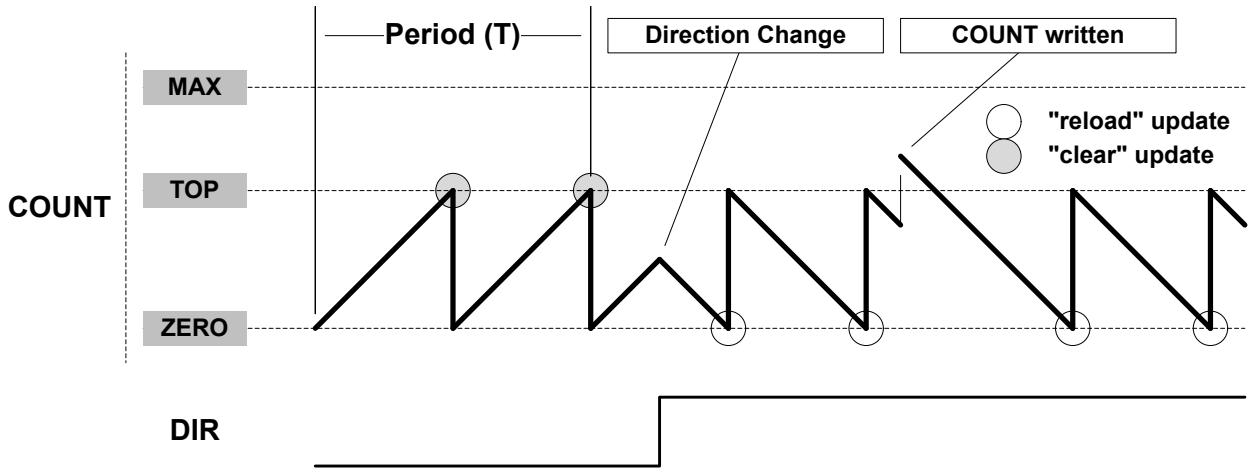
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TCx_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e., a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed when the counter is running. See also the following figure.

Figure 40-3. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

40.6.2.5.1 Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter is set to bottom/top depending on direction: up/down. All waveforms are cleared, and the Stop bit in the Status register is set (STATUS.STOP).

40.6.2.5.2 Re-Trigger Command and Event Action

A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER) or from an event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during the counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger

command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on the corresponding following event.

40.6.2.5.3 Count Event Action

The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Note: If this operation mode is selected, PWM generation is not supported.

40.6.2.5.4 Start Event Action

The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

40.6.2.6 Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). See *Double Buffering* from Related Links. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Related Links

40.6.2.7. Double Buffering

40.6.2.6.1 Waveform Output Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally, invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Peripheral Pin Select (PPS). See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.
   Note: The event must not be used when the compare channel is set in the waveform output operating mode.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) is set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt and/or event can be generated on a comparison match if enabled. The same condition generates a DMA request.

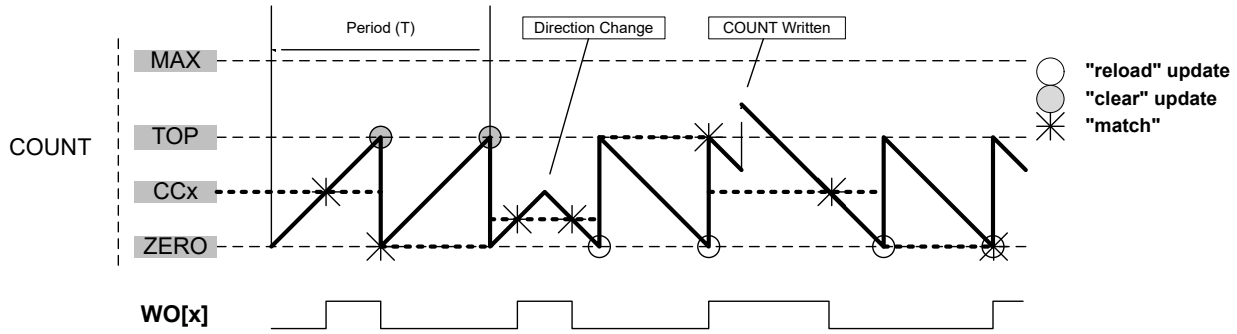
There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This influences how the waveform is generated and imposes restrictions on the top value. The configurations are:

- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, TOP is determined by the Period register (PER). In the 8-bit Counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In the 16- and 32-Bit Counter modes, TOP is fixed to the maximum (MAX) value of the counter.

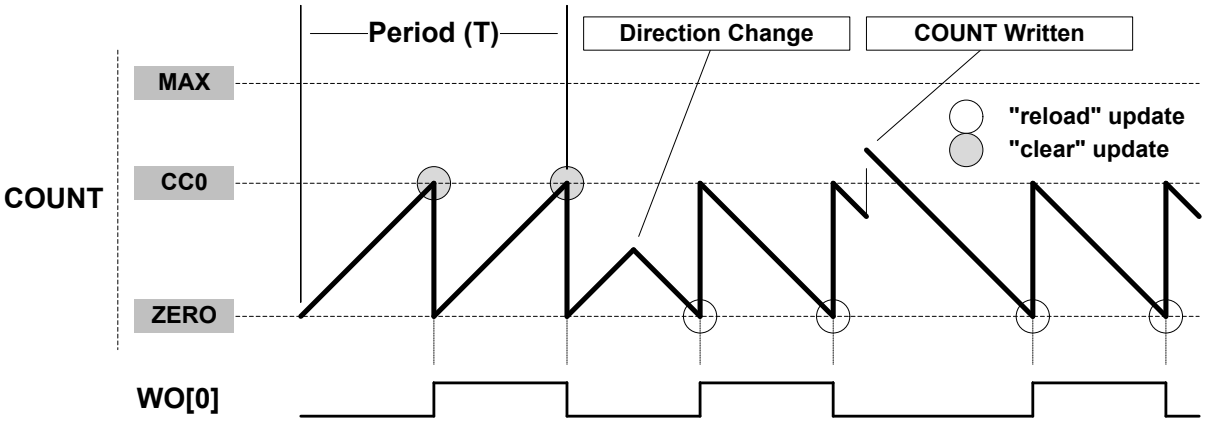
Normal Frequency Generation (NFRQ)
For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for the 8-bit Counter mode and MAX for the 16- and 32-Bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

Figure 40-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)
For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each Update condition.

Figure 40-5. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)
NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values and cleared on compare match between

COUNT and CCx register values. When down-counting, the WO[x] is cleared at the start or compare match between the COUNT and ZERO values and set on the compare match between the COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

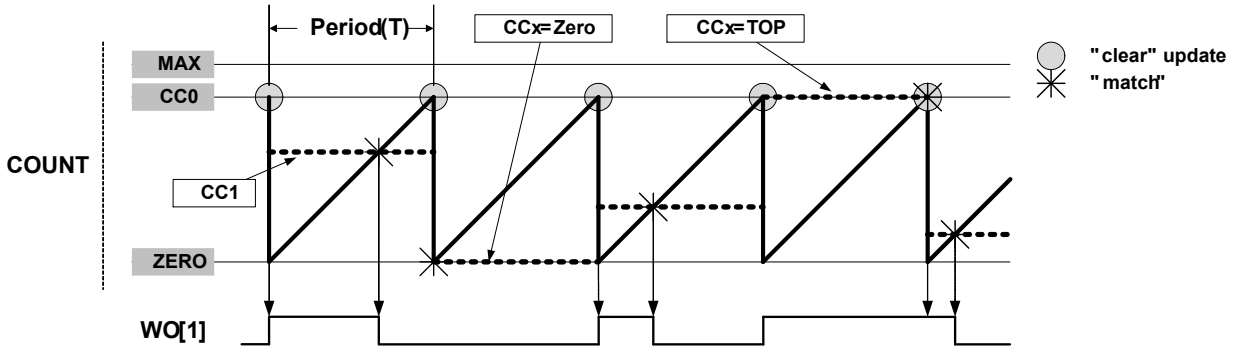
The PWM frequency (f_{PWM_SS}) depends on the TOP value and the peripheral clock frequency (f_{GCLK_TC}) and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TC}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)
In MPWM, the output of WO[1] is depending on CC1 as illustrated in the following figure. On every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 40-6. Match PWM Operation



The following table lists the Update Counter and Overflow Event/Interrupt Generation conditions in different operation modes.

Table 40-3. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

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

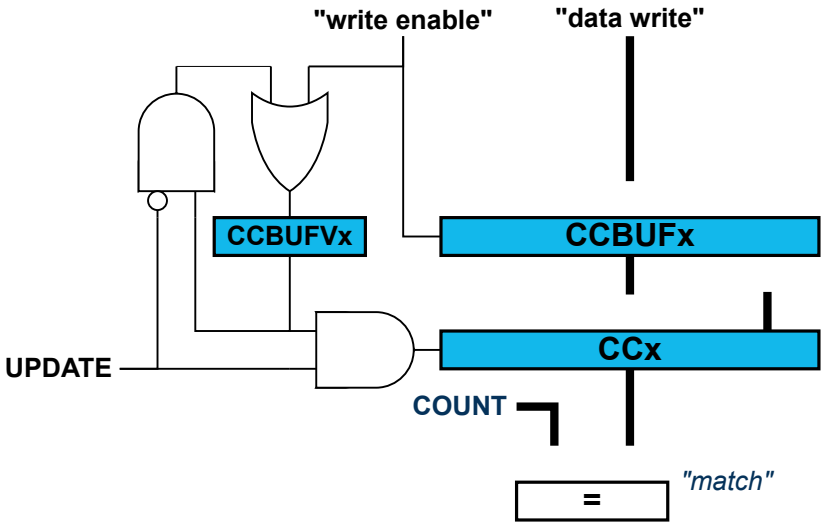
40.6.2.7 Double Buffering

The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
Note: The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

Figure 40-7. Compare Channel Double Buffering



Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

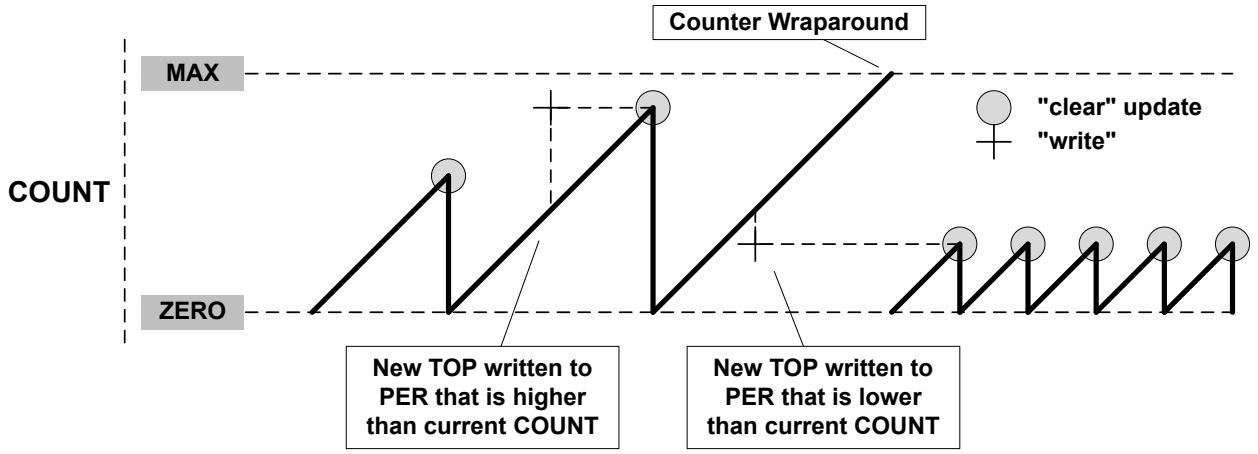
Note: In NFRQ, MFRQ or PWM, down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continuously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), which is available in 8-bit mode. Any period update on registers (PER or CCx) is effective after the synchronization delay.

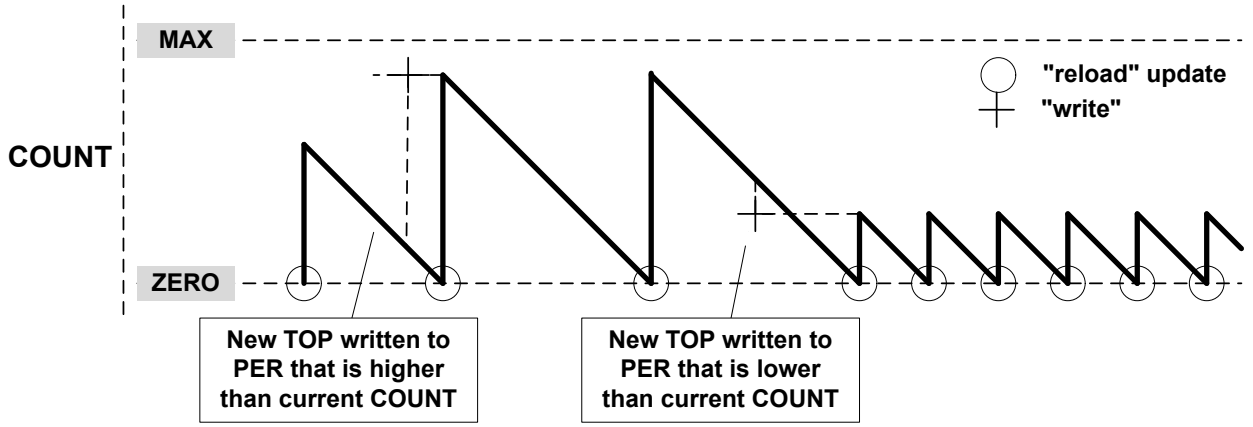
A counter wraparound can occur in any operation mode when up-counting without buffering (see the following figure).

Figure 40-8. Unbuffered Single-Slope Up-Counting Operation



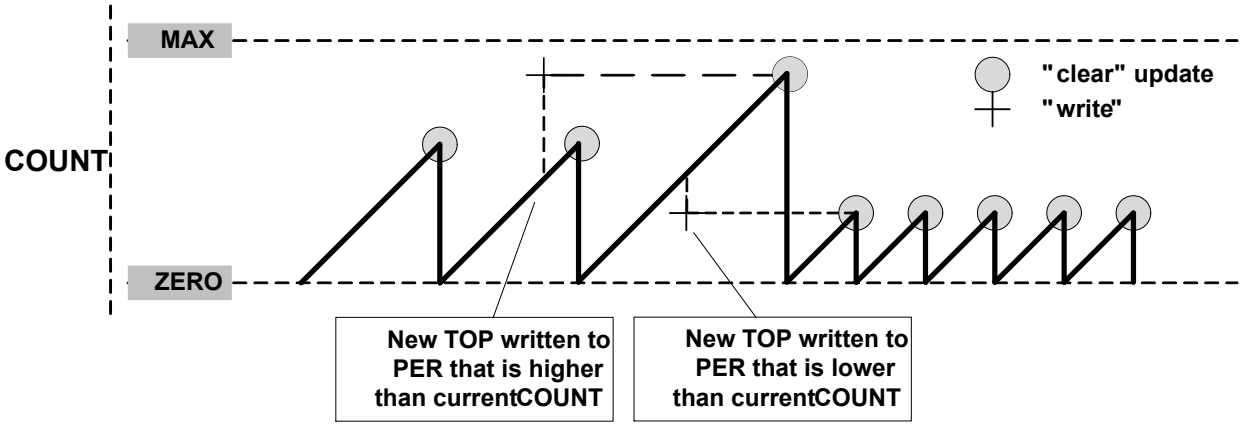
COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 40-9. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in the following figure. This prevents wraparound and the generation of odd waveforms.

Figure 40-10. Changing the Period Using Buffering



40.6.2.8 Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by the input event line TC_EV or by the asynchronous I/O pin WO[x] for each capture channel or by a TC event. To enable the capture from the input event line, the Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be writtento '1'. To enable the capture from the I/O pin, the Capture On Pin x Enable bit in the CTRLA register (CTRLA.COPENx) must be written to '1'.

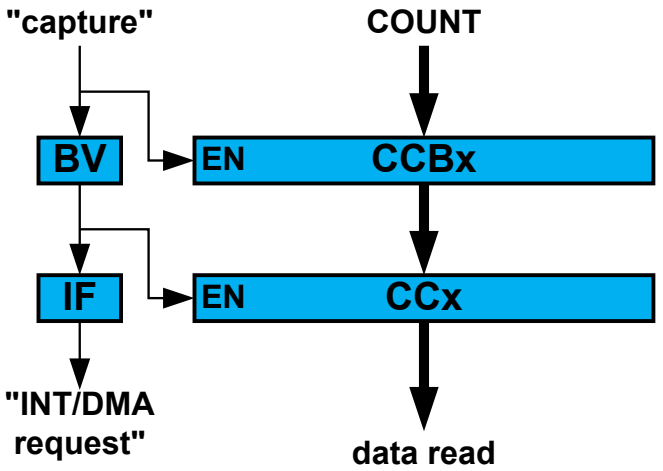
Notes:

1. Capture on I/Os is only possible in the Event and Time-Stamp capture action modes. Other modes can only use internal events. (If I/O toggling is needed in other modes, the I/Os edge must be configured for generating internal events).
2. Capture on an event from the Event System is possible in Event, PPW/PWP/PW and Time-Stamp capture action modes. In this case, the event system channels must be configured to operate in an asynchronous mode of operation.
3. Depending on CTRLA.COPENx, channel x can be configured for I/Os or internal event capture (both are mutually exclusive). One channel can be configured for I/O capture, while the other uses internal event capture.

By default, a capture operation is done when a rising edge is detected on the input signal. Capture on the falling edge is available, its activation is dependent on the input source:

- When the channel is used with an I/O pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

Figure 40-11. Capture Double Buffering



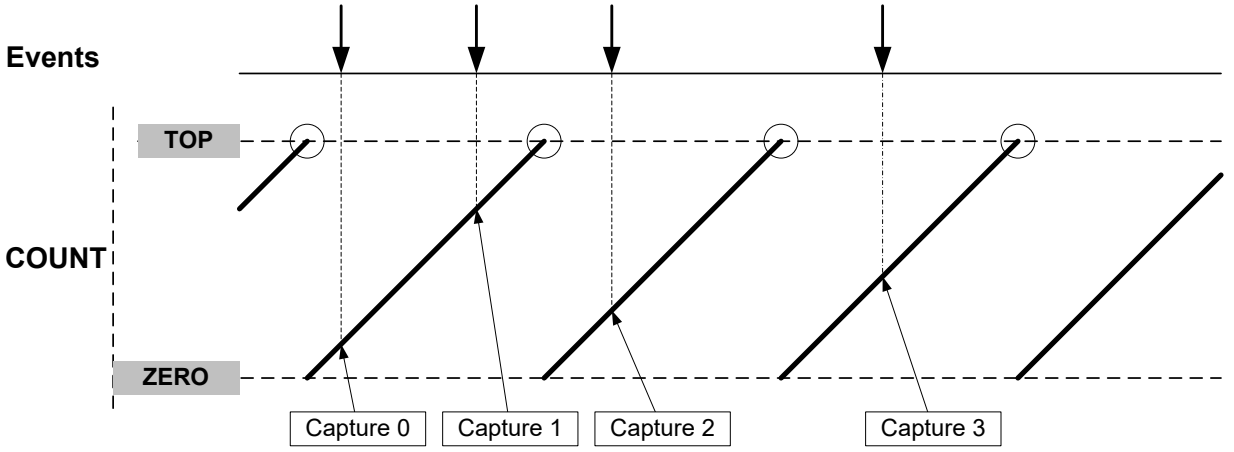
For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value cannot be read; all captured data must be read from the CCx register.

Note:
When up-counting (CTRLBSET.DIR = 0), counter values lower than '1' cannot be captured. To capture the full range including value '0', the TC must be in the Down-counting mode (CTRLBSET.DIR = 0).

40.6.2.8.1 Event Capture Action on Events or I/Os

The compare and capture channels can be used as input capture channels to capture events from the Event System or I/O pins and give them a timestamp. This mode is selected when EVTCTRL.EVACT is configured either as OFF, RETRIGGER, COUNT or START. The following figure shows four capture events for one capture channel.

Figure 40-12. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

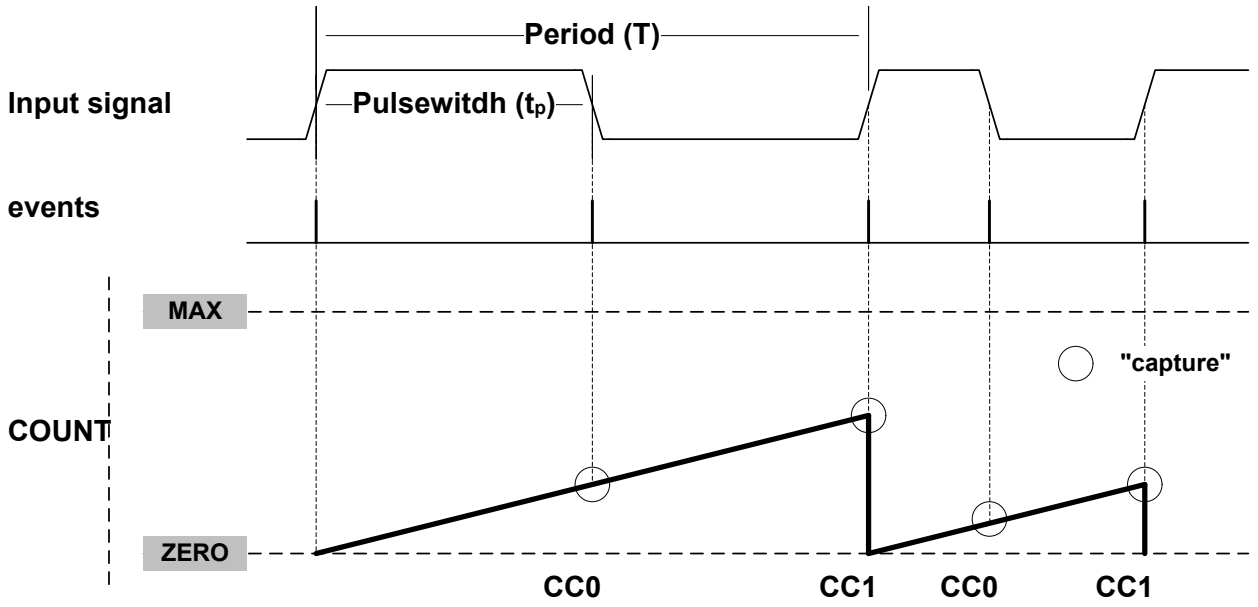
40.6.2.8.2 Period and Pulse-Width (PPW/PWP) Capture Action on Events

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

Figure 40-13. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and another one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT = PPW (period and pulse-width) offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound must occur on the rising edge or the falling edge. If EVCTRL.TCINV = 1, the wraparound will happen on the falling edge. In case pin capture is enabled, this can also be achieved by modifying the value of the DRVCTRL.INVENx bit.

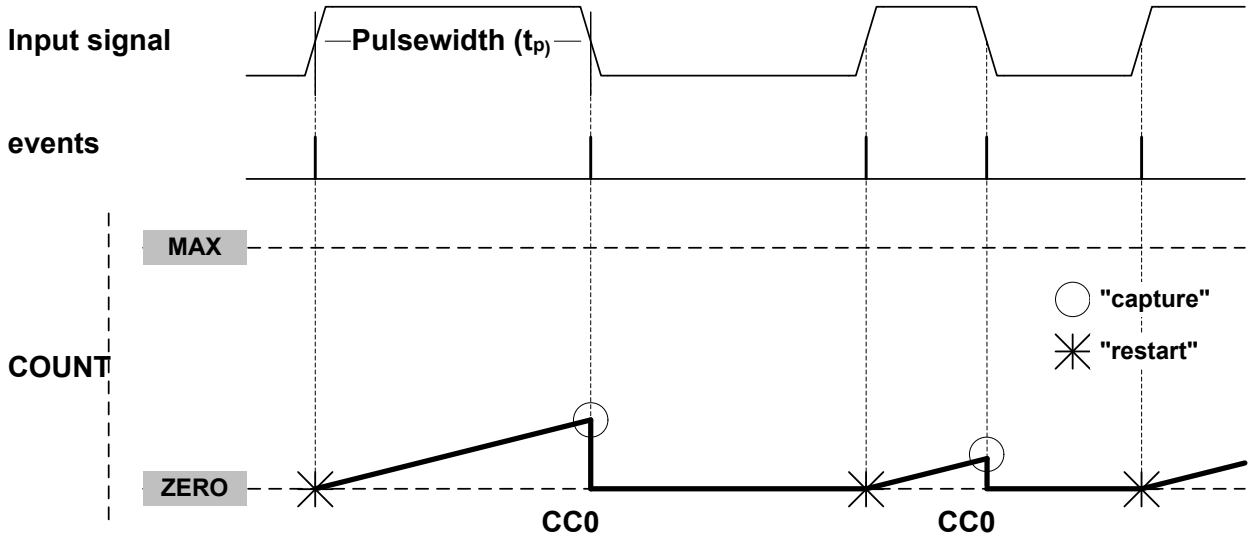
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx = 1). Consequently, both channels must be enabled to fully characterize the input.

40.6.2.8.3 Pulse-Width (PW) Capture Action on Events

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to EVCTRL.TCEINV).

Figure 40-14. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

40.6.3 Additional Features

40.6.3.1 One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to '0'.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC counts until an overflow or underflow occurs and stops the counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

40.6.3.2 Time-Stamp Capture on Events or I/Os

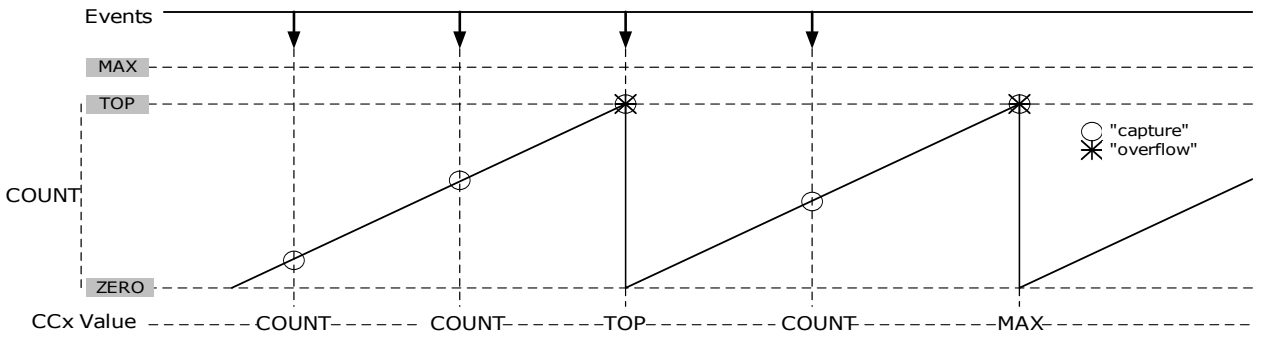
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event from the Event System or the I/O pin is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 40-15. Time Stamp



40.6.3.3 Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

CCx Content:

In CAPTMIN operations, CCx keeps the minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no capture is performed using the corresponding channel.

MCx Behaviour:

In the CAPTMIN operation, the capture is performed only if, when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only if, when on capture event time, the counter value is higher or equal to the value captured on the previous event. So the interrupt flag is set when a new absolute local minimum value is detected.

40.6.3.4 Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

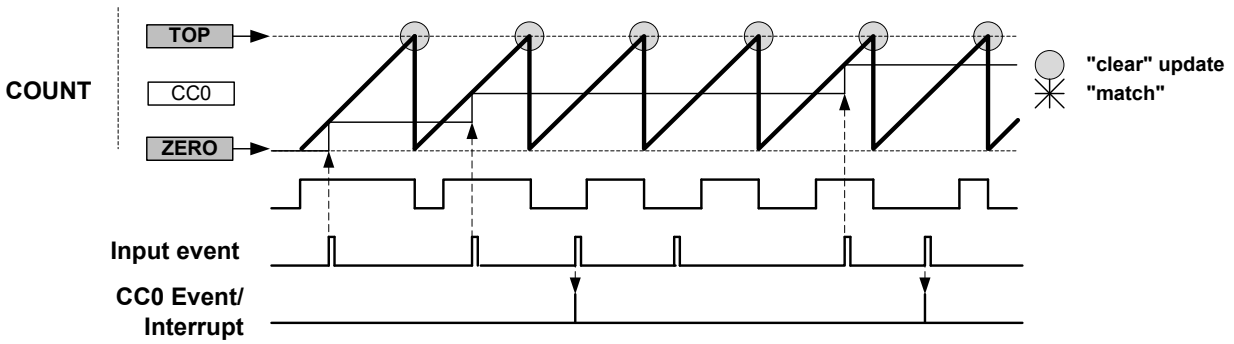
CCx Content:

In CAPTMAX operations, CCx keeps the maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no capture is performed using the corresponding channel.

MCx Behaviour:

In the CAPTMAX operation, the capture is performed only if, when on capture event time, the counter value is higher than the last captured value. The MCx Interrupt flag is set only if, when on capture event time, the counter value is lower or equal to the value captured on the previous event. So the Interrupt flag is set when a new absolute local maximum value is detected.

Figure 40-16. Maximum Capture Operation with CC0 Initialized with ZERO Value



40.6.4 DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

40.6.5 Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) 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 register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). 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 either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See *INTFLAG* from Related Links for more details on how to clear the interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

40.8.7. INTFLAG

40.6.6 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCX0-1)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)
- Capture time stamp (STAMP)

- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events (EVU0-2) to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. See *Event System (EVSYS)* from Related Links for additional information on configuring the asynchronous events.

Related Links

30. Event System (EVSYS)

40.6.7 Sleep Mode Operation

The TC can be configured to operate in any sleep mode (Idle, Standby Sleep). To be able to run in Standby Sleep mode, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in the STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

40.6.8 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 and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): Synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

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.

40.7 Register Summary - 8-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8	DMAOS				ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11 ... 0x13	Reserved									
0x14	COUNT	7:0	COUNT[7:0]							
0x15 ... 0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E ... 0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							

40.8 Register Description - 8-Bit Mode

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 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.

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
- HS = Hardware Set
- HC = Hardware Cleared

40.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
Access				R/W	R/W		R/W	R/W
Reset				0	0		0	0

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
	DMAOS				ALOCK	PRESCALER[2:0]		
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 28:27 – CAPTMODE1[1:0] Capture mode Channel 1
These bits select the channel 1 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 25:24 – CAPTMODE0[1:0] Capture mode Channel 0
These bits select the channel 0 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 20, 21 – COPENx Capture On Pin x Enable [x=1..0]
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input. This bit is not synchronized.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable [x=1..0]
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel. These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 15 – DMAOS DMA One-Shot Trigger Mode

This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit generates a DMA trigger on TC cycle following a TC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit generates DMA triggers on each TC cycle.
This bit is not synchronized.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In Standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC continues to request the clock when its operation is stopped (STATUS.STOP = 1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby

This bit is used to keep the TC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization

These bits select whether the counter must wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock

Value	Name	Description
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	—	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the Counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	—	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
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 resets all registers in the TC, except DBGCTRL, to their initial state and the TC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence; all other writes in the same write-operation are discarded.
This bit is not enable-protected.

40.8.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as zero.
Writing '0x0' to these bits has no effect.
Writing a '1' to any of these bits clears the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with the value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when the input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.8.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as '0'.
Writing '0x0' to these bits has no effect.
Writing a value different from '0x0' to these bits issues a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables one-shot operation.

Value	Description
0	The TC wraps around and continue counting on an overflow/underflow condition.
1	The TC wraps around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with a value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.8.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

40.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
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 register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x [x = 1..0]
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is no place to store the new capture.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

40.8.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid [x=1..0]
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set or automatically cleared by hardware on the UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – CLIENT Client Status Flag
This bit is only available in 32-bit mode on the Client TC (in other words, TC1, TC3, TC5 and/or TC7). The bit is set when the associated Host TC (TC0, TC2, TC4 and/or TC6, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

40.8.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation must be used. The waveform generation operations are explained in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode, it is the respective MAX value.

Related Links
40.6.2.6.1. Waveform Output Operations

40.8.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable [x=1..0]
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

40.8.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and must not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

40.8.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy [x=0..1]
For details on the CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written and cleared on the update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written and cleared on the update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of the ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the ENABLE bit between the clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of the SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the SWRST bit between the clock domains is started.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

40.8.13 Counter Value, 8-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

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 7:0 – COUNT[7:0] Counter Value
These bits contain the current counter value.

40.8.14 Period Value, 8-bit Mode

Name: PER
Offset: 0x1B
Reset: 0xFF
Property: Write-Synchronized

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	1

Bits 7:0 – PER[7:0] Period Value
These bits hold the value of the TC period count.

40.8.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CC[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 7:0 – CC[7:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

40.8.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

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

Bits 7:0 – PERBUF[7:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to the PER register on the UPDATE condition.

40.8.17 Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CCBUF[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 7:0 – CCBUF[7:0] Channel x Compare Buffer Value [x=1..0]
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD = 1), the data from buffer registers is copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD = 0x3), including the software update command.

40.9 Register Summary - 16-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8	DMAOS				ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
0x04	CTRLBCLR	7:0		CMD[2:0]				ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0		CMD[2:0]				ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11 ... 0x13	Reserved									
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x16 ... 0x19	Reserved									
0x1A	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x1E	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x20 ... 0x2D	Reserved									
0x2E	PERBUF	7:0	PERBUF[7:0]							
		15:8	PERBUF[15:8]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
0x32	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							

40.10 Register Description - 16-Bit Mode

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 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.

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
- HS = Hardware Set
- HC = Hardware Cleared

40.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
Access				R/W	R/W		R/W	R/W
Reset				0	0		0	0

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
	DMAOS				ALOCK	PRESCALER[2:0]		
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 28:27 – CAPTMODE1[1:0] Capture mode Channel 1
These bits select the channel 1 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 25:24 – CAPTMODE0[1:0] Capture mode Channel 0
These bits select the channel 0 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 20, 21 – COPENx Capture On Pin x Enable [x=1..0]
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input. This bit is not synchronized.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable [x=1..0]
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel. These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 15 – DMAOS DMA One-Shot Trigger Mode

This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit generates a DMA trigger on TC cycle following a TC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit generates DMA triggers on each TC cycle.
This bit is not synchronized.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In Standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC continues to request the clock when its operation is stopped (STATUS.STOP = 1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby

This bit is used to keep the TC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization

These bits select whether the counter must wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock

Value	Name	Description
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	—	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the Counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	—	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
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 resets all registers in the TC, except DBGCTRL, to their initial state and the TC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence; all other writes in the same write-operation are discarded.
This bit is not enable-protected.

40.10.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as zero.
Writing '0x0' to these bits has no effect.
Writing a '1' to any of these bits clears the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with the value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when the input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.10.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as '0'.
Writing '0x0' to these bits has no effect.
Writing a value different from '0x0' to these bits issues a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables one-shot operation.

Value	Description
0	The TC wraps around and continue counting on an overflow/underflow condition.
1	The TC wraps around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with a value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.10.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

40.10.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
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 register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.10.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.10.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x [x = 1..0]
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is no place to store the new capture.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

40.10.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid [x=1..0]
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set or automatically cleared by hardware on the UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – CLIENT Client Status Flag
This bit is only available in 32-bit mode on the Client TC (in other words, TC1, TC3, TC5 and/or TC7). The bit is set when the associated Host TC (TC0, TC2, TC4 and/or TC6, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

40.10.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation must be used. The waveform generation operations are explained in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode, it is the respective MAX value.

Related Links
40.6.2.6.1. Waveform Output Operations

40.10.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable [x=1..0]
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

40.10.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and must not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

40.10.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy [x=0..1]
For details on the CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written and cleared on the update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written and cleared on the update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of the ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the ENABLE bit between the clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of the SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the SWRST bit between the clock domains is started.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

40.10.13 Counter Value, 16-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

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 contain the current counter value.

40.10.14 Period Value, 16-bit Mode

Name: PER
Offset: 0x1A
Reset: 0xFFFF
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	1

Bits 15:0 – PER[15:0] Period Value
These bits hold the value of the TC period count.

40.10.15 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[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
	CC[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 – CC[15:0] Channel x Compare/Capture Value [x=1..0]
These bits contain the compare/capture value in 16-bit TC mode. In Match Frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

40.10.16 Period Buffer Value, 16-bit Mode

Name: PERBUF
Offset: 0x2E
Reset: 0xFFFF
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERBUF[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
	PERBUF[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	1

Bits 15:0 – PERBUF[15:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

40.10.17 Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CCBUF[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
	CCBUF[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 – CCBUF[15:0] Channel x Compare Buffer Value [x=1..0]
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD = 1), the data from buffer registers is copied into the corresponding CCx register under the UPDATE condition (CTRLBSET.CMD = 0x3) including the software update command.

40.11 Register Summary - 32-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8	DMAOS				ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
0x04	CTRLBCLR	7:0		CMD[2:0]				ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0		CMD[2:0]				ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11 ... 0x13	Reserved									
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x18 ... 0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x20	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x24 ... 0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							
0x34	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							

40.12 Register Description - 32-Bit Mode

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 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.

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
- HS = Hardware Set
- HC = Hardware Cleared

40.12.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
Access				R/W	R/W		R/W	R/W
Reset				0	0		0	0

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
	DMAOS				ALOCK	PRESCALER[2:0]		
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 28:27 – CAPTMODE1[1:0] Capture mode Channel 1
These bits select the channel 1 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 25:24 – CAPTMODE0[1:0] Capture mode Channel 0
These bits select the channel 0 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 20, 21 – COPENx Capture On Pin x Enable [x=1..0]
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input. This bit is not synchronized.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable [x=1..0]
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel. These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 15 – DMAOS DMA One-Shot Trigger Mode

This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit generates a DMA trigger on TC cycle following a TC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit generates DMA triggers on each TC cycle.
This bit is not synchronized.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In Standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC continues to request the clock when its operation is stopped (STATUS.STOP = 1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby

This bit is used to keep the TC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization

These bits select whether the counter must wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock

Value	Name	Description
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	—	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the Counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	—	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
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 resets all registers in the TC, except DBGCTRL, to their initial state and the TC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence; all other writes in the same write-operation are discarded.
This bit is not enable-protected.

40.12.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as zero.
Writing '0x0' to these bits has no effect.
Writing a '1' to any of these bits clears the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with the value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when the input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.12.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command is executed, the CMD bit group is read back as '0'.
Writing '0x0' to these bits has no effect.
Writing a value different from '0x0' to these bits issues a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables one-shot operation.

Value	Description
0	The TC wraps around and continue counting on an overflow/underflow condition.
1	The TC wraps around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with a value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.12.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

40.12.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
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 register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.12.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
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 Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.12.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x [x = 1..0]
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is no place to store the new capture.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

40.12.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid [x=1..0]
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set or automatically cleared by hardware on the UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – CLIENT Client Status Flag
This bit is only available in 32-bit mode on the Client TC (in other words, TC1, TC3, TC5 and/or TC7). The bit is set when the associated Host TC (TC0, TC2, TC4 and/or TC6, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

40.12.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation must be used. The waveform generation operations are explained in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode, it is the respective MAX value.

Related Links
40.6.2.6.1. Waveform Output Operations

40.12.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable [x=1..0]
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

40.12.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and must not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

40.12.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy [x=0..1]
For details on the CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written and cleared on the update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written and cleared on the update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of the ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the ENABLE bit between the clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of the SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the SWRST bit between the clock domains is started.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

40.12.13 Counter Value, 32-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

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 contain the current counter value.

40.12.14 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[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
	CC[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
	CC[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
	CC[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 – CC[31:0] Channel x Compare/Capture Value [x=1..0]
These bits contain the compare/capture value in 32-bit TC mode. In Match Frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

40.12.15 Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF[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
	CCBUF[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
	CCBUF[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
	CCBUF[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 – CCBUF[31:0] Channel x Compare Buffer Value [x=1..0]
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD = 1), the data from buffer registers is copied into the corresponding CCx register under the UPDATE condition (CTRLBSET.CMD = 0x3), including the software update command.

41. Timer/Counter for Control Applications (TCC)

41.1 Overview

The device provides three instances of the Timer/Counter for Control Applications (TCC).

Each TCC instance consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation, such as frequency generation and pulse-width modulation.

Waveform extensions are featured for motor control, ballast, LED, H-bridge, power converters and other types of power control applications. They allow for low-side and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, disabling and/or shut-down of external drivers.

Note: The TCC configurations, such as channel numbers and features, can be reduced for some of the TCC instances.

Table 41-1. TCC Specific Configuration

TCC No.	Counter Size (SIZE)	Link Role (Host_Client_MODE) (0 = NA, 1 = Host, 2 = Client)	Channels (CC_NUM)	Pins WO_NUM (OW_NUM)
0	24	1	6	6
1	24	2	6	6
2	16	0	2	2

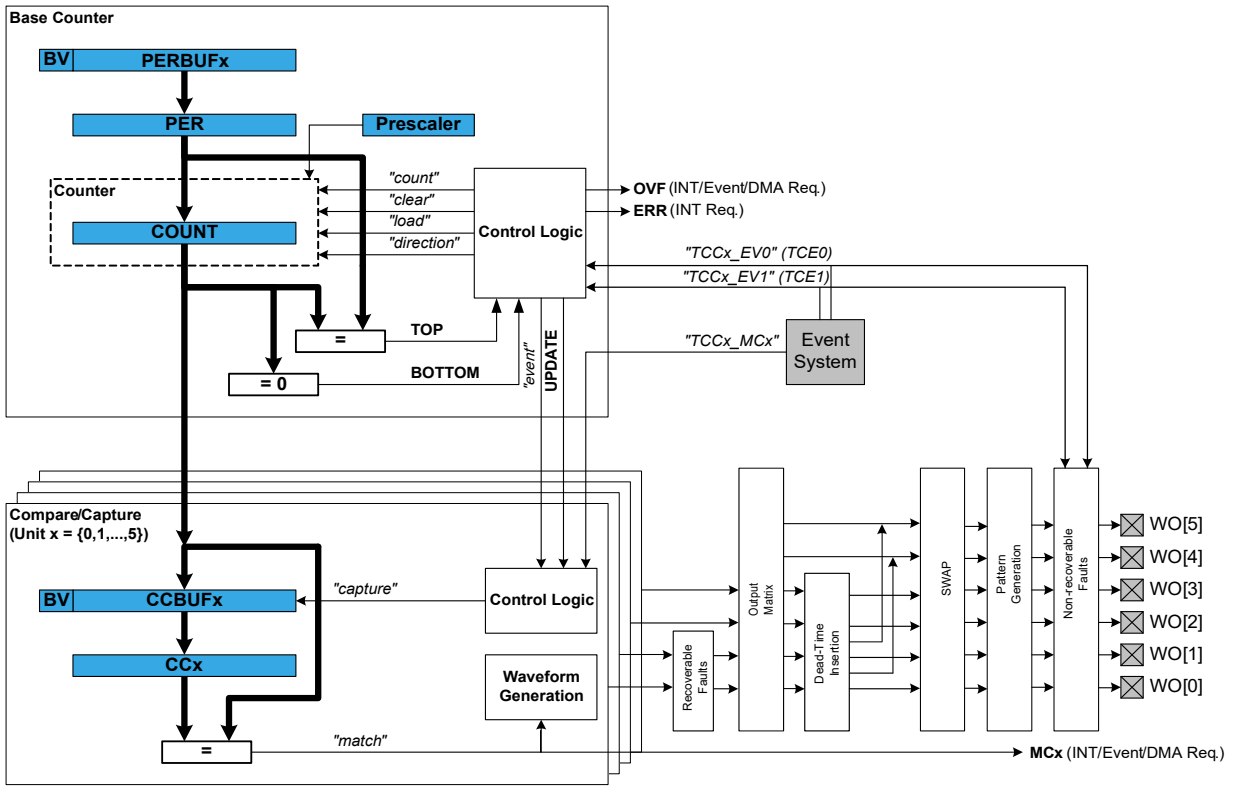
41.2 Features

- Up to Six Compare/Capture Channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Waveform Generation:
  - Frequency generation
  - Single-slope pulse-width modulation (PWM)
  - Dual-slope PWM with half-cycle reload capability
- Input Capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform Extensions:
  - Configurable distribution of compare channels outputs across port pins
  - Low-side and high-side output with programmable dead-time insertion
  - Waveform swap option with double buffer support
  - Pattern generation with double buffer support
  - Dithering support
- Fault Protection for Safe Disabling of Drivers:
  - Two recoverable fault sources

  - Two non-recoverable fault sources
  - Debugger can be a source of non-recoverable fault
- Input Events:
  - Two input events (EVx) for counter
  - One input event (MCx) for each channel
- Output Events:
  - Three output events (Count, re-trigger and overflow) are available for counter
  - One compare match/input capture event output for each channel
- Interrupts:
  - Overflow and re-trigger interrupt
  - Compare match/input capture interrupt
  - Interrupt on fault detection

41.3 Block Diagram

Figure 41-1. Timer/Counter for Control Applications - Block Diagram



41.4 Signal Description

Table 41-2. Signal Description

Pin Name	Type	Description
TCCx/WO[0]	Digital output	Compare channel 0 waveform output
TCCx/WO[1]	Digital output	Compare channel 1 waveform output
...	...	...
TCCx/WO[WO_NUM-1]	Digital output	Compare channel n waveform output

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

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

41.5 Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

41.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Peripheral Pin Select (PPS).

41.5.2 Power Management

This peripheral can continue to operate in any sleep mode (Idle, Standby Sleep) where its source clock is running. The interrupts can wake up the device from the sleep modes. Events connected to the event system can trigger other operations in the system without exiting the sleep modes.

41.5.3 Clocks

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the generic clock controller before using the TCC.
Note: TCC1 and TCC2 share a single peripheral clock generator.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (PB1_CLK). Due to this asynchronicity, writing certain registers requires synchronization between the clock domains.

41.5.4 DMA

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

Related Links

26. Direct Memory Access Controller (DMAC)

41.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

41.5.6 Events

The events of this peripheral are connected to the Event System.

41.5.7 Debug Operation

When the CPU is halted in Debug mode, this peripheral halts normal operation. This peripheral can be forced to continue operation during debugging. See the *DBGCTRL* register from Related Links.

Related Links

41.8.8. DBGCTRL

41.5.8 Register Access Protection

Registers with write access can be optionally write protected by the PAC, except for the following:

- Interrupt Flag register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture and Compare/Capture Buffer registers (CCx, CCBUFx)
- Control Waveform register (WAVE)
- Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTBUF)

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.

41.5.9 Analog Connections

Not applicable.

41.6 Functional Description

41.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 41-3. Timer/Counter for Control Applications – Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the Waveform Generator mode in Waveform Output Operations. See *Waveform Output Generation Operations* from Related Links.
ZERO	The counter reaches ZERO when it contains all zeros.
MAX	The counter reaches maximum when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g., counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

Each TCC instance has up to six compare/capture channels (CCx).

The Counter register (COUNT), Period registers with Buffer (PER and PERBUF), and Compare and Capture registers with buffers (CCx and CCBUFx) are 16- or 24-bit registers, depending on each TCC instance. Each Buffer register has a Buffer Valid (BUFV) flag that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate interrupt requests or generate events for the Event System. In Waveform Generator mode, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event-controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The MCE0 and MCE1 asynchronous event sources are shared with the recoverable fault unit. Only asynchronous events are used internally when fault unit extension is enabled. See *Event System (EVSYS)* from Related Links for further details on how to configure asynchronous events routing.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking and qualification options. See *Recoverable Faults* from Related Links.

In order to support applications with different types of motor control, ballast, LED, H-bridge, power converter and other types of power switching applications, the following independent units are implemented in some of the TCC instances as optional and successive units:

- Recoverable faults and non-recoverable faults
- Output matrix
- Dead-time insertion
- Swap
- Pattern generation

See *Timer/Counter for Control Applications - Block Diagram* in the *Block Diagram* from Related Links.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted Low Side (LS) and inverted High Side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event-controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a preconfigured value that is safe for the application. This is typically used for instant and predictable shut-down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. See *Event System (EVSYS)* from Related Links for further details on how to configure asynchronous events routing.

Related Links

30. Event System (EVSYS)
41.3. Block Diagram
41.6.3.5. Recoverable Faults
41.6.2.5.1. Waveform Output Generation Operations

41.6.2 Basic Operation

41.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE=0):

- Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)

- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Enable-protected bits in the 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.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Enable the TCC bus clock if not already enabled by default (PB1_CLK).
2. If Capture mode is required, enable the channel in Capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESCSYNC).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR) to '1'.
4. Select the Waveform Generation operation in the WAVE register (WAVE.WAVEGEN).
5. Select the Waveform Output Polarity in the WAVE register (WAVE.POL).
6. The waveform output can be inverted for the individual channels using the Waveform Output Invert Enable bit group in the Driver register (DRVCTRL.INVEN).

41.6.2.2 Enabling, Disabling and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a '0' to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, are reset to their initial state and the TCC is disabled. See the *CTRLA* register from Related Links.

The TCC must be disabled before the TCC is reset to avoid undefined behavior.

Related Links

41.8.1. CTRLA

41.6.2.3 Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

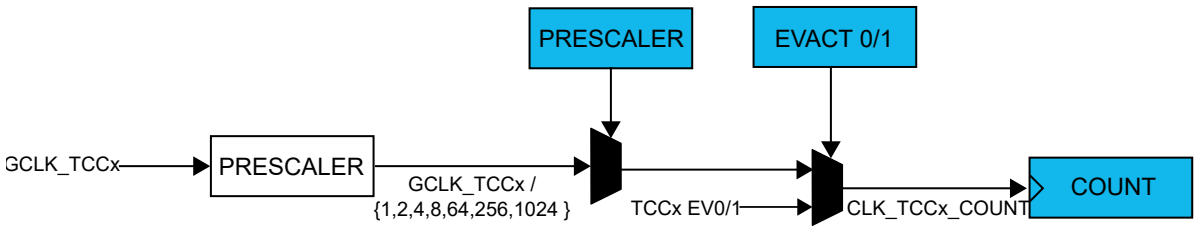
If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCCx clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCCx_COUNT.

Figure 41-2. Prescaler



41.6.2.4 Counter Operation

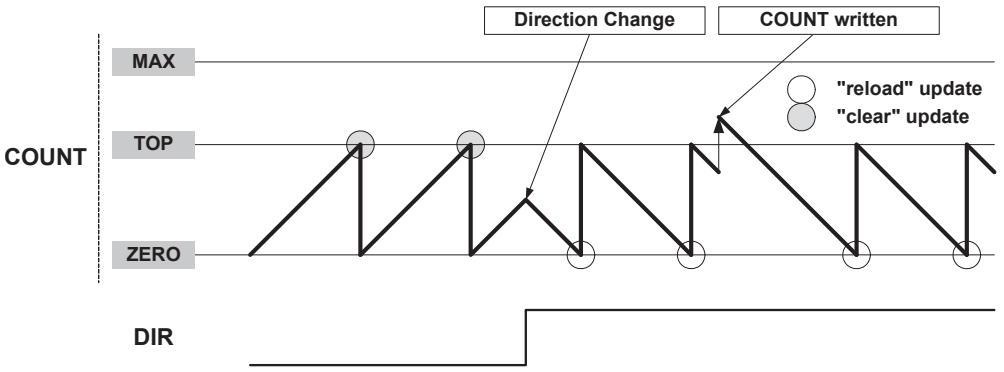
Depending on the mode of operation, the counter is cleared, reloaded, incremented or decremented at each TCC clock input (CLK_TCCx_COUNT). A counter clear or reload mark the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it is counting up, and, if the bit is one, it is counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter is set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) is set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow) and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt or an event. An overflow/underflow occurrence (in other words, a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT). The One-Shot feature is explained in the Additional Features section.

Figure 41-3. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. The COUNT value will always be ZERO or TOP, depending on the direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value was written to it or the TCC was stopped at a value other than ZERO. The write access has higher priority than count, clear or reload. The direction of the counter can also be changed during normal operation. See Figure 41-3 for more information.

41.6.2.4.1 Stop Command

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will get set to bottom/top depending on the direction: up/down. All waveforms are cleared, and the Stop bit in the Status register is set (STATUS.STOP).

41.6.2.4.2 Pause Event Action

A pause command can be issued when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT1 = 0x3, STOP).

When a pause is detected, the counter can stop immediately maintaining its current value and all waveforms keep their current state until a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0 = 0x3, START).

41.6.2.4.3 Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER) or from an event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACTn = 0x1, RETRIGGER).

When the command is detected during the counting operation, the counter will be reloaded or cleared depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACTn = 0x1, RETRIGGER), enabling the counter will not start the counter. The counter starts on the next incoming event and restarts on the corresponding following event.

41.6.2.4.4 Start Event Action

The start action can be selected in the Event Control register (EVCTRL.EVACT0 = 0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note: When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 = 0x3, START), enabling the counter will not start the counter. The counter starts on the next incoming event, but it will not restart on subsequent events.

41.6.2.4.5 Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value depending on the direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0 = 0x5, COUNT).

41.6.2.4.6 Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1 = 0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

41.6.2.4.7 Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0 = 0x4, INC) and can change the Counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments regardless of the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR).

41.6.2.4.8 Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1 = 0x4, DEC) and can change the Counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements regardless of the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR).

41.6.2.4.9 Non-Recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn = 0x7, FAULT). When received, the counter is stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

41.6.2.4.10 Event Action Off

If the event action is disabled (EVCTRL.EVACTn = 0x0, OFF), enabling the counter will also start the counter.

41.6.2.5 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD = 0x3, UPDATE). See *Double Buffering* from Related Links. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Related Links

41.6.2.6. Double Buffering

41.6.2.5.1 Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally, invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.
   Note: Event must not be used when the compare channel is set in waveform output operating mode.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e., INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)

- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other Waveforms Generation modes, the update time occurs on counter wraparound, on overflow, underflow or re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 41-4. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			—	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			—	ZERO
DSBOTH	Dual-slope PWM	PER	TOP(1) & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	—

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

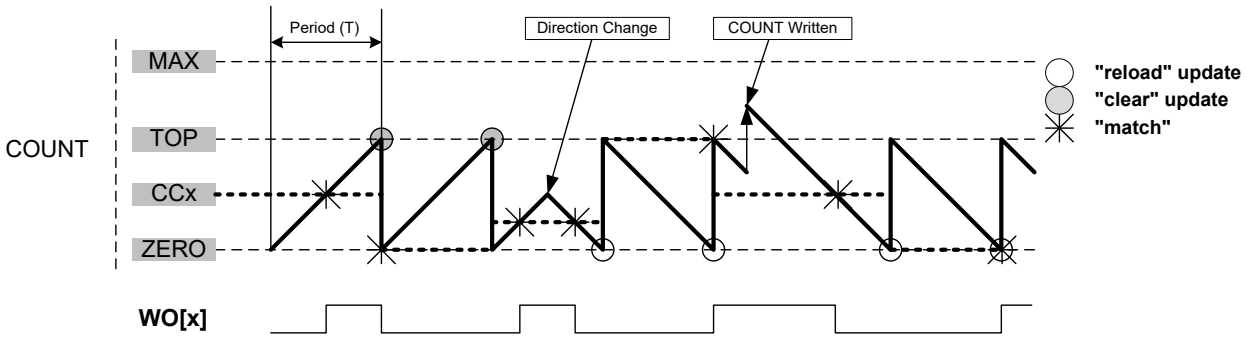
Related Links

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

41.6.2.5.2 Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.

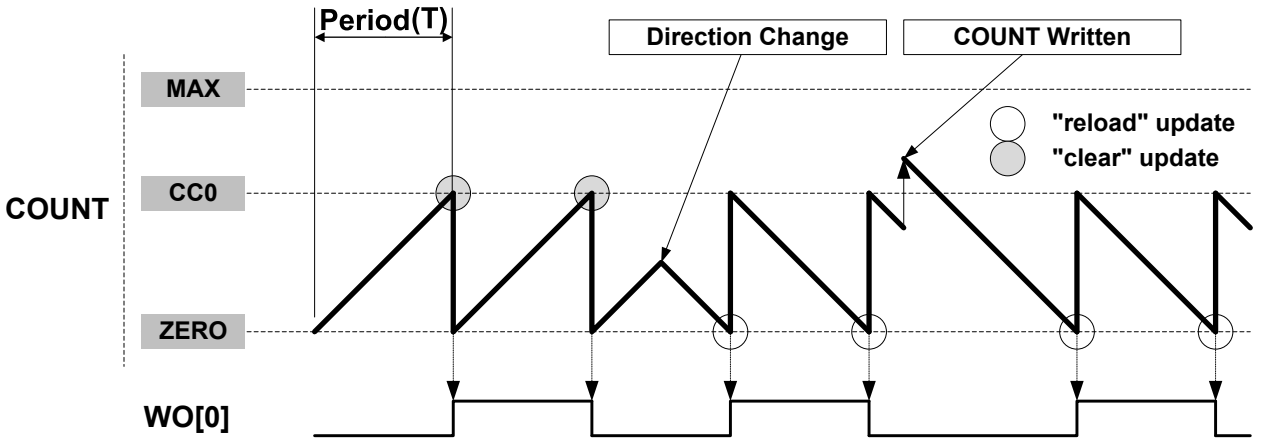
Figure 41-4. Normal Frequency Operation



41.6.2.5.3 Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 41-5. Match Frequency Operation



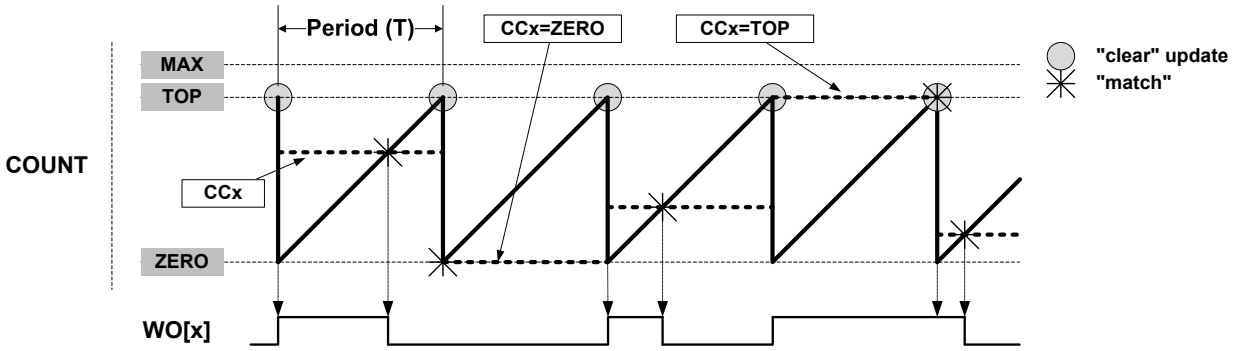
41.6.2.5.4 Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

41.6.2.5.5 Single-Slope PWM Operation

For single-slope PWM generation, the period time (T) is controlled by Top value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

Figure 41-6. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency (f_{GCLK_TCCx}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TCCx}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

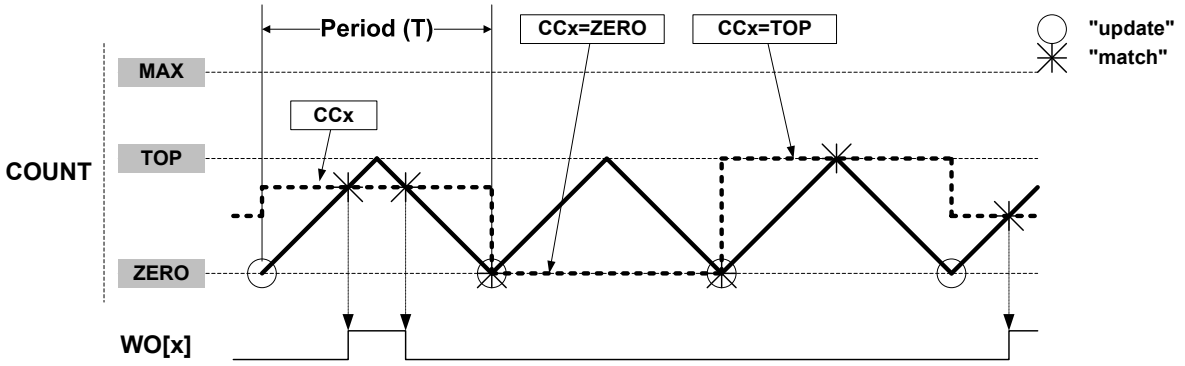
41.6.2.5.6 Dual-Slope PWM Generation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCx control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set

on compare match when up-counting, and cleared on compare match when down-counting. An interrupt and/or event is generated on TOP (when counting upwards) and/or ZERO (when counting up or down).

In DSBOTH operation, the circular buffer must be enabled to enable the update condition on TOP.

Figure 41-7. Dual-Slope Pulse Width Modulation



Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM (R_{PWM_DS}):

$$R_{PWM\_DS} = \frac{\log(PER+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCCx}, and can be calculated by the following equation:

$$f_{PWM\_DS} = \frac{f_{GCLK\_TCCx}}{2N \cdot PER}$$

N represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCCx}) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width (P_{PWM_DS}) depends on the compare channel (CCx) register value and the peripheral clock frequency (f_{GCLK_TCCx}), and can be calculated by the following equation:

$$P_{PWM\_DS} = \frac{2N \cdot (TOP - CCx)}{f_{GCLK\_TCCx}}$$

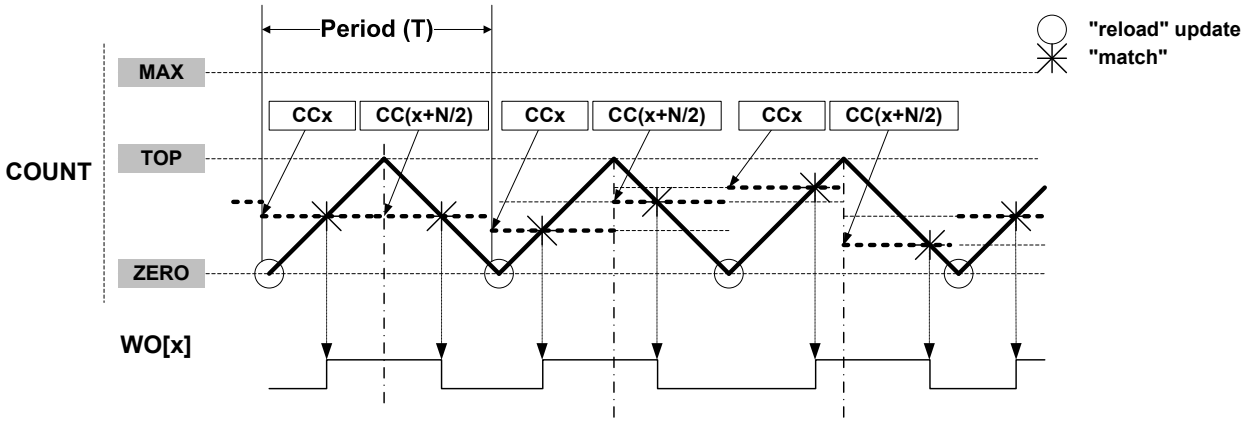
N represents the prescaler divider used.

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the CCx MSB bit defines the ramp on which the CCx Match interrupt or event is generated. (Rising if CCx[MSB] = 0, falling if CCx[MSB] = 1.)

41.6.2.5.7 Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time is controlled by PER while CCx control the generated waveform output edge during up-counting and CC(x+CC_NUM/2) control the generated waveform output edge during down-counting.

Figure 41-8. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



41.6.2.5.8 Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 41-5. Waveform Generation Set/Clear Conditions

Waveform Generation Operation	DIR	POLx	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CCx
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

41.6.2.6 Double Buffering

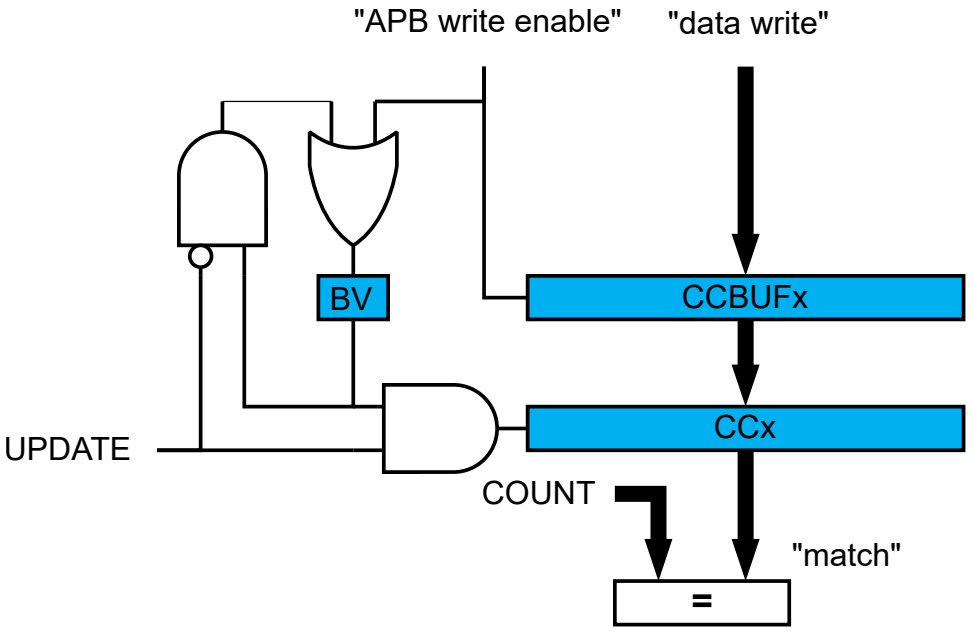
The Pattern (PATT), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV, PERBUFV and CCBUFVx) bit in the STATUS register that indicates that the Buffer register contains a valid value that can be copied into the corresponding register.

When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers is copied into the corresponding register under hardware UPDATE conditions, then, the Buffer Valid flags bit in the STATUS register are automatically cleared by hardware.

Note: The software update command (CTRLBSET.CMD = 0x3) acts independently of the LUPD value.

A compare register is double buffered as illustrated in the following figure.

Figure 41-9. Compare Channel Double Buffering



Both the registers (PATT/PER/CCx) and corresponding Buffer registers (PATTBUFPERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD.

Note: In NFRQ, MFRQ or PWM Down-Counting Counter mode (CTRLBSET.DIR = 1), when double buffering is enabled (CTRLBCLR.LUPD = 1), PERBUF register is continuously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new Top value to the Period register (PER or CC0, depending on the Waveform Generation mode). Any period update on registers (PER or CCx) is effective after the synchronization delay, regardless of double buffering enabling.

Figure 41-10. Unbuffered Single-Slope Up-Counting Operation

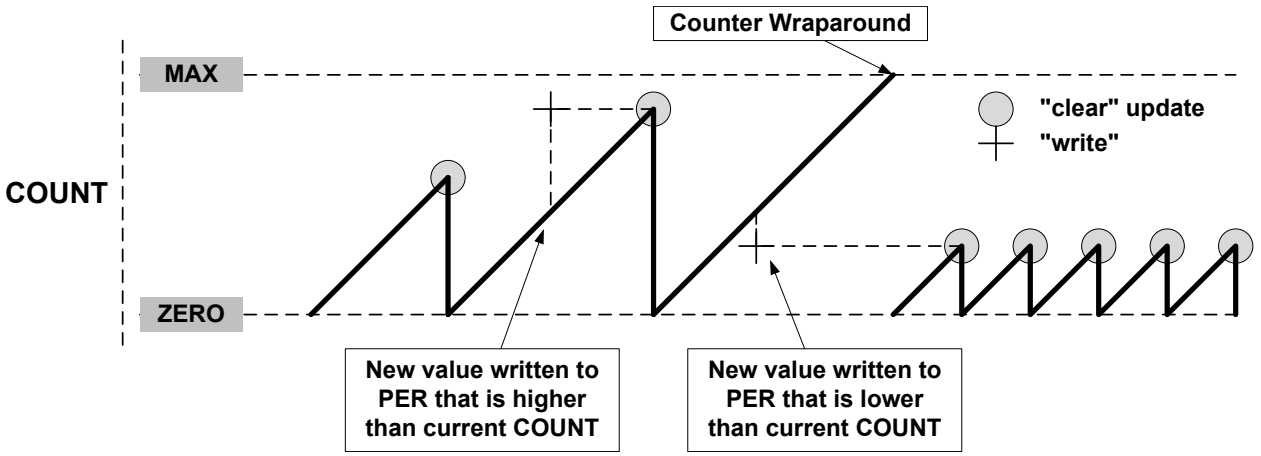
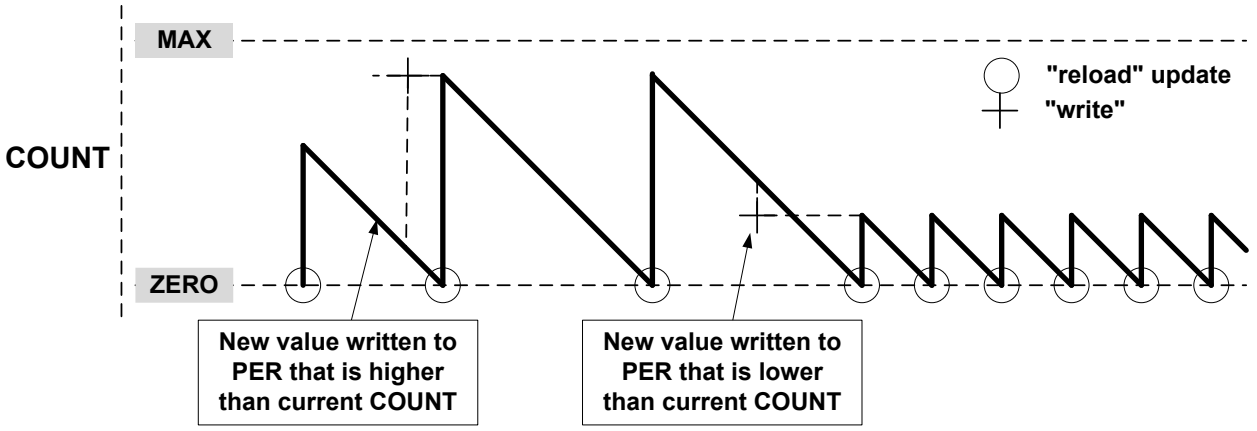
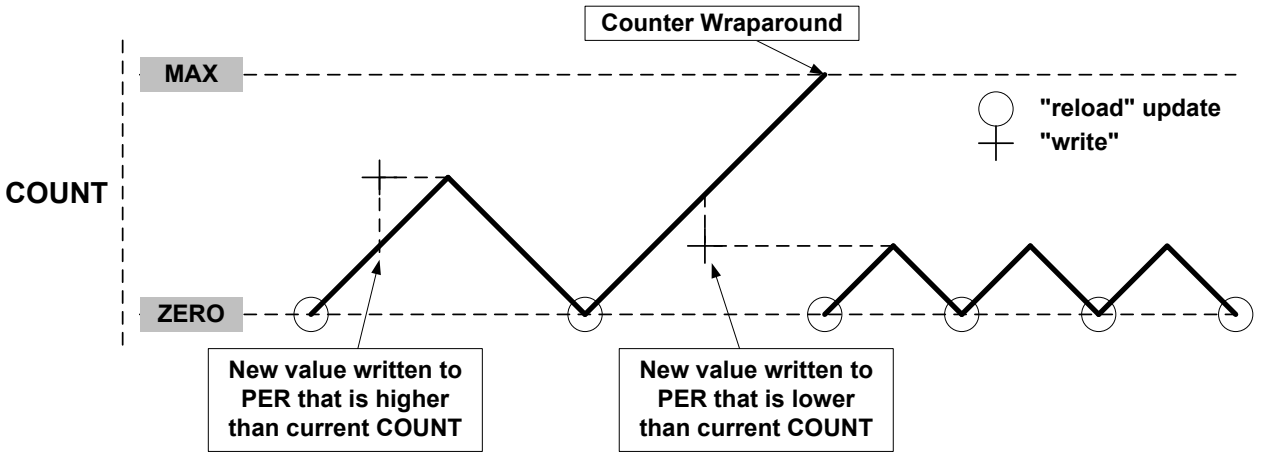


Figure 41-11. Unbuffered Single-Slope Down-Counting Operation



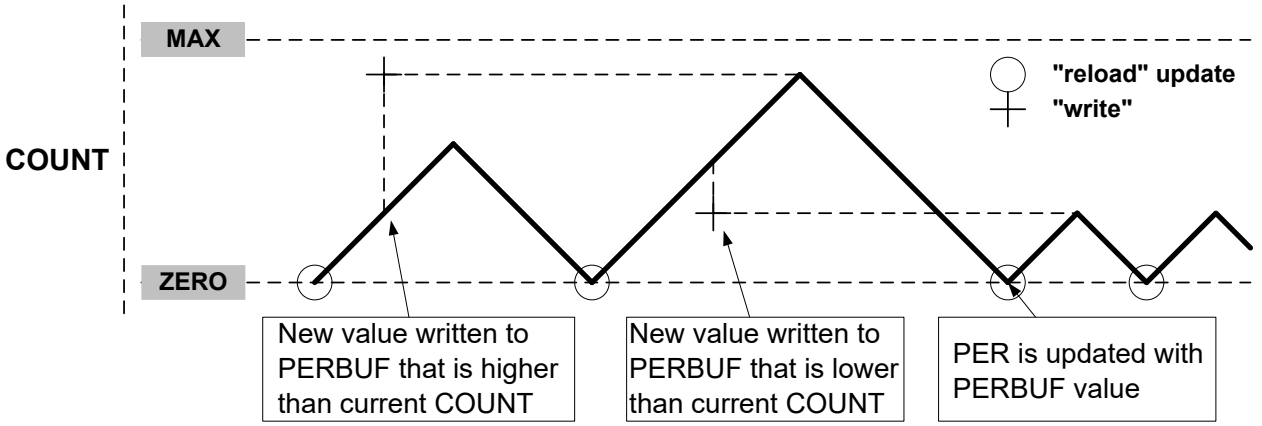
A counter wraparound can occur in any operation mode when up-counting without buffering; see Figure 41-10. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT wraps before a compare match.

Figure 41-12. Unbuffered Dual-Slope Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as illustrated in the following figure. This prevents wraparound and the generation of odd waveforms.

Figure 41-13. Changing the Period Using Buffering



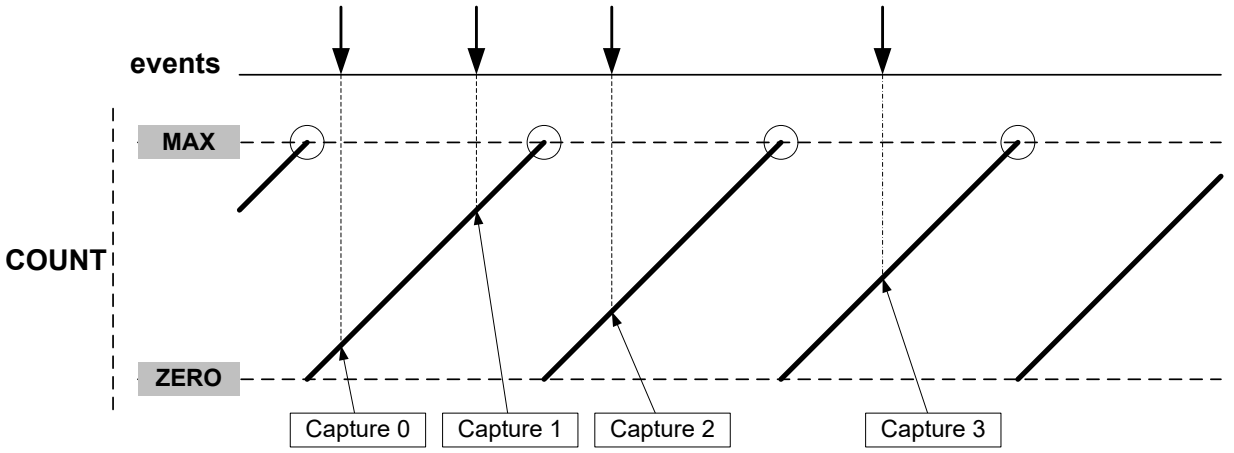
41.6.2.7 Capture Operations

To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

Event Capture Action

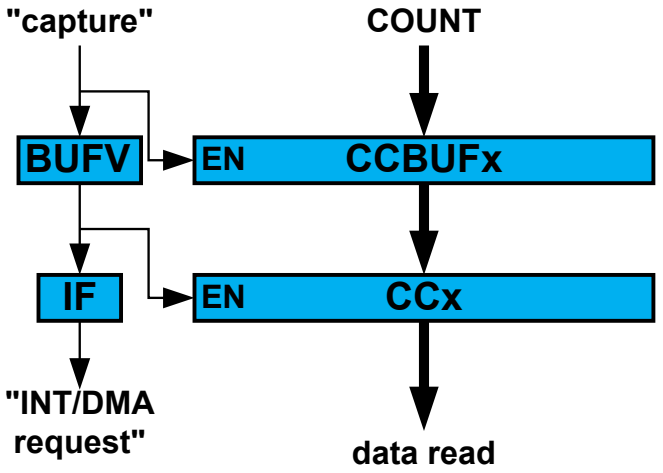
The compare/capture channels can be used as input capture channels to capture events from the Event System and give them a timestamp. The following figure illustrates four capture events for one capture channel. Event system channels must be configured to operate in asynchronous mode when used for capture operations.

Figure 41-14. Input Capture Timing



For input capture, the Buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The Buffer Valid flag is passed to set the CCx Interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value cannot be read; all captured data must be read from the CCx register.

Figure 41-15. Capture Double Buffering



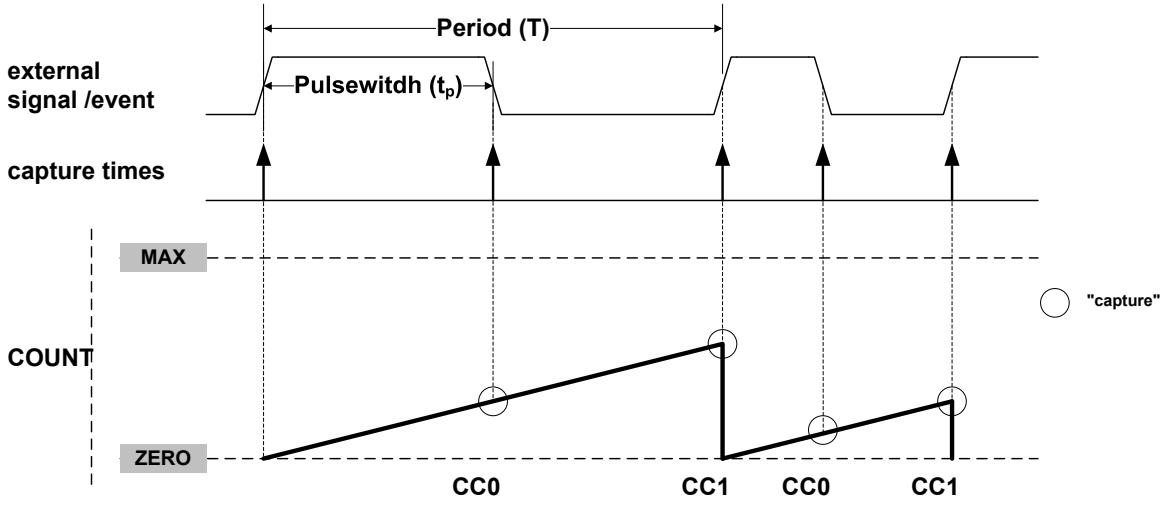
The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBUFV) is still set, the new timestamp will not be stored and INTFLAG.ERR is set.

Period and Pulse-Width (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and *dutyCycle* of an input signal, illustrated as follows.

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 41-16. PWP Capture



Selecting PWP or PPW in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. When using the PPW event action, period *T* is captured into CC0 and the pulse-width t_p into CC1. The PWP (Pulse-width and Period) event action offers the same functionality, but *T* is captured into CC1 and t_p into CC0.

The Timer/Counter Event x Invert Enable bit in the Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound must occur on the rising edge or the falling edge. If EVCTRL.TCEINVx = 1, the wraparound will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in Capture mode (CTRLA.CPTENx = 1). If not, the capture action is ignored and the channel is enabled in compare mode of the operation. When only one of these channels is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the INTFLAG.MCx is still set, the new timestamp will not be stored and INTFLAG.ERR is set.

Notes:

1. When up-counting (CTRLBSET.DIR = 0), counter values lower than '1' cannot be captured in Capture Minimum mode (FCTRLn.CAPTURE = CAPTMIN). To capture the full range including value '0', the TCC must be configured in Down-counting mode (CTRLBSET.DIR = 0).
2. In dual-slope PWM operation and when TOP is lower than MAX/2, the CCx MSB captures the CTRLB.DIR state to identify the ramp where the capture was done. For rising ramps CCx[MSB] is '0'; for falling ramps CCx[MSB] = 1.

41.6.3 Additional Features

41.6.3.1 One-Shot Operation

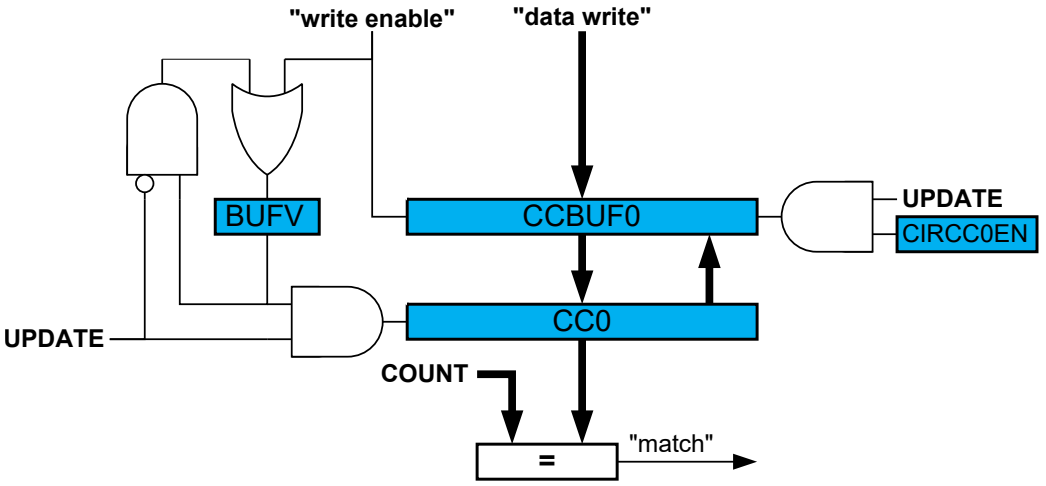
When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC counts until an overflow or underflow occurs and stops counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

41.6.3.2 Circular Buffer

The Period register (PER) and the Compare Channels register (CC0 toCC5) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 41-17. Circular Buffer on Channel 0



41.6.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
  value = cycle * dithercy;
  if (((MASK & value) + dithercy) > MASK)
    return 1;
 return 0;
}
```

Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{16} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

Note: If DITH4 mode is enabled, the last 4 significant bits from PER/CCx or COUNT register correspond to the DITHERCY value, rest of the bits corresponds to PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{32} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{64} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

Dithering on Pulse-Width

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{16} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{32} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{64} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}x}}\right)$$

Note: The PWM period will remain static in this case.

41.6.3.4 Ramp Operations

Three ramp operation modes are supported. All of them require the timer/counter running in the single-slope PWM generation. The Ramp mode is selected by writing to the Ramp Mode bits in the Waveform Control register (WAVE.RAMP).

RAMP1 Operation

This is the default PWM operation, described in Single-Slope PWM Generation. See *Single-Slope PWM Generation* from Related Links.

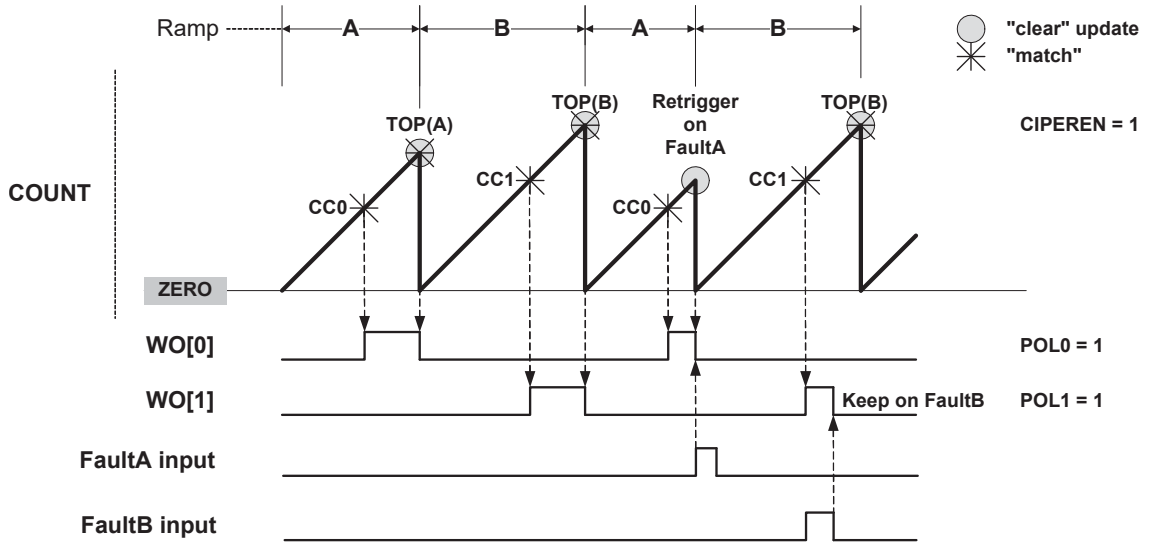
RAMP2 Operation

These operation modes are dedicated for Power Factor Correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved; see Figure 41-18. In cycle A, odd channel output is disabled, and, in cycle B, even channel output is disabled. The ramp index changes after each update but can be software modified using the Ramp index command bits in the Control B Set register (CTRLBSET.IDXCMD).

Standard RAMP2 (RAMP2) Operation

Ramp A and B periods are controlled by the PER register value. The PER value can be different on each ramp by the Circular Period buffer option in the Wave register (WAVE.CIPEREN = 1). This mode uses a two-channel TCC to generate two output signals or one output signal with another CC channel enabled in Capture mode.

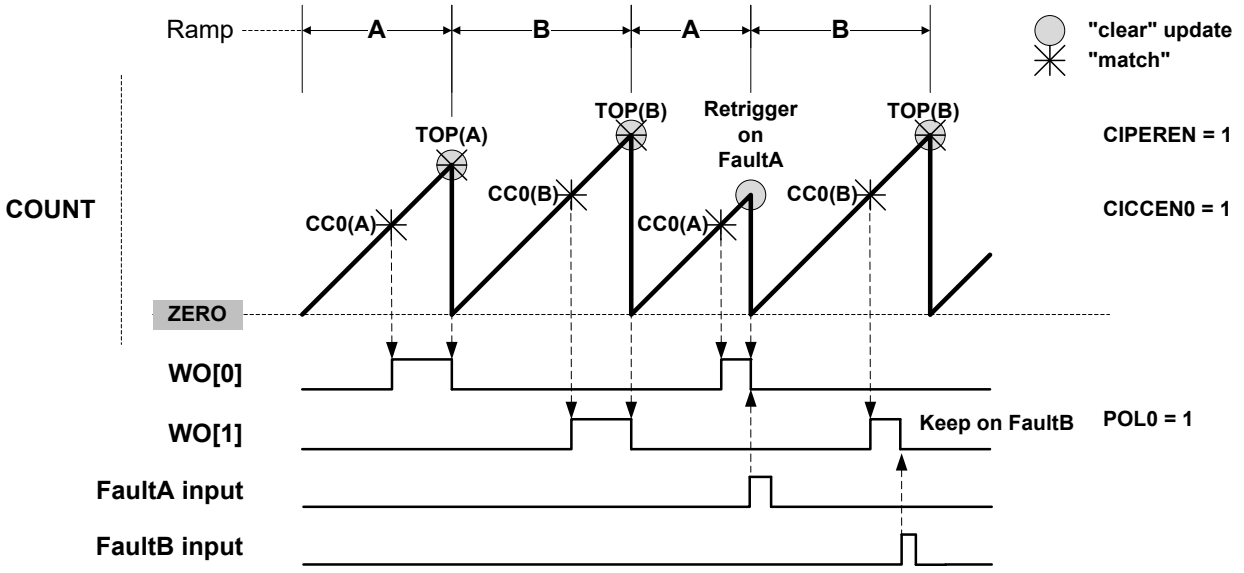
Figure 41-18. RAMP2 Standard Operation



Alternate RAMP2 (RAMP2A) Operation

Alternate RAMP2 operation is similar to RAMP2 but CC0 controls both WO[0] and WO[1] waveforms when the corresponding circular buffer option is enabled (CIPEREN = 1). The waveform polarity is the same on both outputs. Channel 1 can be used in capture mode.

Figure 41-19. RAMP2 Alternate Operation



Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated with the use of circular buffers. In this mode, CC0 is controlling the period of ramp A and PER is controlling the period of ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of ramp A if WAVE.POL0 = 0.

Figure 41-20. RAMP2 Critical Operation With More Than 2 Channels

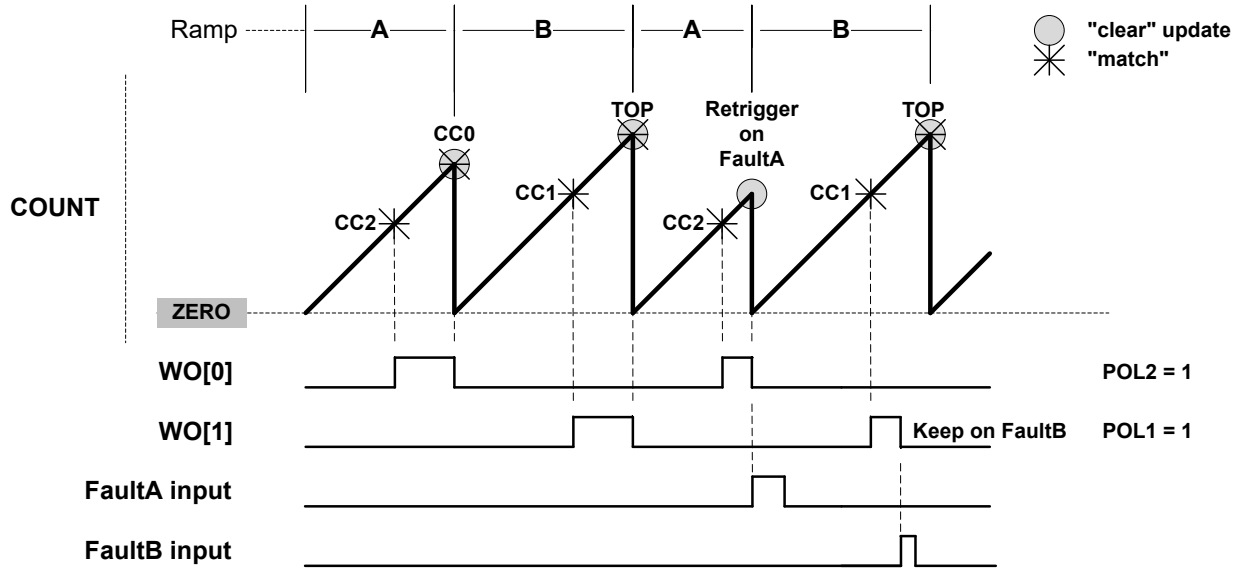
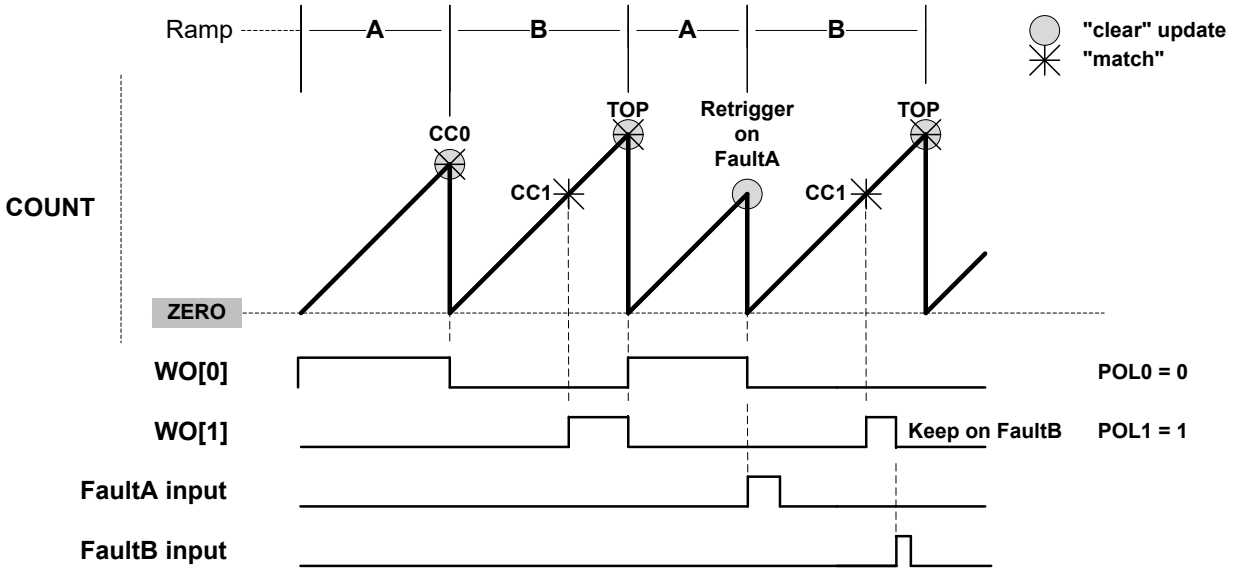


Figure 41-21. RAMP2 Critical Operation With 2 Channels



Related Links

41.6.2.5.5. Single-Slope PWM Operation

41.6.3.5 Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to an inactive state either as long as the fault condition is present or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.

Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering	By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, and, also, in device power modes where the clock is not available. To avoid false fault detection on external events (for example, due to a glitch on an I/O port), a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event is discarded. A valid event is delayed by FILTERVAL clock cycles.
Fault Blanking	This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge-triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL). The blanking time $t_b$is calculated by

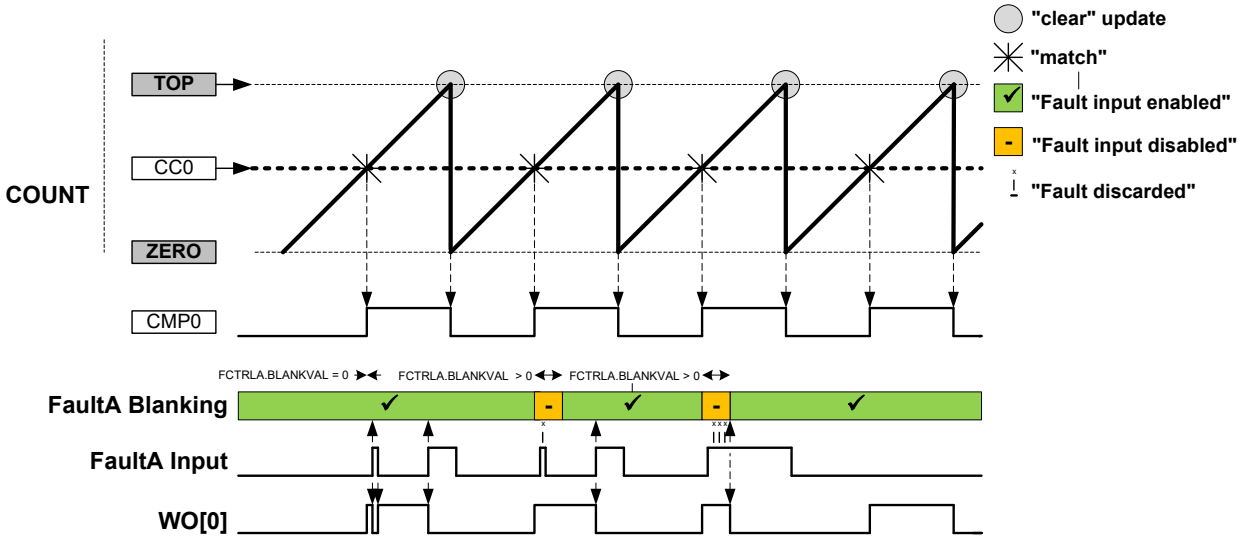
$$t_b = \frac{1 + \text{BLANKVAL}}{f_{\text{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{\text{GCLK\_TCCx\_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\text{GCLK\_TCCx}}$.

The prescaler is enabled by writing '1' to the Fault n Blanking Prescaler bit (FCTRLn.BLANKPRESC). When disabled, $f_{\text{GCLK\_TCCx\_PRESC}} = f_{\text{GCLK\_TCCx}}$. When enabled, $f_{\text{GCLK\_TCCx\_PRESC}} = f_{\text{GCLK\_TCCx}}/64$.

The maximum blanking time (FCTRLn.BLANKVAL = 255) at $f_{\text{GCLK\_TCCx}}$ = 64 MHz is 4 µs (no prescaler) or 256 µs (prescaling). For $f_{\text{GCLK\_TCCx}}$ = 1 MHz, the maximum blanking time is either 256 µs (no prescaling) or 16.4 ms (prescaling enabled).

Figure 41-22. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is enabled (FCTRLn.QUAL = 1), the fault input is disabled all the time the corresponding channel output has an inactive level, as illustrated in the following figures.

Figure 41-23. Fault Qualification in RAMP1 Operation

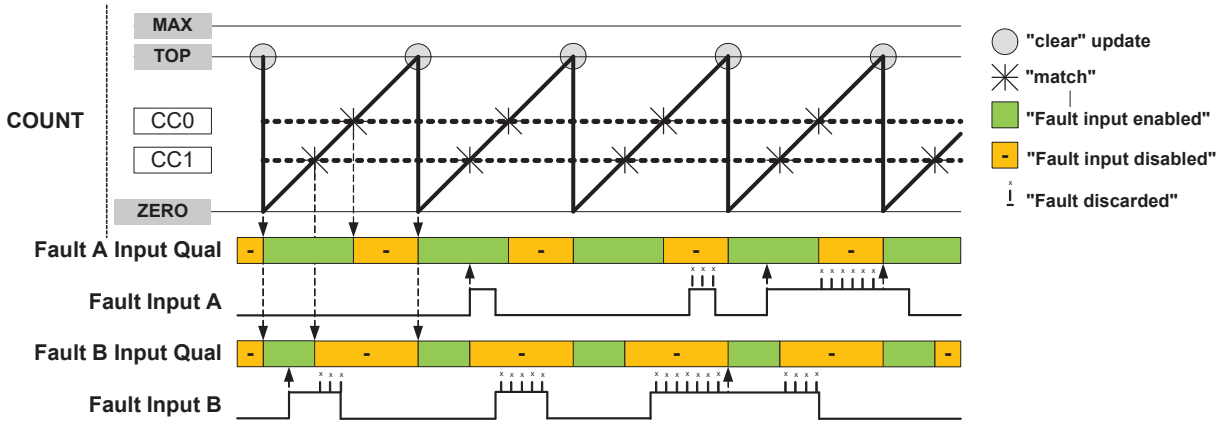
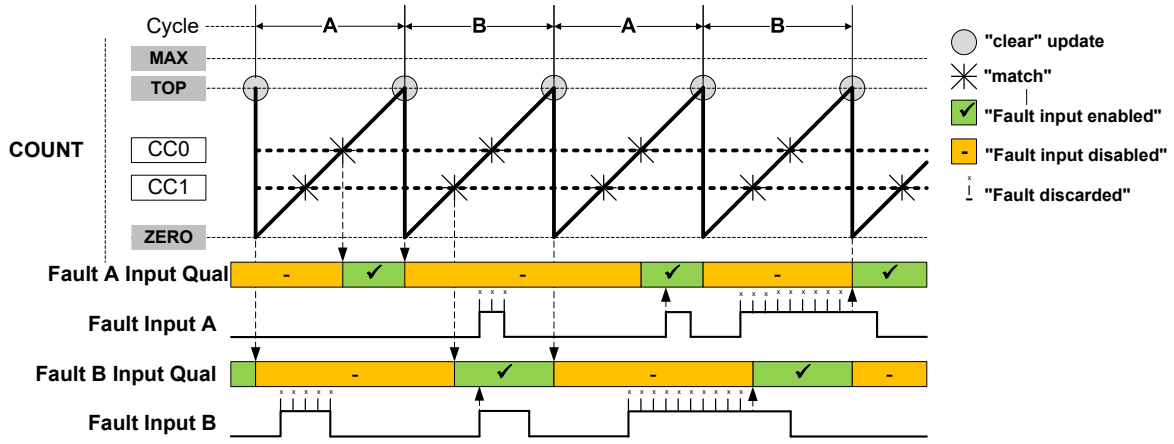


Figure 41-24. Fault Qualification in RAMP2 Operation with Inverted Polarity



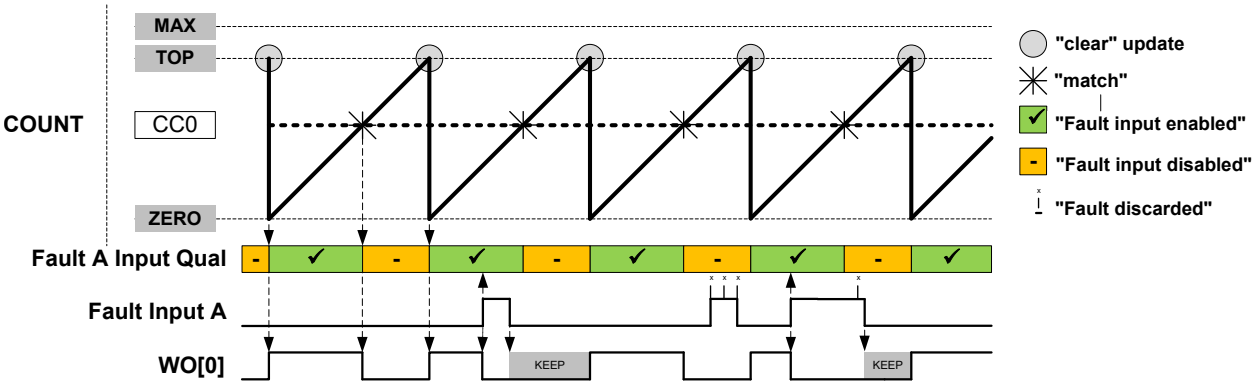
Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; hence, two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

Keep Action

This is enabled by writing the Fault n Keeper bit in the Recoverable Fault n Configuration register (FCTRLn.KEEP) to '1'. When enabled, the corresponding channel output is clamped to '0' as long as the fault condition is present. The clamp is released on the start of the first cycle after the fault condition is no longer present; see Figure 41-25.

Figure 41-25. Waveform Generation with Fault Qualification and Keep Action



Restart Action

This is enabled by writing the Fault n Restart bit in Recoverable Fault n Configuration register (FCTRLn.RESTART) to '1'. When enabled, the timer/counter is restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle; see Figure 41-26. In Ramp 1 mode, when the new cycle starts, the compare outputs is clamped to inactive level as long as the fault condition is present.

Notes: For the RAMP2 operation, when a new timer/counter cycle starts, the cycle index will change automatically; see Figure 41-27. Fault A and Fault B are qualified only during the cycle A and cycle B respectively:

- Fault A is disabled during cycle B
- Fault B is disabled during cycle A

Figure 41-26. Waveform Generation in RAMP1 mode with Restart Action

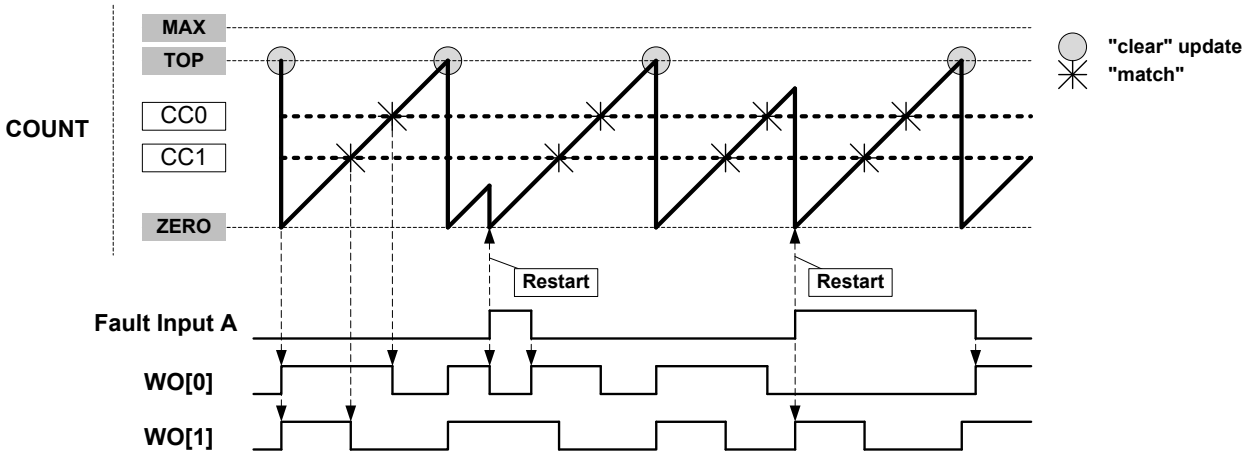
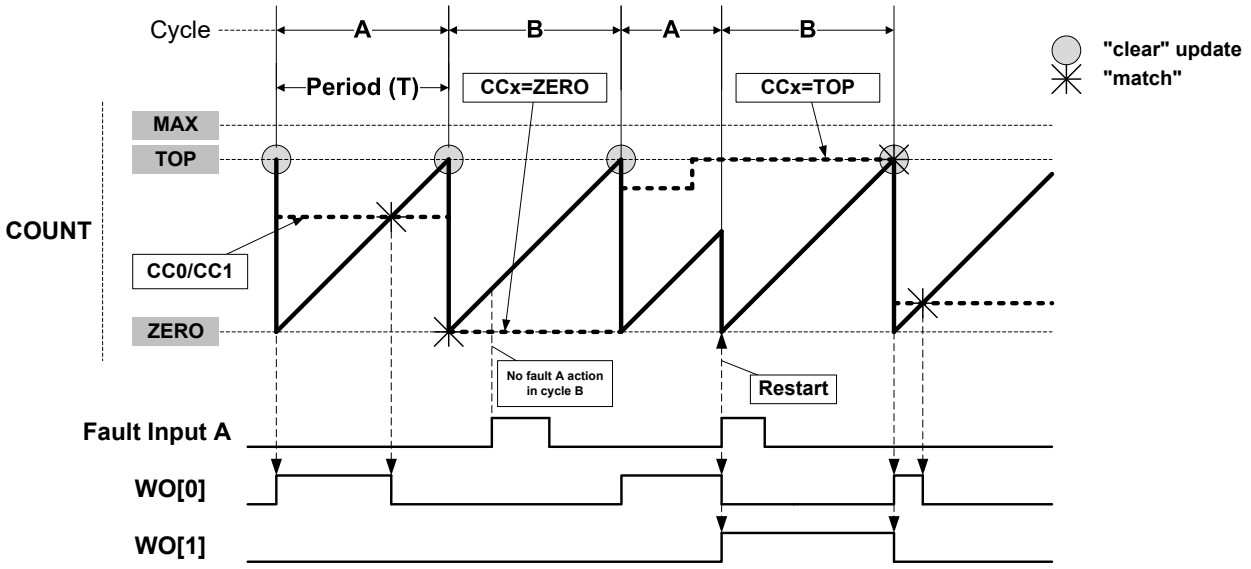


Figure 41-27. Waveform Generation in RAMP2 mode with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT – The equivalent to a standard capture operation; see *Capture Operations* from Related Links
- CAPTMIN – Gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued
- CAPTMAX – Gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued; see Figure 41-28
- LOCMIN – Notifies by event or interrupt when a local minimum captured value is detected
- LOCMAX – Notifies by event or interrupt when a local maximum captured value is detected
- DERIV0 – Notifies by event or interrupt when a local extreme captured value is detected; see Figure 41-29

CCx Content:

In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum captured values; see Figure 41-28. In LOCMIN, LOCMAX or DERIV0 operation, CCx follows the counter value at fault time; see Figure 41-29.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCx register value to a value different from zero (for CAPTMIN) or top (for CAPTMAX). If the CCx register initial value is zero (for CAPTMIN) or top (for CAPTMAX), no captures are performed using the corresponding channel.

MCx Behavior:

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCx interrupt flag is set only when the captured value is above or equal (for LOCMIN) or below or equal (for LOCMAX) to the previous captured value. So the interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value is detected. DERIV0 is equivalent to an OR function of (LOCMIN, LOCMAX).

In CAPT operation, capture is performed on each capture event. The MCx interrupt flag is set on each new capture.

In the CAPTMIN and CAPTMAX operation, the capture is performed only when, on capture event time, the counter value is lower (for CAPTMIN) or higher (for CAPMAX) than the last captured value. The MCx interrupt flag is set only when, on capture event time, the counter value is higher or equal (for CAPTMIN) or lower or equal (for CAPTMAX) to the value captured on the previous event. So the interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value is detected.

Interrupt Generation:

In CAPT mode, an interrupt is generated on each filtered Fault n and each dedicated CCx channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 41-28. Capture Action CAPTMAX

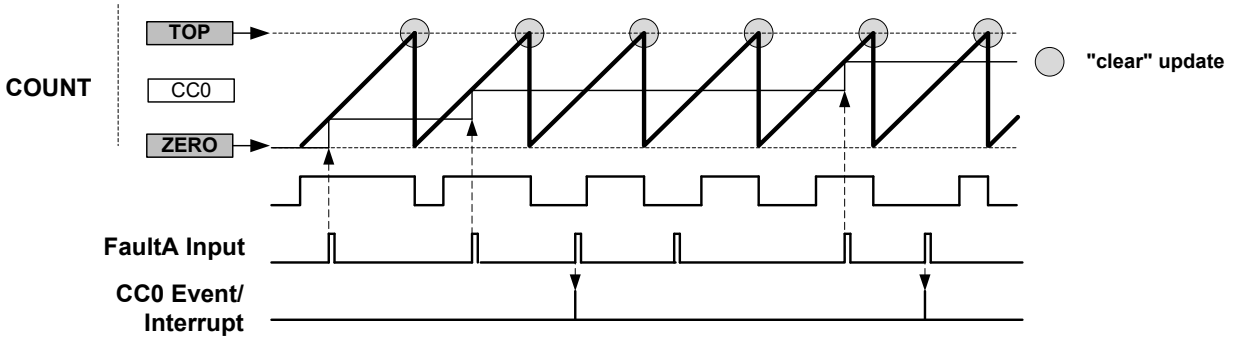
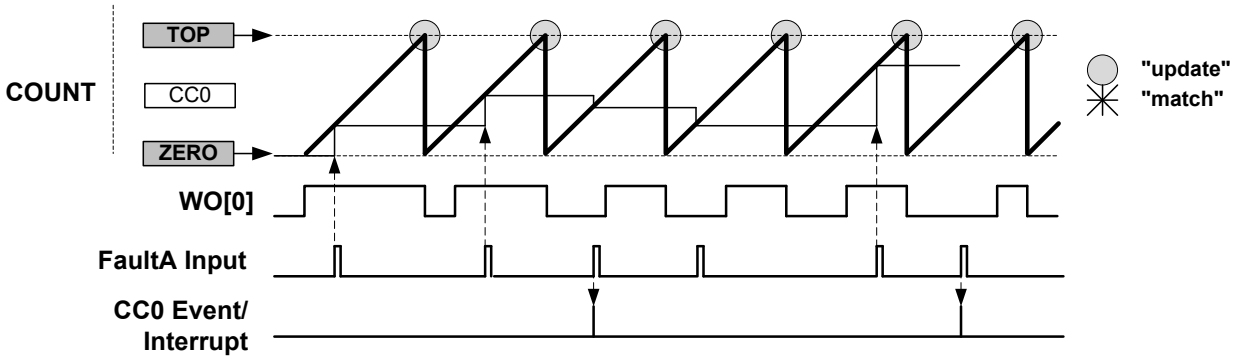


Figure 41-29. Capture Action DERIV0



Hardware Halt Action

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The Figure 41-30 illustrates an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The Figure 41-32 illustrates a similar example but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note: In RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index automatically changes.

Figure 41-30. Waveform Generation with Halt and Restart Actions

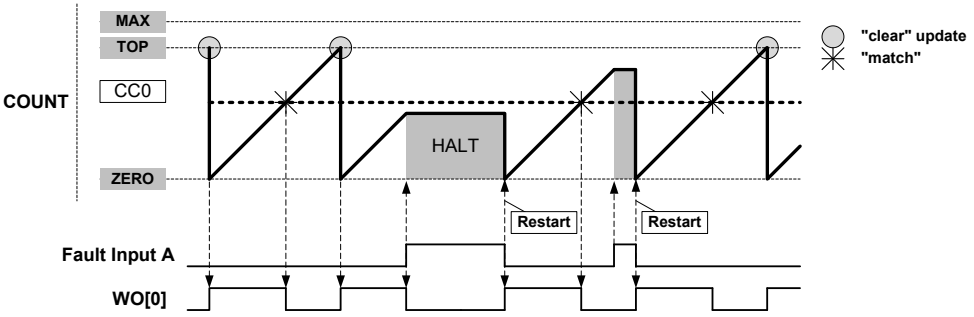
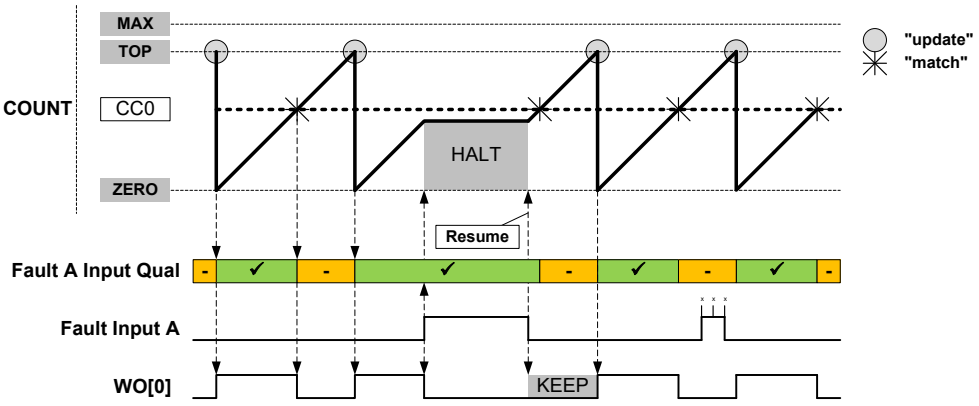
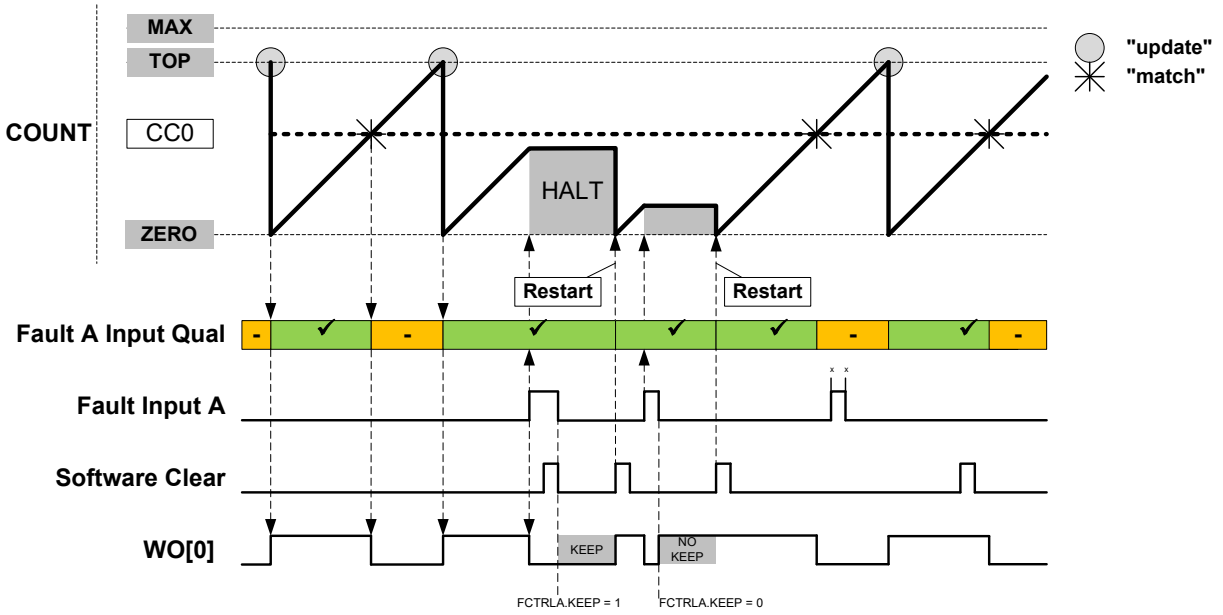


Figure 41-31. Waveform Generation with Fault Qualification and Halt



Software Halt Action This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action but, to restart the timer/counter, the corresponding fault condition must not be present anymore and the corresponding FAULT n bit in the STATUS register must be cleared by software.

Figure 41-32. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



Related Links

41.6.2.7. Capture Operations

41.6.3.6 Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (for example, a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register

(DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD = 1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

41.6.3.7 Time-Stamp Capture on Events or I/Os

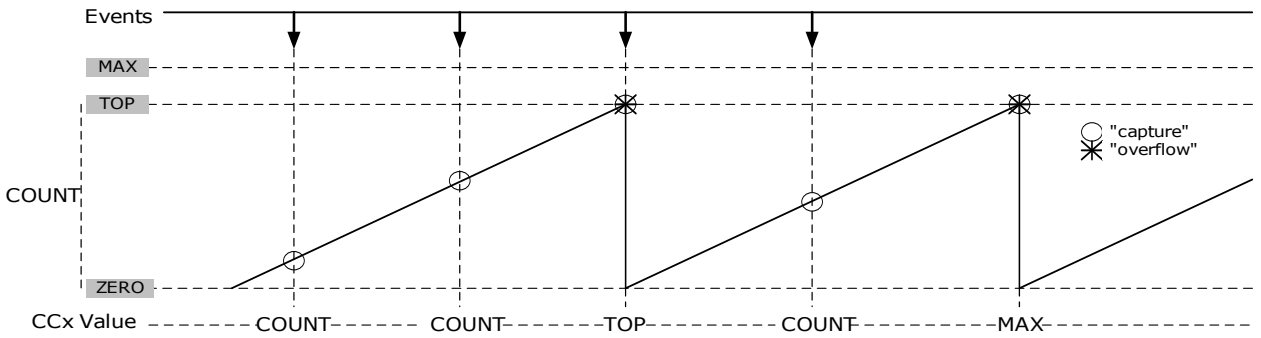
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event from the Event System or the I/O pin is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 41-33. Time Stamp



41.6.3.8 Waveform Extension

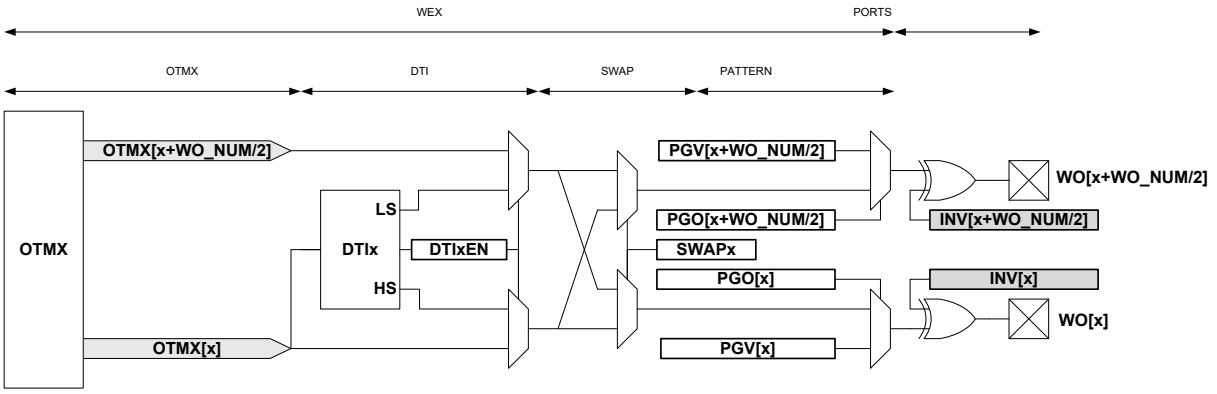
The following figure illustrates the schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as four port pair slices:

- Slice 0 DTI0/SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1/SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And in general:

- Slice n DTIx/SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])

Figure 41-34. Waveform Extension Stage Details



The OTMX unit distributes compare channels according to the selectable configurations in the following table.

Table 41-6. Output Matrix Channel Pin Routing Configuration

WEXCTRL.OTMX	OTMX[5]	OTMX[4]	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC5	CC4	CC3	CC2	CC1	CC0
0x1	CC2	CC1	CC0	CC2	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC0

- Configuration 0x0 is the default configuration. The channel location is the default one and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0] and Channel 1 to OTMX[1]. If there are more outputs than channels, channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
  Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling the flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings with a boost stage.

The following table defines an example showing four compare channels on four outputs.

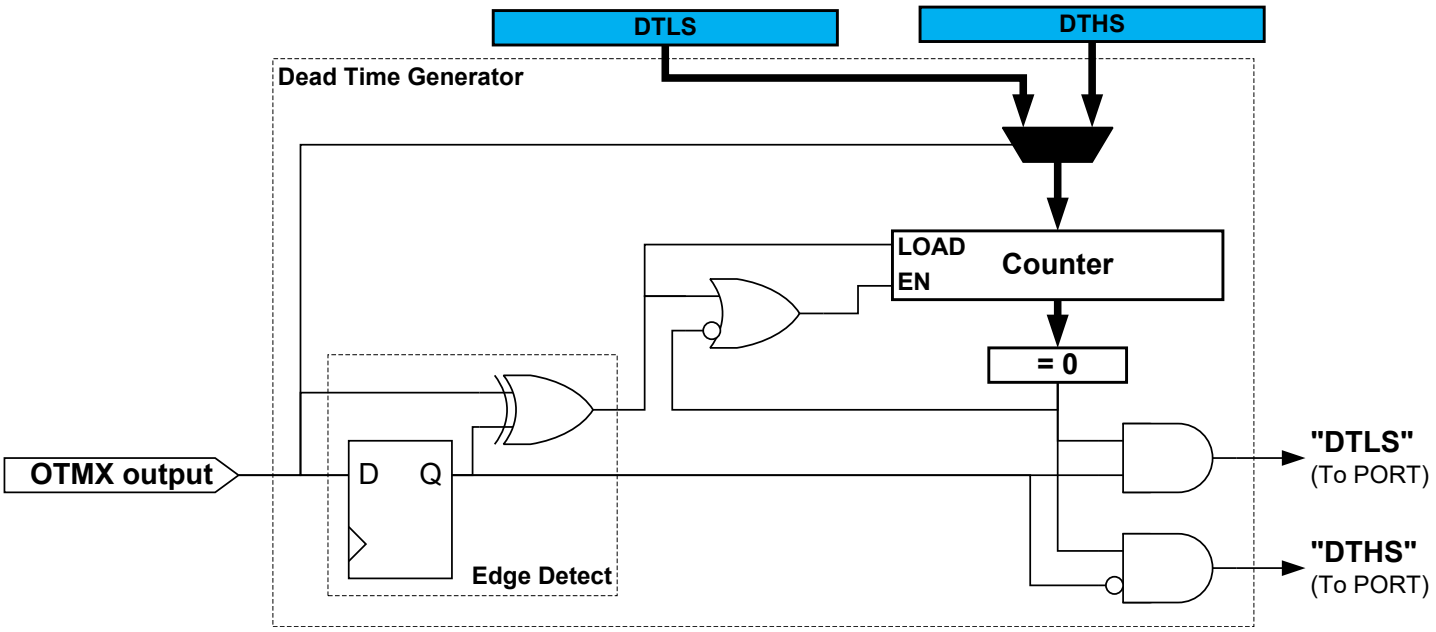
Table 41-7. Four Compare Channels on Four Outputs

WEXCTRL.OTMX	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC0

The DTI unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS will never switch simultaneously.

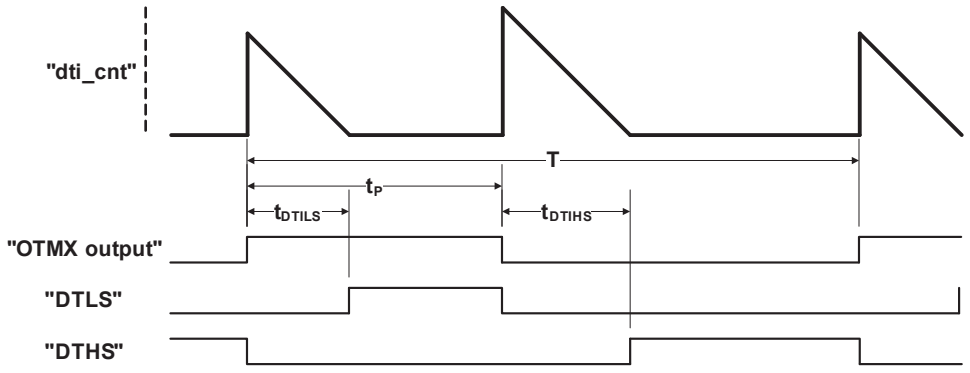
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. The following figure illustrates the block diagram of one DTI generator. The four channels have a common register that controls the dead time, which is independent of high side and low side setting.

Figure 41-35. Dead-Time Generator Block Diagram



As illustrated in the following figure, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the Output Matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge), it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge), it reloads the DTHS register.

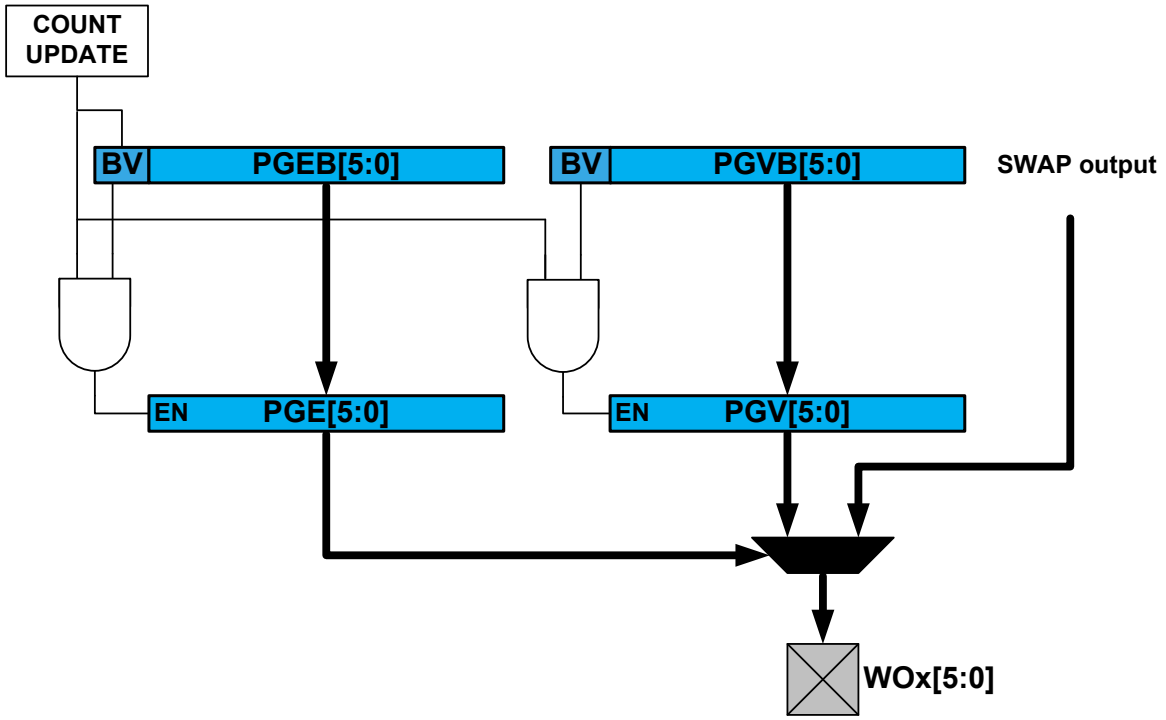
Figure 41-36. Dead-Time Generator Timing Diagram



The pattern generator unit produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation sequence

in Brushless DC motors (BLDC), stepper motors and full bridge control as illustrated in the following figure.

Figure 41-37. Pattern Generator Block Diagram



As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access the PATT.PGE, PATT.PGV bits registers directly.

41.6.4 Host/Client Operation

Two or more TCC instances sharing the same GCLK_TCCx clock, can be linked to provide more synchronized CC channels. The operation is enabled by setting the Host Synchronization bit in Control A register (CTRLA.MSYNC) in the Client instance. When the bit is set, the Client TCC instance will synchronize the CC channels to the Host counter.

41.6.5 DMA, Interrupts and Events

Table 41-8. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow/Underflow	Yes	Yes	—	Yes(1)	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes(2)	Yes(3)	For circular buffering: on DMA acknowledge For capture channel: when CCx register is read
Retrigger	Yes	Yes	—	—	—
Count	Yes	Yes	—	—	—
Capture Overflow Error	Yes	—	—	—	—
Debug Fault State	Yes	—	—	—	—
Recoverable Faults	Yes	—	—	—	—
Non-Recoverable Faults	Yes	—	—	—	—

.....continued					
Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
TCCx Event 0 input	—	—	Yes(4)	—	—
TCCx Event 1 input	—	—	Yes(5)	—	—

Notes:

1. DMA request set on Overflow, Underflow or Re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In Capture or Circular modes.
4. On event input, either action can be executed:
   - Re-trigger counter
   - Control counter direction
   - Stop the counter
   - Decrement the counter
   - Perform period and pulse width capture
   - Generate non-recoverable fault
5. On event input, either action can be executed:
   - Re-trigger counter
   - Increment or decrement counter depending on direction
   - Start the counter
   - Increment or decrement counter based on direction
   - Increment counter regardless of direction
   - Generate non-recoverable fault

41.6.5.1 DMA Operation

The TCC can generate the following DMA requests:

Counter overflow (OVF)	If the One-shot Trigger mode in the control A register (CTRLA.DMAOS) is written to '0', the TCC generates a DMA request on each cycle when an update condition (Overflow, Underflow or Re-trigger) is detected. When an update condition (Overflow, Underflow or Re-trigger) is detected while CTRLA.DMAOS = 1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD = DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCx)	A DMA request is set only on a compare match if CTRLA.DMAOS = 0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS = 1, the DMA requests are not generated.
Channel Capture (MCx)	For a capture channel, the request is set when valid data is present in the CCx register, and cleared when the CCx register is read. In this operation mode, the CTRLA.DMAOS bit value is ignored.

DMA Operation with Circular Buffer

When circular buffer operation is enabled, the Buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.

Note: Circular buffers are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

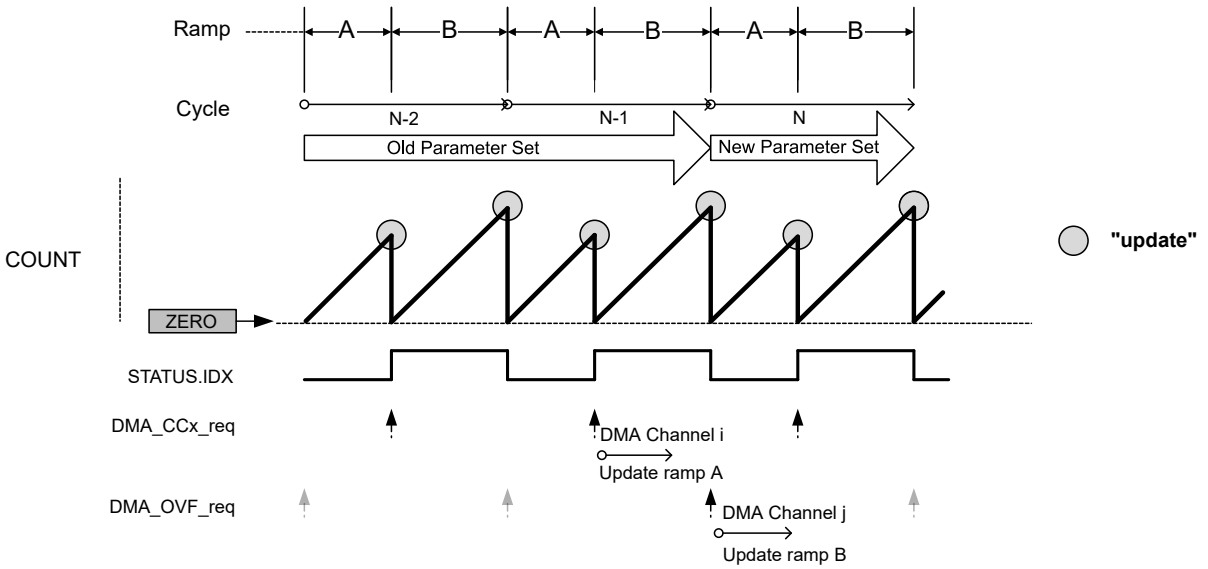
DMA Operation with Circular Buffer in RAMP2 and RAMP2A Mode:

When a CCx channel is selected as a circular buffer, the related DMA request is not set on a compare match detection but on start of ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of ramp A with an effective DMA transfer on previous ramp B (DMA acknowledge).

The update of all circular buffer values for ramp A can be done through a DMA channel triggered on an MC trigger. The update of all circular buffer values for ramp B can be done through a second DMA channel triggered by the overflow DMA request.

Figure 41-38. DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled

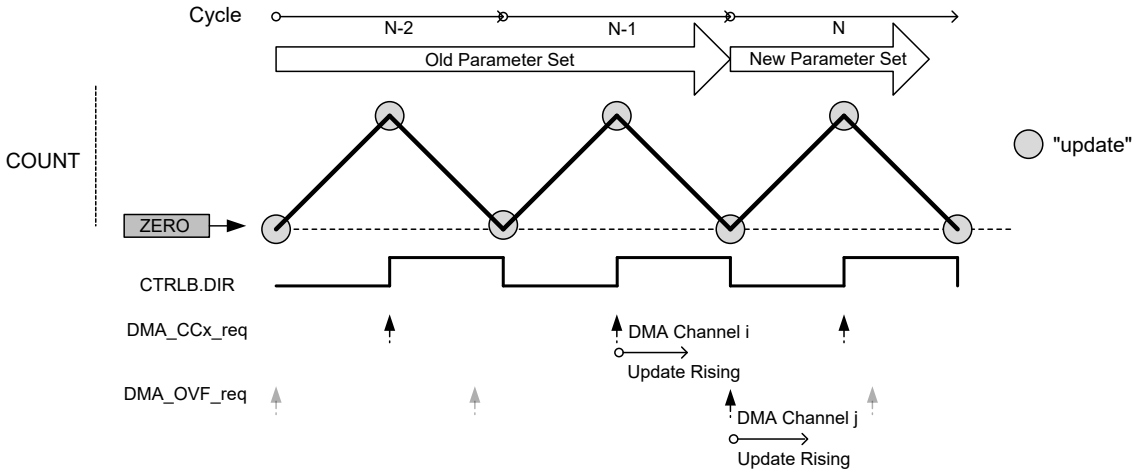


DMA Operation with Circular Buffer in DSBOTH Mode:
When a CC channel is selected as a circular buffer, the related DMA request is not set on a compare match detection but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of the up-counting phase with an effective DMA transfer on the previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by an MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 41-39. DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled



41.6.5.2 Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) – Refer also to the description of EVCTRL.CNTSEL
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources.

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. 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 TCC is reset. See *INTFLAG* from Related Links for details on how to clear Interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which Interrupt condition is present.

Interrupts must be globally enabled for interrupt requests to be generated. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

9.2. Nested Vector Interrupt Controller (NVIC)

41.8.12. INTFLAG

41.6.5.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) (For further details, refer to the EVCTRL.CNTSEL description.)
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. See *Event System (EVSYS)* from Related Links.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start – Start counting on the event rising edge. Further events will not restart the counter; the counter will keep counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). See *EVCTRL* from Related Links.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on input event.

Note: When several events are connected to the TCC, the enabled action will apply for each of the incoming events. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

Related Links

30. Event System (EVSYS)
41.8.9. EVCTRL

41.6.6 Sleep Mode Operation

The TCC can be configured to operate in any sleep mode (Standby Sleep, Idle). To be able to run in standby sleep mode, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This

peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

41.6.7 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 and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

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.

41.7 Register Summary

See the *TCCx (x = 0 to 2)* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the TCC instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		RESOLUTION[1:0]					ENABLE	SWRST
		15:8	MSYNC	ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
		23:16	DMAOS							
		31:24			CPTEN5	CPTEN4	CPTEN3	CPTEN2	CPTEN1	CPTEN0
0x04	CTRLBCLR	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x06 ... 0x07	Reserved									
0x08	SYNCBUSY	7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x0C	FCTRLA	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		23:16	BLANKVAL[7:0]							
		31:24					FILTERVAL[3:0]			
0x10	FCTRLB	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		23:16	BLANKVAL[7:0]							
		31:24					FILTERVAL[3:0]			
0x14	WEXCTRL	7:0							OTMX[1:0]	
		15:8						DTIEN2	DTIEN1	DTIEN0
		23:16	DTLS[7:0]							
		31:24	DTHS[7:0]							
0x18	DRVCTRL	7:0			NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
		15:8			NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
		23:16			INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
		31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
0x1C ... 0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
		15:8	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
		23:16			MCEI5	MCEI4	MCEI3	MCEI2	MCEI1	MCEI0
		31:24			MCEO5	MCEO4	MCEO3	MCEO2	MCEO1	MCEO0
0x24	INTENCLR	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16			MC5	MC4	MC3	MC2	MC1	MC0
		31:24								
0x28	INTENSET	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16			MC5	MC4	MC3	MC2	MC1	MC0
		31:24								
0x2C	INTFLAG	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16			MC5	MC4	MC3	MC2	MC1	MC0
		31:24								
0x30	STATUS	7:0	PERBUFV		PATTBUFV	CLIENT	DFS	UFS	IDX	STOP
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
		23:16			CCBUFV5	CCBUFV4	CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
		31:24			CMP5	CMP4	CMP3	CMP2	CMP1	CMP0

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24								
0x38	PATT	7:0			PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
		15:8			PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
0x3A ... 0x3B	Reserved									
0x3C	WAVE	7:0	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
		23:16			POL5	POL4	POL3	POL2	POL1	POL0
		31:24						SWAP2	SWAP1	SWAP0
0x40	PER	7:0	PER[1:0]		DITHER[5:0]					
		15:8	PER[9:2]							
		23:16	PER[17:10]							
		31:24								
0x44	CCx0	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x48	CCx1	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x4C	CCx2	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x50	CCx3	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x54	CCx4	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x58	CCx5	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24								
0x5C ... 0x63	Reserved									
0x64	PATTBUF	7:0			PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
		15:8			PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
0x66 ... 0x6B	Reserved									
0x6C	PERBUF	7:0	PERBUF[1:0]		DITHERBUF[5:0]					
		15:8	PERBUF[9:2]							
		23:16	PERBUF[17:10]							
		31:24								
0x70	CCBUFx0	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								
0x74	CCBUFx1	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								

.....continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	CCBUFx2	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								
0x7C	CCBUFx3	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								
0x80	CCBUFx4	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								
0x84	CCBUFx5	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24								

Related Links

8. Product Memory Mapping Overview

41.8 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 require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

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

41.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized (ENABLE, SWRST)

Bit	31	30	29	28	27	26	25	24
			CPTEN5	CPTEN4	CPTEN3	CPTEN2	CPTEN1	CPTEN0
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
	DMAOS							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	MSYNC	ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[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
		RESOLUTION[1:0]					ENABLE	SWRST
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bits 24, 25, 26, 27, 28, 29 – CPTEN Capture Channel x Enable
These bits are used to select the capture or compare operation on channel x.
Writing a '1' to CPTENx enables capture on channel x.
Writing a '0' to CPTENx disables capture on channel x.

Bit 23 – DMAOS DMA One-Shot Trigger Mode
This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit generates a DMA trigger on the TCC cycle following a TCC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit generates DMA triggers on each TCC cycle.
This bit is not synchronized.
Note: DMA One-Shot mode is not available in RAMP1/RAMP2C/RAMP2CS modes.

Value	Description
0	The TCC controls its own counter.
1	The counter is controlled by its Host TCC.

Bit 15 – MSYNC Host Synchronization (only for TCC client instance)
This bit must be set if the TCC counting operation must be synchronized on its Host TCC.
This bit is not synchronized.

Value	Description
0	The TCC controls its own counter.
1	The counter is controlled by its Host TCC.

Bit 14 – ALOCK Auto Lock
This bit is not synchronized.

Value	Description
0	The Lock Update bit in the Control B register (CTRLB.LUPD) is not affected by overflow/underflow and re-trigger events
1	CTRLB.LUPD is set to '1' on each overflow/underflow or re-trigger event.

Bits 13:12 – PRESCYNC[1:0] Prescaler and Counter Synchronization

These bits select if, on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on the re-trigger event.
These bits are not synchronized.

Value	Name	Description	
		Counter Reloaded	Prescaler
0x0	GCLK	Reload or reset Counter on next GCLK	—
0x1	PRESC	Reload or reset Counter on next prescaler clock	—
0x2	RESYNC	Reload or reset Counter on next GCLK	Reset prescaler counter
0x3	Reserved	—	—

Bit 11 – RUNSTDBY Run in Standby

This bit is used to keep the TCC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TCC is halted in Standby mode.
1	The TCC continues to run in Standby mode.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the Counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TCC
0x1	DIV2	Prescaler: GCLK_TCC/2
0x2	DIV4	Prescaler: GCLK_TCC/4
0x3	DIV8	Prescaler: GCLK_TCC/8
0x4	DIV16	Prescaler: GCLK_TCC/16
0x5	DIV64	Prescaler: GCLK_TCC/64
0x6	DIV256	Prescaler: GCLK_TCC/256
0x7	DIV1024	Prescaler: GCLK_TCC/1024

Bits 6:5 – RESOLUTION[1:0] Dithering Resolution

These bits increase the TCC resolution by enabling the dithering options.
These bits are not synchronized.

Table 41-9. Dithering

Value	Name	Description
0x0	NONE	The dithering is disabled.
0x1	DITH4	Dithering is done every 16 PWM frames. PER[3:0] and CCx[3:0] contain dithering pattern selection.
0x2	DITH5	Dithering is done every 32 PWM frames. PER[4:0] and CCx[4:0] contain dithering pattern selection.
0x3	DITH6	Dithering is done every 64 PWM frames. PER[5:0] and CCx[5:0] contain dithering pattern selection.

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately, and the ENABLE bit in the

SYNCBUSY register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is 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 TCC (except DBGCTRL) to their initial state and the TCC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence; all other writes in the same write-operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

41.8.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

The user can change this register without doing a read-modify-write operation. Changes in this register will, also, be reflected in the Control B Set (CTRLBSET) register.

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
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 7:5 – CMD[2:0] TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command is executed, the CMD bit field reads back as '0'. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing a '0' to this bit group has no effect.
Writing a '1' to any of these bits clears the pending command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Clear start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force COUNT read synchronization
0x5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A and cycle B changes in the RAMP2 and RAMP2A operation. On the timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing a '0' to these bits has no effect.
Writing a '1' to any of these bits clears the pending command.

Value	Name	Description
0x0	DISABLE	DISABLE Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT One-Shot
This bit controls the one-shot operation of the TCC. When the one-shot operation is enabled, the TCC stops counting on the next overflow/underflow condition or on a stop command.
Writing a '0' to this bit has no effect
Writing a '1' to this bit disables the one-shot operation.

Value	Description
0	The TCC will update the counter value on overflow/underflow condition and continue operation.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is cleared, the hardware UPDATE registers, with the value from their buffered registers, are enabled.
This bit has no effect when the input capture operation is enabled.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the registers updates on the hardware UPDATE condition.

Value	Description
0	The CCBx, PERB, PGVB and PGEB buffer registers values are copied into the corresponding CCx, PER, PGV and PGE registers on hardware update condition.
1	The CCBx, PERB, PGVB and PGEB buffer registers values are not copied into the corresponding CCx, PER, PGV and PGE registers on hardware update condition.

Bit 0 – DIR Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit set the bit and makes the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

41.8.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

The user can change this register without doing a read-modify-write operation. Changes in this register will, also, be reflected in the Control B Clear (CTRLBCLR) register.

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
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 7:5 – CMD[2:0] TCC Command
These bits can be used for software control of the re-triggering and stop commands of the TCC. When a command is executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing a '0' to this bit has no effect.
Writing a valid value to this bit group, as shown in the following table, will set the associated command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT
0x5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A and cycle B changes in the RAMP2 and RAMP2A operation. On the timer/counter update condition, the command is executed, the IDX flag in the STATUS register is updated and the IDXCMD command is cleared.
Writing a '0' to this bit has no effect.
Writing a valid value to these bits will set a command.

Value	Name	Description
0x0	DISABLE	Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT One-Shot
This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the one-shot operation.

Value	Description
0	The TCC will count continuously.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is set, the hardware UPDATE registers, with the value from their buffered registers, are disabled. Disabling the update ensures that all buffer registers are valid before a

hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when the input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will disable the register's updates on the hardware UPDATE condition.

Value	Description
0	The CCBx, PERB, PGVB and PGEB buffer registers values *are* copied into the corresponding CCx, PER, PGV and PGE registers on hardware update condition.
1	The CCBx, PERB, PGVB and PGEB buffer registers values are *not* copied into CCx, PER, PGV and PGE registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

41.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
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
	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER register between the clock domains is complete.
This bit is set when the synchronization of PER register between clock domains is started.

Bit 6 – WAVE WAVE Synchronization Busy
This bit is cleared when the synchronization of WAVE register between the clock domains is complete.
This bit is set when the synchronization of WAVE register between clock domains is started.

Bit 5 – PATT PATT Synchronization Busy
This bit is cleared when the synchronization of PATTERN register between the clock domains is complete.
This bit is set when the synchronization of PATTERN register between clock domains is started.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT register between the clock domains is complete.
This bit is set when the synchronization of COUNT register between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS register between the clock domains is complete.
This bit is set when the synchronization of STATUS register between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.
This bit is set when the synchronization of CTRLB register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

41.8.5 Fault Control A and B

Name: FCTRLn
Offset: 0x0C + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					FILTERVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLANKVAL[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	15	14	13	12	11	10	9	8
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[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
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0] Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCCx periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault n Blanking Value Prescaler
This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is BLANKVAL* prescaled GCLK_TCCx.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCCx.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

Table 41-10. Fault n Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault n is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local minimum detection.

.....continued		
Value	Name	Description
0x3	CAPTMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection.
0x7	CAPTMARK	Capture with ramp index as MSB value.

Bits 11:10 – CHSEL[1:0] Recoverable Fault n Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault n.

Value	Name	Description
0x0	CC0	Capture value stored into CC0
0x1	CC1	Capture value stored into CC1
0x2	CC2	Capture value stored into CC2
0x3	CC3	Capture value stored into CC3

Bits 9:8 – HALT[1:0] Recoverable Fault n Halt Operation

These bits select the halt action for recoverable Fault n.

Value	Name	Description
0x0	DISABLE	Halt action disabled
0x1	HW	Hardware halt action
0x2	SW	Software halt action
0x3	NR	Non-recoverable fault

Bit 7 – RESTART Recoverable Fault n Restart

Setting this bit enables restart action for Fault n.

Value	Description
0	Fault n restart action is disabled.
1	Fault n restart action is enabled.

Bits 6:5 – BLANK[1:0] Recoverable Fault n Blanking Operation

These bits, select the blanking start point for recoverable Fault n.

Value	Name	Description
0x0	START	Blanking applied from start of the Ramp period
0x1	RISE	Blanking applied from rising edge of the waveform output
0x2	FALL	Blanking applied from falling edge of the waveform output
0x3	BOTH	Blanking applied from each toggle of the waveform output

Bit 4 – QUAL Recoverable Fault n Qualification

Setting this bit enables the recoverable Fault n input qualification.

Value	Description
0	The recoverable Fault n input is not disabled on CMPx value condition.
1	The recoverable Fault n input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP Recoverable Fault n Keep

Setting this bit enables the Fault n keep action.

Value	Description
0	The Fault n state is released as soon as the recoverable Fault n is released.
1	The Fault n state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Recoverable Fault n Source

These bits select the TCC event input for recoverable Fault n.

Event system channel connected to MCEx event input, must be configured to route the event asynchronously, when used as a recoverable Fault n input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCEx (x=0,1) event input
0x2	INVERT	Inverted MCEx (x=0,1) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

41.8.6 Waveform Extension Control

Name: WEXCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DTHS[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	23	22	21	20	19	18	17	16
	DTLS[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	15	14	13	12	11	10	9	8
						DTIEN2	DTIEN1	DTIEN0
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 31:24 – DTHS[7:0] Dead-Time High Side Outputs Value
This register holds the number of GCLK_TCCx clock cycles for the dead-time high side.

Bits 23:16 – DTLS[7:0] Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCCx clock cycles for the dead-time low side.

Bits 8, 9, 10 – DTIENx Dead-time Insertion Generator x Enable [x=0..2]
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO_NUM/2] with the low-side and high-side waveform, respectively.

Value	Description
0	No dead-time insertion override.
1	Dead time insertion override on signal outputs[x] and [x+WO_NUM/2] from the matrix outputs[x] signal.

Bits 1:0 – OTMX[1:0] Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Waveform Extension. See *Waveform Extension* from Related Links.

Related Links
41.6.3.8. Waveform Extension

41.8.7 Driver Control

Name: DRVCTRL
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
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
			INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
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
			NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 31:28 – FILTERVAL1[3:0] Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on the TCE1 event input line. When the TCE1 event input line is configured as a synchronous event, this value must be 0x0.

Bits 27:24 – FILTERVAL0[3:0] Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on the TCE0 event input line. When the TCE0 event input line is configured as a synchronous event, this value must be 0x0.

Bits 16, 17, 18, 19, 20, 21 – INVENx Waveform Output x Inversion [x=0..5]
These bits are used to select inversion on the output of channel x.
Writing a '1' to INVENx inverts output from WO[x].
Writing a '0' to INVENx disables inversion of output from WO[x].

Bits 8, 9, 10, 11, 12, 13 – NRVx NRVx Non-Recoverable State x Output Value [x=0..5]
These bits define the value of the enabled override outputs under the non-recoverable fault condition.

Bits 0, 1, 2, 3, 4, 5 – NREx Non-Recoverable State x Output Enable [x=0..5]
These bits enable the override of individual outputs by NRVx value under the non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

41.8.8 Debug control

Name: DBGCTRL
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access						R/W		R/W
Reset						0		0

Bit 2 – FDDBD Fault Detection on Debug Break Detection
This bit is not affected by software Reset and must not be changed by software while the TCC is enabled.
By default, this bit is '0', and the On-Chip Debug (OCD) fault protection is disabled. When this bit is written to '1', an OCD break request from the OCD system triggers a non-recoverable fault. When this bit is set, the OCD fault protection is enabled and an OCD break request from the OCD system triggers a non-recoverable fault.

Value	Description
0	No faults are generated when TCC is halted in Debug mode.
1	A non recoverable fault is generated and FAULTD flag is set when TCC is halted in Debug mode.

Bit 0 – DBGRUN Debug Running State
This bit is not affected by a software Reset and must not be changed by software while the TCC is enabled.

Value	Description
0	The TCC is halted when the device is halted in Debug mode.
1	The TCC continues normal operation when the device is halted in Debug mode.

41.8.9 Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
			MCEO5	MCEO4	MCEO3	MCEO2	MCEO1	MCEO0
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
			MCEI5	MCEI4	MCEI3	MCEI2	MCEI1	MCEI0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
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	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[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

Bits 24, 25, 26, 27, 28, 29 – MCEOx Match or Capture Channel x Event Output Enable [x=0..5]
These bits control if the match/capture event on channel x is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture x event is disabled and will not be generated.
1	Match/capture x event is enabled and will be generated for every compare/capture on channel x.

Bits 16, 17, 18, 19, 20, 21 – MCEIx Match or Capture Channel x Event Input Enable [x=0..3]
These bits indicate if the match/capture x incoming event is enabled.
These bits are used to enable match or capture input events to the CCx channel of TCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 14, 15 – TCEIx Timer/Counter Event Input x Enable [x=0..1]
This bit is used to enable input event x to the TCC.

Value	Description
0	Incoming event x is disabled.
1	Incoming event x is enabled.

Bits 12, 13 – TCINVx Timer/Counter Event x Invert Enable [x=0..1]
This bit inverts the event x input.

Value	Description
0	Input event source x is not inverted.
1	Input event source x is inverted.

Bit 10 – CNTEO Timer/Counter Event Output Enable

This bit is used to enable the counter cycle event. When enabled, an event will be generated on the beginning or end of the counter cycle depending on the CNTSEL[1:0] settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depending on CNTSEL[1:0] value.

Bit 9 – TRGEO Retrigger Event Output Enable

This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

Value	Description
0	Counter retrigger event is disabled and will not be generated.
1	Counter retrigger event is enabled and will be generated for every counter retrigger.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable

This bit is used to enable the overflow/underflow event. When enabled, an event will be generated when the counter reaches the TOP or the ZERO value.

Value	Description
0	Overflow/underflow counter event is disabled and will not be generated.
1	Overflow/underflow counter event is enabled and will be generated for every counter overflow/underflow.

Bits 7:6 – CNTSEL[1:0] Timer/Counter Interrupt and Event Output Selection

These bits define on which part of the counter cycle the counter event output is generated.

Value	Name	Description
0x0	BEGIN	An interrupt/event is generated at begin of each counter cycle
0x1	END	An interrupt/event is generated at end of each counter cycle
0x2	BETWEEN	An interrupt/event is generated between each counter cycle.
0x3	BOUNDARY	An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle.

Bits 5:3 – EVACT1[2:0] Timer/Counter Event Input 1 Action

These bits define the action the TCC will perform on the TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TC on event
0x2	DIR (asynch)	Direction control
0x3	STOP	Stop TC on event
0x4	DEC	Decrement TC on event
0x5	PPW	Period captured into CC0 Pulse Width on CC1
0x6	PWP	Period captured into CC1 Pulse Width on CC0
0x7	FAULT	Non-recoverable Fault

Bits 2:0 – EVACT0[2:0] Timer/Counter Event Input 0 Action

These bits define the action the TCC will perform on TCE0 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TC on event
0x4	INC	Increment TC on EVENT
0x5	COUNT (async)	Count on active state of asynchronous event
0x6	STAMP	Capture overflow times (Max value)
0x7	FAULT	Non-recoverable Fault

41.8.10 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x24
Reset: 0x00000000
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	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			MC5	MC4	MC3	MC2	MC1	MC0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
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
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21 – MCx Match or Capture Channel x Interrupt Disable [x=0..5]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 1 interrupt.

Value	Description
0	The Non-Recoverable Fault 1 interrupt is disabled.
1	The Non-Recoverable Fault 1 interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault 0 Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 0 interrupt.

Value	Description
0	The Non-Recoverable Fault 0 interrupt is disabled.
1	The Non-Recoverable Fault 0 interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA Recoverable Fault A Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS Non-Recoverable Debug Fault State Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS Non-Recoverable Update Fault Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT Counter Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

41.8.11 Interrupt Enable Set

Name: INTENSET
Offset: 0x28
Reset: 0x00000000
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 Interrupt Enable Clear (INTENCLR) register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			MC5	MC4	MC3	MC2	MC1	MC0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
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
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21 – MC Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 1 interrupt.

Value	Description
0	The Non-Recoverable Fault 1 interrupt is disabled.
1	The Non-Recoverable Fault 1 interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault 0 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 0 interrupt.

Value	Description
0	The Non-Recoverable Fault 0 interrupt is disabled.
1	The Non-Recoverable Fault 0 interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS Non-Recoverable Debug Fault State Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS Non-Recoverable Update Fault Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which enables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT Counter Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Counter Interrupt Disable/Enable bit, which enables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

41.8.12 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			MC5	MC4	MC3	MC2	MC1	MC0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
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
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21 – MCx Match or Capture Channel x Interrupt Flag [x=0..5]
This flag is set on the next CLK_TCC_COUNT cycle after a match with the compare condition or when the CCx register contains a valid capture value.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits clears the corresponding Match or Capture Channel x interrupt flag.
In the Capture operation, this flag is automatically cleared when the CCx register is read.

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Non-Recoverable Fault 1 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 1 interrupt flag.

Bit 14 – FAULT0 Non-Recoverable Fault 0 Interrupt Flag
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 0 interrupt flag.

Bit 13 – FAULTB Recoverable Fault B Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 12 – FAULTA Recoverable Fault A Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault A occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault A interrupt flag.

Bit 11 – DFS Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Debug Fault State occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 10 – UFS Non-Recoverable Update Fault Interrupt Flag
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Update Fault Interrupt Flag.

Bit 3 – ERR Error Interrupt Flag
This flag is set if a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is one. In which case, there is no place to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 2 – CNT Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Retrigger interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

41.8.13 Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: -

Note: Clear STATUS register bits by 32-bits write access only.

Bit	31	30	29	28	27	26	25	24
			CMP5	CMP4	CMP3	CMP2	CMP1	CMP0
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
			CCBUFV5	CCBUFV4	CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUFV		PATTBUFV	CLIENT	DFS	UFS	IDX	STOP
Access	R/W		R/W	R	R/W	R/W	R	R
Reset	0		0	0	0	0	0	1

Bits 24, 25, 26, 27, 28, 29 – CMPx Channel x Compare Value [x=0..5]
This bit reflects the channel x output compare value.

Value	Description
0	Channel compare output value is 0.
1	Channel compare output value is 1.

Bits 16, 17, 18, 19, 20, 21 – CCBUFVx Channel x Compare or Capture Buffer Valid [x=0..5]
For a compare channel, this bit is set when a new value is written to the corresponding CCBUFx register. The bit is cleared either by writing a '1' to the corresponding location when CTRLB.LUPD is set or automatically on an UPDATE condition.
For a capture channel, the bit is set when a valid capture value is stored in the CCBUFx register. The bit is automatically cleared when the CCx register is read.

Bits 14, 15 – FAULTx Non-recoverable Fault x State [x=0..1]
This bit is set by hardware as soon as the non-recoverable Fault x condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding FAULTxIN status bit is low. When this bit is clear, the timer/counter restarts from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding FAULTx bit. For further details on timer/counter commands, refer to the available command description (CTRLBSET.CMD).

Bit 13 – FAULTB Recoverable Fault B State
This bit is set by hardware as soon as the recoverable Fault B condition occurs.

This bit can be cleared by hardware when Fault B action is resumed or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If the software halt command is enabled (FAULTB.HALT = SW), clearing this bit releases the timer/counter.

Bit 12 – FAULTA Recoverable Fault A State
This bit is set by hardware as soon as the recoverable Fault A condition occurs.
This bit can be cleared by hardware when Fault A action is resumed or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If the software halt command is enabled (FAULTA.HALT = SW), clearing this bit releases the timer/counter.

Bit 11 – FAULT1IN Non-Recoverable Fault 1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

Bit 10 – FAULT0IN Non-Recoverable Fault 0 Input
This bit is set while an active Non-Recoverable Fault 0 input is present.

Bit 9 – FAULTBIN Recoverable Fault B Input
This bit is set while an active Recoverable Fault B input is present.

Bit 8 – FAULTAIN Recoverable Fault A Input
This bit is set while an active Recoverable Fault A input is present.

Bit 7 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. This bit is automatically cleared by hardware on the UPDATE condition when CTRLB.LUPD is set or by writing a '1' to this bit.

Bit 5 – PATTBUFV Pattern Generator Value Buffer Valid
This bit is set when a new value is written to the PATTBUF register. This bit is automatically cleared by hardware on the UPDATE condition when CTRLB.LUPD is set or by writing a '1' to this bit.

Bit 4 – CLIENT Client Mode Enable
This bit is set when TCC is set in Client mode. This bit follows the CTRLA.MSYNC bit state.

Bit 3 – DFS Debug Fault State
This bit is set by hardware in Debug mode when the DDBGCTRL.FDDBD bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in Debug mode.
When the bit is set, the counter is halted and the Waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 2 – UFS Non-recoverable Update Fault State
This bit is set by hardware when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD). The bit is cleared by writing a one to this bit.
When the bit is set, the waveforms state depends on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 1 – IDX Ramp Index
In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads '0'. See *Ramp Operations* from Related Links.

Bit 0 – STOP Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when the One-Shot operation mode is enabled (CTRLBSET.ONESHOT = 1).
This bit is cleared on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

Related Links

41.6.3.4. Ramp Operations

41.8.14 Counter Value

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by the user writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD = READSYNC).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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 23:0 – COUNT[23:0] Counter Value
These bits hold the value of the Counter register.
Notes:

- When the TCC is configured as a 16-bit timer/counter, the excess bits are read '0'.
- This bit field occupies the MSBs of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0 (depicted)
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6

41.8.15 Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
			PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
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
			PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 8, 9, 10, 11, 12, 13 – PGVx Pattern Generation Output Value [x=0..5]
This register holds the values of pattern for each waveform output.

Bits 0, 1, 2, 3, 4, 5 – PGEx Pattern Generation Output Enable [x=0..5]
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' overrides the corresponding SWAP output with the corresponding PGVx value.

41.8.16 Waveform

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
						SWAP2	SWAP1	SWAP0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
			POL5	POL4	POL3	POL2	POL1	POL0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CICCEN3	CICCEN2	CICCEN1	CICCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bits 24, 25, 26 – SWAPx Swap DTI Output Pair x [x=0..2]
Setting these bits enables the output swap of DTI outputs [x] and [x+WO_NUM/2]. Note: the DTIxEN settings will not affect the swap operation.

Bits 16, 17, 18, 19, 20, 21 – POLx Channel Polarity x [x=0..5]
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

Value	Name	Description
0	(single-slope PWM waveform generation)	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value
1	(single-slope PWM waveform generation)	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value.
0	(dual-slope PWM waveform generation)	Compare output is set to ~DIR when TCC counter matches CCx value
1	(dual-slope PWM waveform generation)	Compare output is set to DIR when TCC counter matches CCx value.

Bits 8, 9, 10, 11 – CICCENx Circular CC Enable x [x=0..3]
Setting these bits enables the compare circular buffer option on the first four Compare/Capture channels. When the bit is set, the CCx register value is copied back into the CCBUFx register on the UPDATE condition.

Bit 7 – CIPEREN Circular Period Enable
Setting this bit enables the period circular buffer option. When the bit is set, the PER register value is copied back into the PERBUF register on the UPDATE condition.

Bits 5:4 – RAMP[1:0] Ramp Operation
These bits select the Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation

Bits 2:0 – WAVEGEN[2:0] Waveform Generation Operation

These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation must be used. These bits are not synchronized.

Value	Name	Description						
		Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVFIF/Event Up	Down
0x0	NFRQ	Normal Frequency	PER	TOP/Zero	Toggle	Stable	TOP	Zero
0x1	MFRQ	Match Frequency	CC0	TOP/Zero	Toggle	Stable	TOP	Zero
0x2	NPWM	Normal PWM	PER	TOP/Zero	Set	Clear	TOP	Zero
0x3	Reserved	—	—	—	—	—	—	—
0x4	DSCRITICAL	Dual-slope PWM	PER	Zero	~DIR	Stable	—	Zero
0x5	DSBOTTOM	Dual-slope PWM	PER	Zero	~DIR	Stable	—	Zero
0x6	DSBOTH	Dual-slope PWM	PER	TOP & Zero	~DIR	Stable	TOP	Zero
0x7	DSTOP	Dual-slope PWM	PER	Zero	~DIR	Stable	TOP	—

41.8.17 Period Value

Name: PER
Offset: 0x40
Reset: 0x00FFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PER[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[1:0]		DITHER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 23:6 – PER[17:0] Period Value

These bits hold the value of the TCC period count. The number of bits in this field corresponds to the size of the counter.

Note: When the TCC is configured as a 16-bit timer/counter, the excess bits are read as zero.

Note: This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6 (depicted)

Bits 5:0 – DITHER[5:0] Dithering Cycle Number

These bits hold the number of extra cycles that are added on the PWM pulse period every 64 PWM frames.

Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

41.8.18 Compare/Capture Channel x

Name: CCx
Offset: 0x44 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

The CCx register represents the 16-, 24- bit value, CCx. The register has two functions depending on the mode of operation.

For the capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For the compare operation, this register is continuously compared to the counter value. Normally, the output from the comparator is, then, used for generating waveforms.

The CCx register is updated with the buffer value from their corresponding CCBUFx register when an UPDATE condition occurs.

In addition, in the match frequency operation, the CC0 register controls the counter period.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CC[17:10]							
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
	CC[9:2]							
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
	CC[1:0]		DITHER[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 23:6 – CC[17:0] Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.
Notes:

1. When the TCC is configured as a 16-bit timer/counter, the excess bits are read as zero.
2. This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6 (depicted)

Bits 5:0 – DITHER[5:0] Dithering Cycle Number

These bits hold the number of extra cycles that are added on the PWM pulse width every 64 PWM frames.

Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

41.8.19 Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
			PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
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
			PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 8, 9, 10, 11, 12, 13 – PGVBx Pattern Generation Output Value Buffer [x=0..5]
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGVx register on an UPDATE condition.

Bits 0, 1, 2, 3, 4, 5 – PGEBx Pattern Generation Output Enable Buffer [x=0..5]
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGEx register at an UPDATE condition.

41.8.20 Period Buffer Value

Name: PERBUF
Offset: 0x6C
Reset: 0x00FFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PERBUF[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 23:6 – PERBUF[17:0] Period Buffer Value
These bits hold the value of the Period Buffer register. The value is copied to the PER register on the UPDATE condition.
Note: When the TCC is configured as a 16-bit timer/counter, the excess bits are read as '0'.

Note: This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enabled, the value of this bit field is copied to the PER.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	—
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

41.8.21 Channel x Compare/Capture Buffer Value

Name: CCBUFx
Offset: 0x70 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

CCBUFx is copied into CCx at TCC update time.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CCBUF[17:10]							
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
	CCBUF[9:2]							
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
	CCBUF[1:0]		DITHERBUF[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 23:6 – CCBUF[17:0] Channel x Compare/Capture Buffer Value [x=0..5]
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBUFVx status bit.
Notes:

1. When the TCC is configured as a 16-bit timer/counter, the excess bits are read as '0'.
2. This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enabled, the DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	—
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

42. Zigbee Bluetooth Radio Subsystem (ZBT)

42.1 Overview

The PIC32CX-BZ3 provides an on-chip IEEE 802.15.4-compliant Zigbee, Bluetooth Low Energy 5.2 interface with integrated transceivers. The Wireless Subsystem block comprises the following modules:

- Single ultra-low power 2.4 GHz ISM band RF transceiver
- Bluetooth modem
- Bluetooth link layer
- ZigBee modem
- ZigBee MAC

The radio arbiter hardware allows the establishment of Bluetooth or ZigBee links with programmable QoS. The RF transceiver includes a switching-power amplifier architecture and a transmit/receive switch. Therefore, medium to high power application use cases are supported without an external front-end module (FEM).

With integrated Ultra Low-Power 2.4 GHz ISM band single transceiver, the radio supports both Zigbee and Bluetooth 5.2 link protocols. An on-board intelligent Radio arbiter hardware allows the establishment of Bluetooth or ZigBee links with programmable QoS. The RF transceiver includes switching power amplifiers architecture and a TR switch. Therefore, medium to high power application use cases are supported without external FEM.

The RTOS running on the Cortex M4F CPU handles the arbitration between the application, the Bluetooth link stack, the ZigBee link stack and miscellaneous maintenance tasks.

42.2 Features

2.4 GHz RF Transceiver

- Integrated 2.4 GHz Ultra-Low Power RF Transceiver Shared Between Bluetooth and Zigbee Modems and Link (MAC) Controllers
- Integrated Crystal Oscillator with Support for 16 MHz Crystal
- Internal Parallel PA Paths, Which May Be Shut Down to Control Pout as well as Current Consumption to Improve TX Power Efficiency
- Low BOM Single-Ended TRX RFFE Architecture
  - Integrated balun (single ended RF output) and TRX switch
- Hardware Radio Arbiter with Programmable QoS:
  - Resolution: up to per packet level
  - Based on shared transceiver and antenna

Bluetooth

- Bluetooth Low Energy 5.2 Certified
- Up to +10 dBm Programmable Transmit Output Power
- Typical Receiver Power Sensitivity:
  - -97 dBm for Bluetooth Low Energy 1 Mbps
  - -95 dBm for Bluetooth Low Energy 2 Mbps
  - -108 dBm for Bluetooth Low Energy 125 Kbps
  - -102 dBm for Bluetooth Low Energy 500 Kbps
- Wideband RSSI

- Enables Interference Robustness and Higher Tolerance to Out-of-Band Blockers
- 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
- HCI Interface via UART
- LE Power Control
- Bluetooth Low Energy Profiles/Services:
  - Bluetooth Low Energy peripheral and central roles
  - Bluetooth Low Energy APIs for application layer to implement standard or customize GATT based profiles/ services
  - Alert notification service
  - Proximity reporter (PXP)
  - Time information

Zigbee

- PSDU Data Rate: 250 Kbps
- Programmable RX Mode
  - -103 dBm RX sensitivity (typical) in the Continuous mode
  - -96 dBm sensitivity in the RPC mode
  - RPC mode provides lower power consumption in the RX mode to support California Green Energy Specification at the system level
- TX Output Power Up to +10 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)
- Zigbee Stack Support
  - Zigbee 3.0 ready

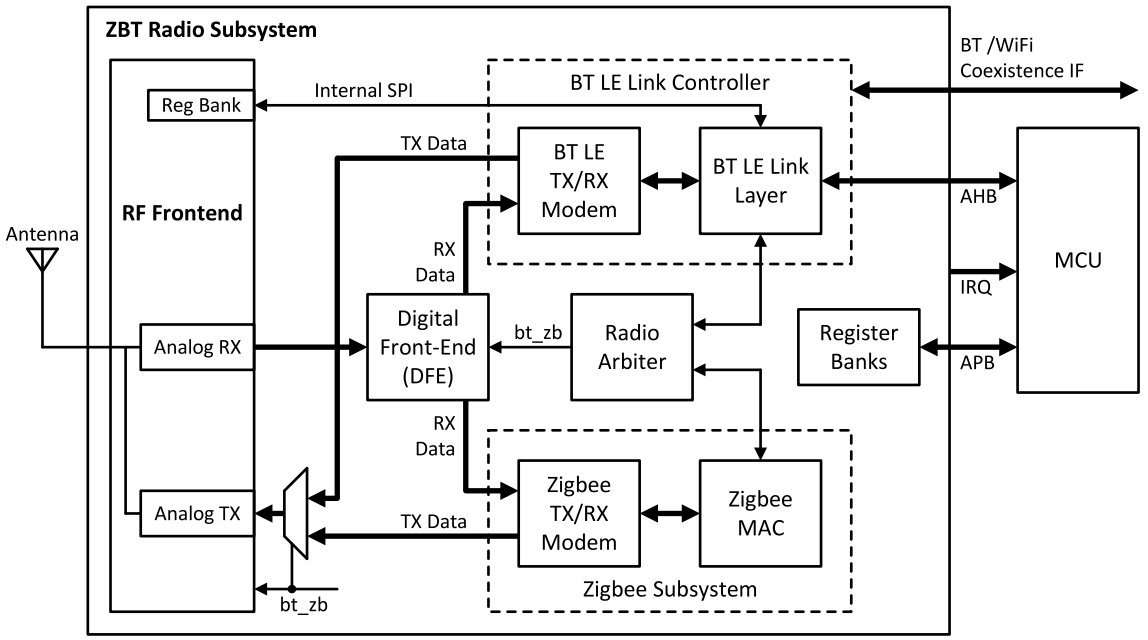
- Zigbee Pro 2017
- Zigbee green power support (proxy, sink and multi sensor)

Proprietary Modulation Schemes

- 500 kbps and 1 Mbps are 2.4 GHz Proprietary with DSSS
- 2 Mpbs are 2.4 GHz Proprietary Without DSSS
- Typical Sensitivity Level for the Proprietary Data Rates:
  - 500 Kbps: -98 dBm
  - 1 Mbps: -96 dBm
  - 2 Mbps: -90 dBm

42.3 Wireless Subsystem Top Level Diagram

Figure 42-1. Wireless Subsystem Top Level Diagram



42.4 Analog RF Front-End

The analog front-end contains the following:

- RF antenna switch
- XTO, connecting to a 16 MHz crystal oscillator
- RF synthesizer
- Analog RX path (LNA, Mixer, IF amplifier)
- ADC
- Analog TX path (Power amplifier)
- System clock PLL
- RF-related power management (RF-LDOs)
- Analog control logic

42.5 Digital Front-end

The Digital Front-End (DFE) connects to the analog front-end on one side and the Bluetooth or Zigbee baseband on the other side. In the receive path, the I/Q data from the ADC are filtered and down-converted. Automatic gain control is also implemented in the DFE. In the transmit path, the Zigbee power control is done in the DFE.

42.6 Bluetooth Low Energy Link Controller

The Bluetooth Low Energy link controller consists of the TX/RX modem and the link layer. The modem is responsible for modulation and demodulation of the digital IF data for both Bluetooth classic and Bluetooth Low Energy.

The baseband link layer carries out all Bluetooth operations as transmit or receive tasks. There are several task managers:

1. Firmware – Firmware can trigger tasks by writing the task controller registers.
2. Hardware schedule controller – This is Bluetooth Low Energy advertisement scheduler controller (Advertiser role).
3. Hardware scanner – This is Bluetooth Low Energy advertisement scanner (Central role).
4. Bluetooth Low Energy advertisement controller – This is Bluetooth Low Energy 4.2 advertisement controller (Advertiser role).

The requests from the task managers are arbitrated and carried out by the task controller.

The RX and TX data are written to common memory via DMA over an AHB interface.

42.7 Zigbee Subsystem

The Zigbee Subsystem deals with the physical/modem layer and the MAC layer of the Zigbee transceiver. A digital baseband processor down-converts and demodulates the received IF data. The subsequent modules implement basic MAC functionality and hardware accelerators for features, such as automatic acknowledgment, CSMA_CA and retransmission or automatic FCS check. The flow control is done by a Finite State Machine (FSM). The data is stored in integrated 128-byte RX and TX frame buffers.

42.8 Radio Arbiter

The Radio Arbiter provides a low-level arbitration for using the single RF frontend for both Zigbee and Bluetooth links. The arbiter supports the following modes:

- BT static – Radio ownership is with the Bluetooth link controller
- ZB static – Radio ownership is with the Zigbee subsystem

In the static modes, either the Bluetooth link controller or the Zigbee subsystem continuously owns the radio. No arbitration is done.

42.9 Register Banks

The ZBT subsystem contains various register banks to configure and control the digital baseband hardware and state machines. The firmware accesses the registers by the ARM APB bus.

42.10 Coexistence Interface

Bluetooth and Wi-Fi partly operate in the same 2.4 GHz band. Therefore, transmissions can interfere with each other and impact the performance and reliability of the wireless systems. The Coexistence interface is built from a 3-wire bus that mutually signalizes RF activity when Bluetooth and Wi-Fi chips are available in the same hardware module. Interference can, thus, be avoided.

43. Electrical Characteristics

This chapter provides the electrical specifications and characteristics of the PIC32CX-BZ3 and the WBZ35x Module across the operating temperature range of the device.

Note: All the electrical specifications of the PIC32CX-BZ3 apply to the WBZ35x Module as well unless specified explicitly.

43.1 Absolute Maximum Ratings

Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions above the parameters indicated in the operation listings of this specification is not implied.

Table 43-1. Absolute Maximum Ratings

Parameter	Value
Ambient temperature under bias (PIC32CX-BZ3)(1)	-40°C to +125°C
Ambient temperature under bias (WBZ35x)(1)	-40°C to +85°C
Storage temperature	-65°C to +150°C
Voltage on V_{DD}/V_{DDIO} with respect to GND	-0.3V to +4.0V
Voltage on V_{DDREG} with respect to GND	-0.3V to +1.4V
Voltage on V_{DDCORE} with respect to GND	-0.3V to +1.32V
Voltage on any digital I/O pin, with respect to GND(3)	-0.3V to (V_{DDIO} + 0.3V)
Maximum current out of GND pins	200 mA
Maximum current into V_{DD} pins(2)	200 mA
Maximum output current sunk by any I/O pin	10 mA
Maximum output current sourced by any I/O pin	15 mA
Maximum current sunk by all ports	120 mA
Maximum current sourced by all ports(2)	120 mA
Maximum current sunk/sourced by RF supply pin	20 mA
Maximum voltage on RF supply pins with respect to GND	1.4V
Maximum Junction Temperature	+135°C
ESD Qualification	
Human Body Model (HBM) per JESD22-A114	±2000V
Charged Device Model (CDM) (ANSI/ESD STM 5.3.1)...(All pins/Corner pins)	±500V

Notes:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
2. Maximum allowable current is a function of the device's maximum power dissipation (see the *Thermal Operating Conditions* table in the *Thermal Specifications* from Related Links).

Related Links

43.4. Thermal Specifications

43.2 Operating Conditions

Table 43-2. Operating Conditions

Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions(1)
V_{DD}	Voltage range of V_{DD}/V_{DDIO}	1.9	3.3	3.6	V	Operating range of 1.9-3.6V

.....continued						
Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions[1]
V_{DDREG} ($PMU_{O/P}$)	Regulator voltage range of V_{DDREG}	1.3	1.35	1.4	V	PMU output voltage
V_{DDCORE}	Core V_{DD} supply ($CLDO_{O/P}$)	1.11	1.2	1.32	V	CLDO output voltage
AV_{DD}	Analog V_{DD} supply	V_{DD}-0.3	—	V_{DD}+0.3	V	—
AV_{SS}	Analog V_{SS} supply	V_{SS}-0.3	—	V_{SS}+0.3	V	—
Note: The same voltage must be applied to V_{DD} and AV_{DD}.						

43.3 DC Electrical Characteristics

Table 43-3. Operating Frequency VS. Voltage

Param. No.	V_{DDIO}, V_{DDANA} Range	Temp. Range (in °C)	Max. MCU Frequency	Comments
DC_5	1.9V to 3.6V	-40°C to +85°C	64 MHz	Industrial
DC_7	1.9V to 3.6V	-40°C to +125°C	64 MHz	Extended
Note: The same voltage must be applied to V_{DD} and AV_{DD}.				

43.4 Thermal Specifications

Table 43-4. Thermal Operating Conditions

Rating	Symbol	Min.	Typ	Max.	Unit
Industrial Temperature Devices:					
Operating ambient temperature range	T_A	-40	—	+85	°C
Operating junction temperature range	T_J	-40	—	+95	°C
Extended Temperature Range:					
Operating ambient temperature range	T_A	-40	—	+125	°C
Operating junction temperature range	T_J	-40	—	+135	°C
Power Dissipation: Internal Chip Power Dissipation: P_{INT} = (VDDIOx x (IDD –∑ IOH)) I/O Pin Power Dissipation: $P_{I/O}$ = ∑ (({VDDIO – VOH} x IOH) + ∑ (VOL x IOL))	P_D	$P_{INT} + P_{I/O}$			W
Maximum allowed power dissipation	P_{DMAX}	$(T_J – T_A)/\Theta_{JA}$			W
Note: The WBZ35x Module supports -40°C to +85°C.					

Table 43-5. Thermal Packaging Characteristics

Characteristics	Symbol	Typ	Max.	Unit
Thermal Resistance, 48-pin VQFN (6 mm x 6 mm x 0.9 mm) Package	Θ_{JA}	<18	—	°C/W
Thermal Resistance, 32-pin VQFN (5 mm x 5 mm x 0.9 mm) Package	Θ_{JA}	<18	—	°C/W
Note: The junction-to-ambient thermal resistance, Θ_{JA}, numbers are achieved by package simulations.				

43.5 Power Supply DC Module Electrical Specifications

Table 43-6. Power Supply DC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_20	V_{DD}	Voltage range of V_{DD} (main power supply)	1.9	3.3	3.6	V	Operating range of 1.9-3.6V
REG_39	AV_{DD}	Analog V_{DD} supply input voltage range	V_{DD}-0.3	—	V_{DD}+0.3	V	Operating range
REG_21	AV_{SS}	Analog V_{SS} supply	V_{SS}-0.3	—	V_{SS}+0.3	V	GND
REG_36	V_{DDCORE}	V_{DDCORE} voltage range	—	1.2	—	V	CLDO output voltage
REG_37	V_{DDIO}	V_{DDIO} input voltage range	—	3.3	3.6	V	—
REG_38	RFLDO_OUT	RF LDO power supply	—	1.2	—	V	RF LDO output voltage
REG_40	V_{DDREG} (PMU_MLDO_OUT)	PMU output voltage - regulator voltage range	—	1.35	—	V	PMU output voltage
REG_40A	V_{PMU_VDD} (Buck mode)	PMU input supply voltage range	2.4	3.3	3.6	V	Input supply voltage range (Buck mode)
REG_40B	V_{DDREG_IN} (MLDO mode)	PMU input supply voltage range	1.9	3.3	3.6	V	Input supply voltage range (MLDO mode)

Notes:

- These parameters are characterized but not tested in manufacturing.
- V_{DD} and AV_{DD} must be at the same voltage level.
- For more details on power supply pins filtering, refer to the Design Package available on the product page.

Table 43-7. POR Electrical Characteristics

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DC16	VPOR	V_{DD} Start voltage to ensure internal Power-on Reset signal	1.52	—	1.58	V	V_{DD} voltage must remain at V_{SS} for a minimum of 200 µs to ensure POR. V_{DDIO} Power-up/Power-down (See V_{DDIO} Ramp Rate)
DC17	SVDD_R	V_{DDIO} Rise ramp rate to ensure internal Power-on Reset signal	0.03	—	0.115	V/ms	Failure to meet this specification may lead to start-up or unexpected behaviors
DC18	SVDD_F	VDDIO Falling ramp rate to ensure internal Power-on Reset signal	—	1.39	—	V/ms	Failure to meet this specification may cause the device to not detect reset

Note: These parameters are characterized but not tested in manufacturing.

Table 43-8. BOR Electrical Characteristics

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
B011	V_{BHYS}	Brown-out hysteresis	51	53.4	54	mV	—
B012	VZPBOR	Zero-power BOR	—	1.9	—	V	—
Note: These parameters are characterized but not tested in manufacturing.							

Table 43-9. Reset Timing Characteristics

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SY00	T_{PU}	Power-up period	—	400	600	µs	—
SY02	T_{SYSDLY}	System delay period before first instruction is fetched	—	1 µs + 8 SYSCLK cycles	—	—	—
SY20	$T_{\overline{MCLR}}$	$\overline{MCLR}$ pulse width (low)	2	—	—	µs	Minimum reset active time to guarantee MCU reset
SY30	T_{BOR}	BOR pulse width (low)	—	1	—	µs	CRU combines BOR12 and BOR33
Note: These parameters are characterized but not tested in manufacturing.							

43.6 Active Current Consumption DC Electrical Specifications

Table 43-10. Active Current Consumption DC Electrical Specifications

DC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ TA ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions
APWR_1	IDD_ACTIVE	MCU I_{DD} in Active mode w/LDO mode selected	PLL 64 MHz	9.9	—	mA	V_{DD} = 3.3V, T_A = 25°C

.....continued

DC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ TA ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions

Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (in other words, all peripherals inactive).
   - IOs are configured as Input & Pulled Up.
   - PMU 1 MHz Clock from FRC is divided by 8.
   - All PB Clocks are divided by 16.
   - Disabled JTAG_En and disconnected DSU internally.
   - Disable Prefetch Cache.
   - MCU is running on Flash with automatic wait state.
   - I/Os are inactive input mode with input trigger disabled.
   - All clock generation sources disabled unless otherwise specified.
   - WDT, RTCC, CFD Clock Fail Detect disabled.
   - µA/MHz varies over temperature. Worst case is given by max.
3. MCU running 50 NOPs in while loop.
4. These parameters are characterized but not tested in manufacturing.

43.7 Idle Current Consumption DC Electrical Specifications

Table 43-11. Idle Current Consumption DC Electrical Specifications

DC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions
IPWR_1	I_{DD_IDLE}	MCU I_{DD} in IDLE mode w/LDO mode selected	PLL 64 MHz	6.5	—	mA	V_{DD} = 3.3V, T_A = 25°C

.....continued

DC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions

Notes:

1. Typical value measured during characterization across voltage and temperature.
2. Conditions:
   - All GPIO are input and pulled up.
   - All peripherals are disabled with PMD bits and ON bits.
   - All PB clocks are divided by 16.
   - LPRC is set as LPCLK.
   - SOSC is disabled.
   - PMU 1 MHz clock from FRC is divided by 8.
   - Disable Prefetch Cache.
   - WCM memories are configured in Retention + NAP mode.
   - JTAG and DSU are disconnected.
   - Clock Gate Bluetooth® Zigbee® Subsystem.
   - Entry to Sleep mode is disabled and WFI instruction is executed.
   - On Exit by External interrupt, Verified RCON status to ensure System was in IDLE mode.
3. These parameters are characterized but not tested in manufacturing.

43.8 Sleep Current Consumption DC Electrical Specifications

Table 43-12. Sleep Current Consumption DC Electrical Specifications

DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	V_{DDIO}	Typ.	Max.	Units	Conditions
SPWR_1	I_{DD_SLEEP}	MCU I_{DD} in Sleep mode w/LDO mode selected	3.3V	0.55	19.25	mA	XTAL = OFF, T_A = 25°C
SPWR_5			3.3V	0.8	19.64	mA	XTAL = ON, T_A = 25°C

.....continued

DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	V_{DDIO}	Typ.	Max.	Units	Conditions

Notes:

1. Typical value measured during characterization across voltage and temperature.
2. Conditions:
   - All GPIO are input and pulled up.
   - All peripherals are disabled with PMD bits and ON bits.
   - All PB clocks are divided by 16.
   - LPRC are set as LPCLK.
   - SOSC is disabled.
   - PMU 1 MHz clock from FRC is divided by 8.
   - Cache enabled and configured wait time as 0xF.
   - WCM memories configured in Retention + NAP mode.
   - JTAG and DSU are disconnected.
   - RF powered down.
   - Xtal turned off or on based on the configuration.
   - Entry to Sleep mode enabled and WFI instruction executed.
   - On exit by external interrupt, verified RCON status to ensure system was in Sleep mode.
3. These parameters are characterized but not tested in manufacturing.

43.9 Deep Sleep Current Consumption DC Electrical Specifications

Table 43-13. Deep Sleep Current Consumption DC Electrical Specifications

DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	V_{DDIO}	Typ.	Max.	Units	Conditions
BPWR_1	I_{DD_BACKUP}	MCU I_{DD} in Deep Sleep mode powered from V_{DDIO}	3.6V	1.7	40	µA	16K Backup RAM
BPWR_9			3.6V	1.9	60	µA	32K Backup RAM

Notes:

1. Typical value measured during characterization across voltage and temperature.
2. Conditions:
   - All GPIO are input and pulled up.
   - All Peripherals are disabled with PMD bits and ON bits.
   - All PB clocks are divided by 16.
   - Configure LPRC or SOSC as LPCLK based on configuration.
   - POSC and SOSC are disabled when not in use.
   - Configure Flex RAM retention size as 16k or 32k based on configuration.
   - JTAG and DSU are disconnected.
   - RTCC and DSWDT are turned on.
   - Entry to Deep Sleep mode is enabled and WFI instruction is executed.
3. These parameters are characterized but not tested in manufacturing.

43.10 XDS (Extreme Deep Sleep) Current Consumption DC Electrical Specifications

Table 43-14. XDS (Extreme Deep Sleep) Current Consumption DC Electrical Specifications

DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	V_{DDIO}	Typ.	Max.	Units	Conditions
OPWR_1	I_{DD_OFF}	MCU I_{DD} in XDS mode powered from V_{DDIO}	3.3V	0.09	—	µA	V_{DD} = 3.3 V, T_A = 25°C In OFF mode, the device is entirely powered-off. (SLEEPCFG.SLEEPMODE = OFF), and subsequent execution of the WFI instruction. **Note:** This mode is left by pulling the RESET pin low, or when a power Reset is done.

Notes:

1. Typical value measured during characterization across voltage and temperature.
2. Conditions:
   - All GPIO are input and pulled up.
   - All peripherals are disabled with PMD bits and ON bits.
   - All PB clocks are divided by 16.
   - Configure LPRC as LPCLK.
   - POSC and SOSC are disabled.
   - WCM memory powered off.
   - JTAG and DSU are disconnected.
   - RTCC and DSWDT are turned off.
   - Entry to Deep Sleep mode is enabled and WFI instruction is executed.
3. These parameters are characterized but not tested in manufacturing.

43.11 Wake-Up Timing from Low Power Modes AC Electrical Specifications

Table 43-15. Wake-Up Timing from Low Power Modes AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
WKUP_1	WKUP_IDLE	Wake from IDLE mode	—	7.03	—	µs	—
WKUP_3	WKUP_STDBY	Wake from STANDBY/Sleep mode	—	14.23	—	µs	STDBYCFG.FASTWKUP = 0
WKUP_13	WKUP_BCKUP	Wake from BACKUP/Deep Sleep mode	—	19.827	—	ms	—
WKUP_15	WKUP_OFF	Wake from OFF mode	—	19.84	—	ms	—

Note: These parameters are characterized but not tested in manufacturing.

43.12 I/O PIN AC/DC Electrical Specifications

Table 43-16. I/O PIN AC/DC Electrical Specifications

AC - DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DI_1	V_{IL}	Input low voltage I/O pins (Drive strength, 8x)	V_{SS}	—	0.2*V_{DD}	V	—
		Input low voltage I/O pins (Drive strength, 4x)		—	0.2*V_{DD}		—
DI_3	V_{IH}	Input high voltage, I/O pins (Drive strength, 8x)	—	—	V_{DD}	V	—
		Input high voltage, I/O pins (Drive strength, 4x)	—	—	V_{DD}	V	—
DI_5	V_{OL}	4x Drive strength I/O pins (Output low)	—	—	0.4	V	—
		8x Drive strength I/O pins (Output low)	—	—	0.4		—
		12x Drive strength I/O pins (Output low)	—	—	0.4		—
DI_9	V_{OH}	4x Drive strength I/O pins (Output high)	2.4	—	—	V	—
		8x Drive strength I/O pins (Output high)	2.4	—	—		—
		12x Drive strength I/O pins (Output high)	2.4	—	—		—
DI_13	I_{IL}	Input pin leakage current	-1	—	+1	µA	GND ≤ V_{PIN} ≤ $V_{DDIO(max)}$ (V_{PIN} = Voltage present on pin)
DI_15	R_{PDWN}	Internal pull-down resistance	—	13	—	kΩ	$V_{DDIO(min)}$ to $V_{DDIO(max)}$
DI_17	R_{PUP}	Internal pull-up resistance	—	13	—	kΩ	
DI_19	I_{ICL}	Input low injection current	0	—	-5	mA	This parameter applies to all I/O pins except VDD, VSS, AVDD,AVSS, $\overline{MCLR}$[1,3,4]
DI_21	I_{ICH}	Input high injection current	0	—	+5	mA	This parameter applies to all pins, except V_{DD}, V_{SS}, AV_{DD},AV_{SS}, $\overline{MCLR}$

.....continued

AC - DC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DI_25	T_{RISE}	I/O pin rise time (Drive strength, 4x)	—	—	9.5	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin rise time (Drive strength, 4x)	—	—	6	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf
		I/O pin rise time (Drive strength, 8x)	—	—	8	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin rise time (Drive strength, 8x)	—	—	6	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf
		I/O pin rise time (Drive strength, 12x)	—	—	3.5	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin rise time (Drive strength, 12x)	—	—	2	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf
DI_27	T_{FALL}	I/O pin fall time (Drive strength, 4x)	—	—	9.5	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin fall time (Drive strength, 4x)	—	—	7.5	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf
		I/O pin fall time (Drive strength, 8x)	—	—	8	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin fall time (Drive strength, 8x)	—	—	7.5	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf
		I/O pin fall time (Drive strength, 12x)	—	—	4	ns	V_{DDIO} = 3.3V, C_{LOAD} = 50 pf
		I/O pin fall time (Drive strength, 12x)	—	—	3.1	ns	V_{DDIO} = 3.3V, C_{LOAD} = 20 pf

Notes:

1. V_{IL} source < (GND – 0.3). Characterized but not tested in manufacturing.
2. V_{IH} source > (V_{DDIO} + 0.3). Characterized but not tested in manufacturing.
3. If the sum of all injection currents are > | ∑IICT |, it can affect the ADC results by approximately 4 to 6 counts (in other words, V_{IH} Source > (V_{DDIO} + 0.3) or V_{IL} source < (GND – 0.3)).
4. Any number and the combination of I/O pins not excluded under IICL or IICH conditions are permitted provided the absolute instantaneous sum of the input injection currents from all pins do not exceed the specified ∑IICT limit. To limit the injection current, the user must insert a resistor in series RSERIES (RS), between the input source voltage and device pin. The resistor value is calculated according to:
   - For negative input voltages less than (GND – 0.3): RS ≥ absolute value of | ((V_{IL} source – (GND – 0.3))/IICL) |
   - For positive input voltages greater than (V_{DDIO} + 0.3): RS ≥ ((V_{IH} source – (V_{DDIO} +0.3))/IICH)
   - For V_{PIN} voltages greater than V_{DDIO} + 0.3 and less than GND – 0.3: RS = the larger of the values calculated above

43.13 External XTAL and Clock AC Electrical Specifications

Table 43-17. External XTAL and Clock AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions(1)
XOSC_1	F_{OSC_XOSC}	External CLKI frequency	—	16	—	MHz	XIN, XOUT Primary Osc EC (±20 ppm)

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions(1)
XOSC_1A	TOSC	TOSC = 1/ FOSC_XOSC	—	0.0625	—	ns	See parameter XOSC_1 for FOSC_XOSC value
XOSC_2	XOSC_ST(3)	XOSC crystal start-up time	—	—	2.5	ms	Crystal stabilization time only not oscillator ready
XOSC_3	C_{XIN}	XOSC X_{IN} parasitic pin capacitance	—	0.35	—	pF	With default crystal trim settings
XOSC_5	C_{XOUT}	XOSC X_{OUT} parasitic pin capacitance	—	0.35	—	pF	With default crystal trim settings
XOSC_11	C_{LOAD}(4)	XOSC crystal FOSC = 16 MHz	—	9	—	pF	—
XOSC_21	ESR	XOSC crystal FOSC = 16 MHz	—	100	—	Ω	—
XOSC_33	D_{LEVEL}	MCU crystal oscillator power drive level	—	100	—	µW	—
XOSC_34	Gm	XOSC Transconductance	14	16	18	mA/V	XOSC Auto gain control disabled V_{DD} = 1.2V, T_A = +25°C

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions(1)

Notes:

1. V_{DDIO} = AV_{DD} = 3.3V.
2. The parameters are characterized but not tested in manufacturing.
3. This is for guidance only. A major component of crystal start-up time is based on the 2nd party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern, the customer must characterize this based on their design choices.
4. The test conditions for the crystal load capacitor calculation are as follows:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (in other words, PCB STD TRACE W = 0.175 mm, H = 36 µm, T = 113 µm).
   - Xtal PCB capacitance typical; therefore, ~= 2.5 pF for a tight PCB xtal layout
   - For CXIN and CXOUT within 4 pF of each other, assume CXTAL_EFF = ((CXIN+CXOUT)/2).
   - **Note:** Averaging CXIN and CXOUT will affect the final calculated CLOAD value by less than 0.25 pF.

   Equation 43-1. Equation 1:

   $$MFG\ \ CLOAD\ Spec = \{([CXIN + C1] * [CXOUT + C2])/[CXIN + C1 + C2 + CXOUT]\} + estimated\ oscillator\ PCB\ stray\ capacitance$$

   Assuming C1 = C2 and CXin ~= CXout, the formula can be further simplified and restated to solve for C1 and C2 by:

   Equation 43-2. Equation 2 (In other words: Simplified Equation 1)

   $$C1 = C2 = ((2 * MFG\ \ CLOAD\ Spec) - CXTAL\_EFF - (2 * PCB\ \ capacitance))$$

   Example:
   - XTAL Mfg CLOAD Data Sheet Spec = 12 pF
   - PCB XTAL trace Capacitance = 2.5 pF
   - CXIN pin = 6.5 pF, CXOUT pin = 4.5 pF; therefore, CXTAL_EFF = ((CXIN+CXOUT) / 2)

   CXTAL_EFF = ((6.5 + 4.5)/2) = 5.5 pF

   C1 = C2 = ((2 * MFG Cload spec) – CXTAL_EFF – (2 * PCB capacitance))

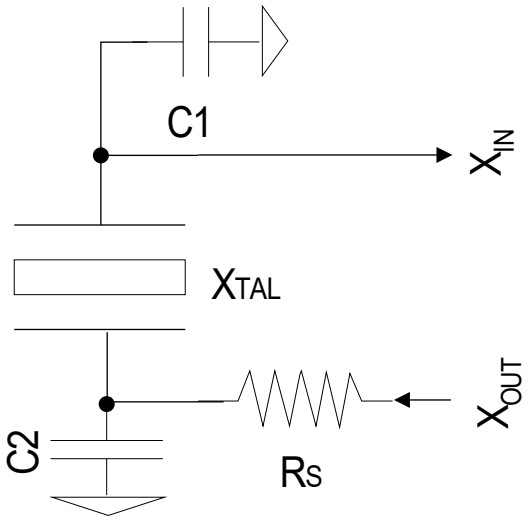
   C1 = C2 = (24 - 5.5 – (2 * 2.5))

   C1 = C2 = 13.5 pF (Always rounded down)

   C1 = C2 = 13 pF (in other words, for hypothetical example crystal external load capacitors)

   User C1 = C2 = 13 pF CLOAD (max) spec
5. The maximum start-up time is user-selectable in XOSCCTRL.STARTUP.

Figure 43-1. External XTAL and Clock Diagram



43.14 XOSC32 AC Electrical Specifications

Table 43-18. XOSC32 AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions[1]
XOSC32_1	F_{OSC}_XOSC32	XOSC32 oscillator crystal frequency	—	32.764	—	kHz	XIN32, XOUT32 secondary oscillator
XOSC32_3	C_{XIN32}	XOSC32 XIN32 parasitic pin capacitance	—	0.4 2.4	—	pF	0.4 pF at the SOC pins and 2.4 pF on the WBZ35x Module
XOSC32_5	C_{XOUT32}	XOSC32 XOUT32 parasitic pin capacitance	—	0.4 2.4	—	pF	0.4 pF at the SOC pins and 2.4 pF on the WBZ35x Module
XOSC32_11	C_{LOAD}_X32[4]	32.768 kHz crystal load capacitance	—	11	—	pF	—
XOSC32_13	ESR_X32	32.768 kHz crystal ESR	—	75	100	kΩ	—
XOSC32_15	TOSC32	TOSC32 = 1/ FOSC_XOSC32	—	30.5	—	µs	See parameter XOSC32_1 for FOSC_XOSC32 value
XOSC32_17	XOSC32_ST[3]	XOSC32 crystal start-up time	—	1024	—	TOSC	Crystal stabilization time only not oscillator ready

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions(1)

Notes:

1. V_{DDIO} = AV_{DD} = 3.3V.
2. The parameters are characterized but not tested in manufacturing.
3. This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitic that is outside the scope of this specification. If this is a major concern, the customer might need to characterize this based on their design choices.
4. The test conditions for the crystal load capacitor calculation are as follows:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (in other words, PCB STD TRACE W = 0.175 mm, H = 36 µm, T = 113 µm)
   - Xtal PCB capacitance typical; therefore, ~= 2.5 pF for a tight PCB xtal layout
   - For CXIN and CXOUT within 4 pF of each other, assume CXTAL_EFF = ((CXIN / 2)
   - **Note:** Averaging CXIN and CXOUT will affect the final calculated CLOAD value by less than the tolerance of the capacitor selection.

   Equation 1:

   $$MFG\ \ CLOAD\ Spec\ = \{([CXIN\ +\ C1]\ *\ [CXOUT\ \ +\ C2])/[CXIN\ +\ C1\ +C2\ +CXOUT]\}\ +\ estimated\ oscillator\ PCB\ stray\ capacitance$$

   Assuming C1 = C2 and CXin ~= CXout, the formula can be further simplified and restated to solve for C1 and C2 by:

   Equation 43-3. Equation 2 (In other words: Simplified Equation 1)

   $$C1\ =\ C2\ =\ ((2\ *\ MFG\ \ CLOAD\ Spec)\ -CXTAL\_EFF\ -\ (2\ *\ \ PCB\ \ \ capacitance))$$

   Example:
   - XTAL Mfg CLOAD Data Sheet Spec = 12 pF
   - PCB XTAL trace Capacitance = 2.5 pF
   - CXIN pin = 6.5 pF, CXOUT pin = 4.5 pF therefore CXTAL_EFF = ((CXIN / 2)

   CXTAL_EFF = ((6.5 + 4.5)/2) = 5.5 pF

   C1 = C2 = ((2 * MFG Cload spec) – CXTAL_EFF – (2 * PCB capacitance))

   C1 = C2 = (24 – 5.5 – (2 * 2.5))

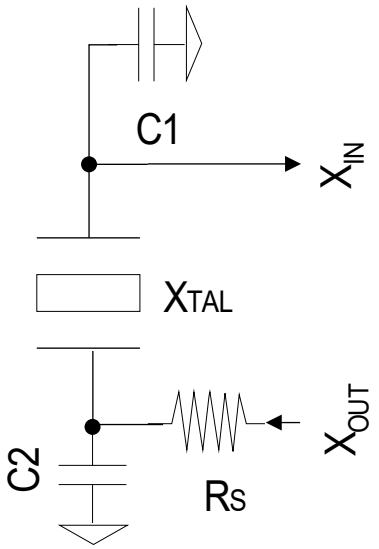
   C1 = C2 = (24 – 5.5 – 5)

   C1 = C2 = 13.5 pF (Always rounded down)

   C1 = C2 = 13 pF (in other words, for hypothetical example crystal external load capacitors)

   User C1 = C2 = 13 pF ≤ CLOAD_X32 (max.) spec
5. User selectable in OSC32KCTRL.STARTUP.

Figure 43-2. XOSC32 Block Diagram



43.15 Low Power Internal 32 kHz RC Oscillator AC Electrical Specifications

Table 43-19. Low Power Internal 32 kHz RC Oscillator AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
LP32K_1 (F_{INTRC})	FOSC_LPRC32K	Output frequency	32.276	32.768	33.26	kHz	V_{DDIO} = V_{DDANA} ≥ $V_{DDANA(min)}$ -40°C ≤ T_A ≤ +125°C (in other words, Factory default calibration) Calibrated range, across voltage and at 25°C, (± 1.0%)
LP32K_2	T_{INTRC}	LPRC period	—	1/F_{INTRC}	—	µs	1/F_{INTRC}
LP32K_6	T_{SURC}	LPRC Start-up time	—	—	300	µs	Non LP mode
LP32K_9	LPRC32K_Duty	LPRC32K OSC duty cycle	—	50	—	%	V_{DDIO} = V_{DDANA} ≥ $V_{DDANA(min)}$
Note: The parameters are characterized but not tested in manufacturing.							

43.16 DAC Module Electrical Specifications

Table 43-20. DAC Module Electrical Specifications

DC Characteristics(7)			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DAC_1	D_{RES}	DAC resolution	—	—	7	Bits	—
DAC_3	D_{CLK}	Internal DAC clock frequency (GCLK_DAC)	—	—	—	MHz	V_{DDANA}(min)

.....continued

DC Characteristics[7]				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DAC_5	D_{SAMP}	DAC sampling rate	Low power	—	—	—	Msps	±4 LSB of final value for step size ≤ 100 LSB at C_{LOAD} and R_{LOAD} w/V_{DDANA} = 3.3V
			High power	—	—	1	Msps	
DAC_7	V_{OUT}	Output voltage range		GNDANA+___	—	VDDANA-___	V	External pin (Buffered) V_{REF} = V_{DDANA} at C_{LOAD} and R_{LOAD}
				GNDANA+___	—	VREF-___	V	External pin (Buffered) at C_{LOAD} and R_{LOAD} (V_{REF} < (V_{DDANA}-150 mv))
				GNDANA	—	VREF-___	V	Internal (No buffer)
DAC_9	V_{REF}[1,2,3]	DAC reference input option	REFSEL = External	$V_{DDANA(min)}$ or ≥ 2.4V whichever is greater	—	V_{DDANA}	V	External reference CTRLB.REFSEL[1:0] = 0x2, V_{REFAB} ≤ V_{DDANA} and V_{REF} bypass Cap = 0.01 µf
				$V_{DDANA(min)}$ or ≥ 2.4V whichever is greater	—	V_{DDANA}	V	External reference CTRLB.REFSEL[1:0] = 0x0, V_{REFAU} ≤ V_{DDANA} and V_{REF} bypass Cap = 0.01 µf
			REFSEL = Internal	$V_{DDANA(min)}$ or ≥ 2.4V whichever is greater	—	V_{DDANA}		V_{DDANA} ≥ 2.4V
DAC_11	C_{LOAD}	DAC Out max load to meet VOUT and TSET		—	—	40	pf	For buffered output
DAC_13	R_{LOAD}	DAC Out max load to meet VOUT and TSET		33	—	—	KΩ	Minimum of 33K Ω resistance needed for buffered output path
DAC_15	Tset	DAC settling time		—	4	—	µs	±4 LSB of final value for step size ≤ 100 LSB at C_{LOAD} and R_{LOAD} w/ V_{DDANA} = 3.3V
DAC_17	Tset_FS	DAC full scale settling time		—	10	—	µs	±4 LSB of final value for step size from 10% to 90% at C_{LOAD} and R_{LOAD} w/ V_{DDANA} = 3.3V
Single Ended Mode[1,2,3,5]								

.....continued

DC Characteristics[7]				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
SDAC_19	INL[6]	Integral non linearity		-3	0	2	LSB	V_{REF} Internal = V_{DDANA} = 3.3V w/ C_{LOAD} and R_{LOAD}
SDAC_21	DNL[6]	Differential non linearity		-2	1	2	LSB	
SDAC_23	GERR[6]	Gain error	REFSEL = V_{DDANA}	-1.56	—	-0.119	LSB	
SDAC_25	EOFF[6]	Offset error	REFSEL = V_{DDANA}	-1.111	—	0.485	LSB	V_{REF} Internal = V_{DDANA} = 3.3V w/ C_{LOAD} and R_{LOAD}

Notes:

1. DAC internal bandgap reference voltage 2.4V ≤ REFSEL ≤ V_{DDANA}.
2. DAC reference voltages < 2.4V are not practical, and peripheral performance is not ensured.
3. DAC functional device operation with either internal or external V_{REF} < 2.4V is ensured but not characterized. DAC_ will function but with degraded performance. DAC accuracy is limited by user's application noise/accuracy on V_{DDANA}, GNDANA and V_{REF} accuracy/drift.
4. Value taken over 7 harmonics.
5. 12-bit mode
6. Over V_{OUT} range defined by the DAC_7 parameter
7. These parameters are characterized but not tested in manufacturing.

43.17 ADC Electrical Specifications

Table 43-21. ADC AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
Device Supply							
ADC_1	AV_{DD}	ADC module supply	$AV_{DD(min)}$	—	$AV_{DD(max)}$	V	—
Reference Inputs							
ADC_2	V_{REF}[4]	Absolute reference voltage (V_{REFH}-V_{REFL}) [4]	AV_{SS}-0.3	—	AV_{DD}+0.3	V	—
ADC_3	V_{REF}	Reference voltage high (external reference buffers)	AV_{SS}+1.8	—	AV_{DD}	V	—
ADC_4	V_{REFL}	Reference voltage low (external reference buffers)	AV_{SS}	—	AV_{DD}-1.8	V	—
ADC_4a	R_{REF}	Suggested V_{REF} impedance	—	—	25	Ω	Resistance from source to the V_{REFP}/V_{REFM} input, including RSource, PCB trace, pads and on-chip routing
ADC_4b	V_{CM}	Analog input Common mode voltage	AV_{SS} + V_{REF}/2	—	AV_{DD} - V_{REF}/2	V	—

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC_4c	V_{DIFF}	Differential analog input voltage (seldiff = 1) (VINP - VINN)	2*(V_{REFH} - V_{REFL})			V	—
ADC_4d	VSING	Single-ended analog input voltage (seldiff = 0) (VINP - VINN)	V_{REFL}	—	V_{REFH}	—	—
ADC_4e	RIN_INT	ADC internal resistance	—	—	200	Ω	Internal sampling switch resistance
ADC_4f	RIN_SYS	External input resistance to meet the maximum speed	—	—	125	Ω	Resistance from VINP/VINM to signal source. Include RSource + RPad + RPCB + Rroute
ADC_4g	RSRC	Maximum source impedance to meet 2 MSPS at 12-bit	—	—	500	Ω	—
		Maximum source impedance to meet 1 MSPS at 12-bit	—	—	1400	Ω	—
Analog Input Range							
ADC_7	A_{FS}	Full-scale analog input signal range (Single-ended)	AV_{SS}	—	V_{REF}	V	V_{REF} = $AV_{DD(max)}$
ADC_11	$T_{SETTLING}$	ADC stabilization time	—	10	—	µs	—

Table 43-22. ADC Single-Ended Mode AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
Single-Ended Mode ADC Accuracy							
SADC_11	Res	Resolution	6	—	12	bits	Selectable 8-, 10-, 12-bit resolution ranges
SADC_13	EN0B(3)	Effective number of bits	6.3	—	—	bits	2 Msps, Internal V_{REF}, AV_{DD} = V_{DDIO} = 3.3V
SADC_19	INL(3)	Integral non linearity	-13.737	—	2.869	LSb	2 Msps, Internal V_{REF}, AV_{DD} = V_{DDIO} = 3.3V
SADC_25	DNL(3)	Differential non linearity	-1.628	—	1.736	LSb	2 Msps, Internal V_{REF}, AV_{DD} = V_{DDIO} = 3.3V
SADC_31	GERR(3)	Gain error	-8.198	—	-4.697	LSb	2 Msps, Internal V_{REF}, AV_{DD} = V_{DDIO} = 3.3V
SADC_37	EOFF(3)	Offset error	4.905	—	25.094	LSb	2 Msps, Internal V_{REF}, AV_{DD} = V_{DDIO} = 3.3V
Single-Ended Mode ADC Dynamic Performance							

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
SADC_49	SINAD(1,2,3)	Signal to noise and distortion	39.728	—	—	dB	V_{REF} = AV_{DD} = V_{DDIO} = 3.3V at 12-bit resolution, max sampling rate(1,2)
SADC_51	SNR(1,2,3)	Signal to noise ratio	39.747	—	—		
SADC_53	SFDR(1,2,3)	Spurious free dynamic range	61.32	—	—		
SADC_55	THD(1,2,3)	Total harmonic distortion	—	—	-59.346		

Notes:

1. Characterized with an analog input sine wave = (FTP(max)/100). Example: FTP(max) = 1 Msps/100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12-bit resolution.
3. Spec values collected under the following additional conditions:
   a. At least (3) SERCOM, (2) TCC and (2) TC peripheral clocks active but the same peripherals disabled, not running.
   b. At least (6) I/O pins toggling simultaneously at > 6 MHz, not adjacent to analog input pin; (3) with external 2 ma pull-up loads & (3) with external 2 ma pull-down loads on the side of the package shared by V_{DDANA}.
   c. 12-bit resolution mode.
4. ADC functional device operation with either internal or external V_{REF} < 2.4V is functional but not characterized. ADC will function, but with degraded accuracy of approximately ~((0.06 * 2n)/V_{REF}) LSBs over full scale range, where “n”=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus user's application noise/accuracy on V_{DDANA}, GNDANA.
5. Value taken over 7 harmonics.

Table 43-23. ADC Conversion AC Electrical Requirements

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC_ Clock Requirements							
ADC_57	TAD	ADC clock period	—	20.8	—	ns	V_{REF} = AV_{DD} = 3.3V
ADC Single-Ended Throughput Rates							
ADC_59	FTPR (Single-ended mode)	Throughput rate(4) (Single-ended)	0.01	—	2	Msps	12-bit resolution, DIV_SHR = 2

Notes:

1. ADC_ Sample time = ((SAMPCTRL.SAMPLEN + 1) * TAD) and SAMPCTRL.OFFCOMP = 0.
2. ADC_ HDW forces sample time to 4*TAD when SAMPCTRL.OFFCOMP = 1; user SAMPCTRL.SAMPLEN is ignored.
3. ADC Throughput Rate FTP = ((1/((TSAMP + TCNV) * TAD))/(# of user active analog inputs in use on specific target ADC module)).
   Note: Specification values assume only one AINx channel in use.
4. SAMPCTRL.R2R = 1. (Must be set to ‘1’ in ADC differential mode).

Table 43-24. ADC Sample AC Electrical Requirements

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC_63	TSAMP	ADC sample time (1,2,3,5)	1(1,5)	—	—	TAD	12-bit TAD(min), Ext Analog Input Rsource ≤ 147Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 504Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 1,000Ω
			2(1,5)	—	—		12-bit TAD(min), Ext Analog Input Rsource ≤ 2,272Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 3,008Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 4,000Ω
			3(1,5)	—	—		12-bit TAD(min), Ext Analog Input Rsource ≤ 4,416Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 5,504Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 6,976Ω
			4(1,2,5)	—	—		12-bit TAD(min), Ext Analog Input Rsource ≤ 6,560Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 8,000Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 9,984Ω
			5(1,5)	—	—		12-bit TAD(min), Ext Analog Input Rsource ≤ 8,704 Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 10,496Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 12,992Ω
			6(1,5,6)	—	—		12-bit TAD(min), Ext Analog Input Rsource ≤ 10,880Ω
							10-bit TAD(min), Ext Analog Input Rsource ≤ 12,992Ω
							8-bit TAD(min), Ext Analog Input Rsource ≤ 16,000Ω
			—	—	—	ns	With DAC as input
			—	—	—		With temperature sensor as input
ADC_65	TCNV	Conversion time (3) (Single-Ended mode)	12			TAD	12-bit resolution
			11				10-bit resolution
			9				8-bit resolution

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. When SAMPCTRL.OFFCOMP = 0:
   - TSAMP = (((RSAMPLE + RSOURCE) * CSAMPLE * (#Bits Resolution + 2) * ln(2))/TAD)+1 rounded down to nearest whole integer
   - User SAMPCTRL.SAMPLEN = (TSAMP – 1)
2. When SAMPCTRL.OFFCOMP = 1:
   - TSAMP = 4 (Forced by HDW)
   - User SAMPCTRL.SAMPLEN = (n/a, Ignored by HDW)
3. ADC Throughput Rate FTP = ((1/((TSAMP + TCNV) * TAD))/(# of user active analog inputs in use on specific target ADC module)).
   Note: Specification values assume only one AINx channel in use.
4. SAMPCTRL.R2R = 1. (Must be set to '1' in ADC differential mode).
5. TSAMP ≥ (INT[((RSAMPLE + RSOURCE) * CSAMPLE * (#Bits Resolution+2) * ln(2))/TAD]+1).
6. For RSOURCE values exceeding TSAMP = 6 sample time condition, use the formula in note 5 to calculate.

43.18 Comparator AC Electrical Specifications

Table 43-25. Comparator AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp Operating Conditions: Vcm = Vdd/2, Vout ~ Vdd/2, CL = 25 pf, RL = 8k, unless other wise stated				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions
CMP_1	VIOFF_00	Input offset voltage	-15	—	13	mV	Comparator ref voltage = $AV_{DD}/2$
CMP_3	VICM	Input Common mode voltage	0	—	AV_{DD}	V	AV_{DD} = $V_{DDIO\ (min\text{-}to\text{-}max)}$
CMP_4	VIN	Input voltage range	AV_{SS}	—	AV_{DD}	mV	With respect to GND and AV_{DD}
CMP_15	$TRESP_{SS}$	Small signal response time	—	100	160	ns	Comparator ref voltage = $AV_{DD}/2$, input step of 50 mV with Input overdrive = ± 15 mV
CMP_19	C_{OUTVAL}	Comparator enabled to output valid	—	—	10	ms	Comparator module is configured before enabling it
CMP_23	$C_{VREFRNG}$	Comparator voltage reference input range	AV_{SS}(3)	—	AV_{DD}(3)	V	See Note 3
CMP_25	F_{GCLK_AC}	Analog comparator peripheral module clock freq	—	—	64	MHz	V_{DDANA} = $V_{DDIO\ (min\text{-}to\text{-}max)}$

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp Operating Conditions: Vcm = Vdd/2, Vout ~ Vdd/2, CL = 25 pf, RL = 8k, unless other wise stated				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions

Notes:

1. These parameters are characterized but not tested in manufacturing.
2. Values in typical column area taken at 25°C.
3. Comparator Ref voltage cannot exceed: (VIN(max) – VIOFF(max) – CMP_5(max) – 50 mV) ≥ CMP VREF ≥ (VIN(min) + VIOFF(min) + CMP_5(max) + 50 mV).

43.19 SPI Module Electrical Specifications

Table 43-26. SPI Module Host Mode Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
MSP_1	TSCK	SCK period	—	—	62.5	ns	16 MHz SCK on fixed pins
			—	—	93.75		10.6 MHz SCK on remappable pins
MSP_3	TSCL	SCK output low time	19	—	—	ns	—
MSP_5	TSCH	SCK output high time	22	—	—	ns	—
MSP_7	TSCF	SCK and MOSI output fall time	—	—	12	ns	See I/O Pin characteristics DI27
MSP_9	TSCR	SCK and MOSI output rise time	—	—	13	ns	See I/O Pin characteristics DI25
MSP_11	TMOV	MOSI data output valid after SCK	—	-9	—	ns	V_{DDIO}(min), CLOAD = 30 pf(MAX)
MSP_13	TMOH	MOSI hold after SCK	8.8	—	—	ns	
MSP_14	TMOS	MOSI set-up SCK	TSCK/2 - 5	—	—	ns	
MSP_15	TMIS	MISO set-up time of data input to SCK	23	—	—	ns	
MSP_17	TMIH	MISO hold time of data input to SCK	20	—	—	ns	

Note: Assumes $V_{DDIO\ (min)}$ and 30 pF external load on all SPIx pins unless otherwise noted.

Figure 43-3. SPI Host Module CPHA = 0 Timing Diagrams

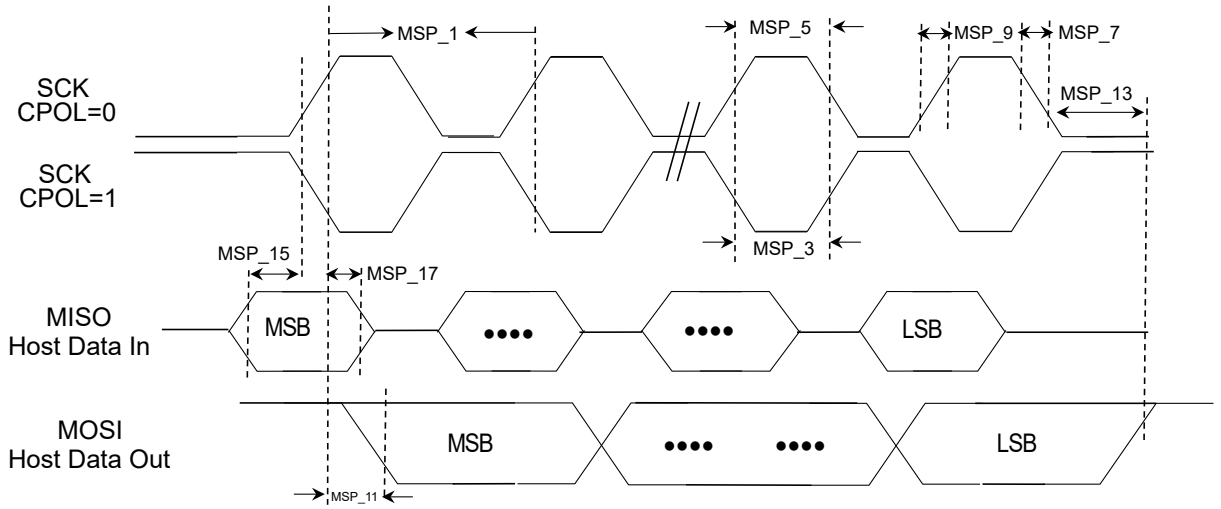


Figure 43-4. SPI Host Module CPHA = 1 Timing Diagrams

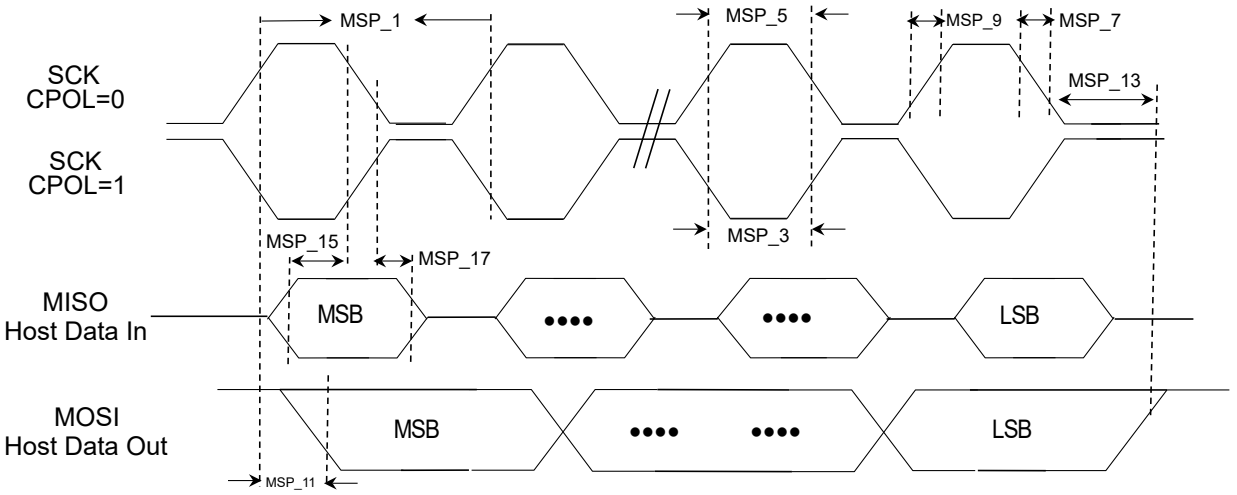


Table 43-27. SPI Module Client Mode Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SSP_1	FSCK	SCK frequency	—	—	16	MHz	16 MHz SCK on fixed pins
			—	—	12		12 MHz SCK on remappable pins
SSP_3	TSCL	SCK output low time	19	—	—	ns	—
SSP_5	TSCH	SCK output high time	29	—	—	ns	—
SSP_7	TSCF	SCK and MOSI output fall time	—	—	6	ns	See I/O Pin characteristics DI27
SSP_9	TSCR	SCK and MOSI output rise time	—	—	7	ns	See I/O Pin characteristics DI25

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SSP_11	TSOV	MOSI data output valid after SCK	—	—	—	ns	V_{DDIO} = 3.3V, C_{LOAD} = (31 pF to 47 pF) (Max)
SSP_13	TSOH	MOSI hold after SCK	73	—	—	ns	
SSP_15	TSIS	MISO set-up Time of data input to SCK	27	—	—	ns	
SSP_17	TSIH	MISO hold time of data input to SCK	24	—	—	ns	
SSP_19	TSSS	SS set-up to SCK (PRELOADEN = 1)	50	—	—	ns	
SSP_21	TSSH	SS hold after SCK client	80	—	—	ns	

Figure 43-5. SPI Client Module CPHA = 0 Timing Diagrams

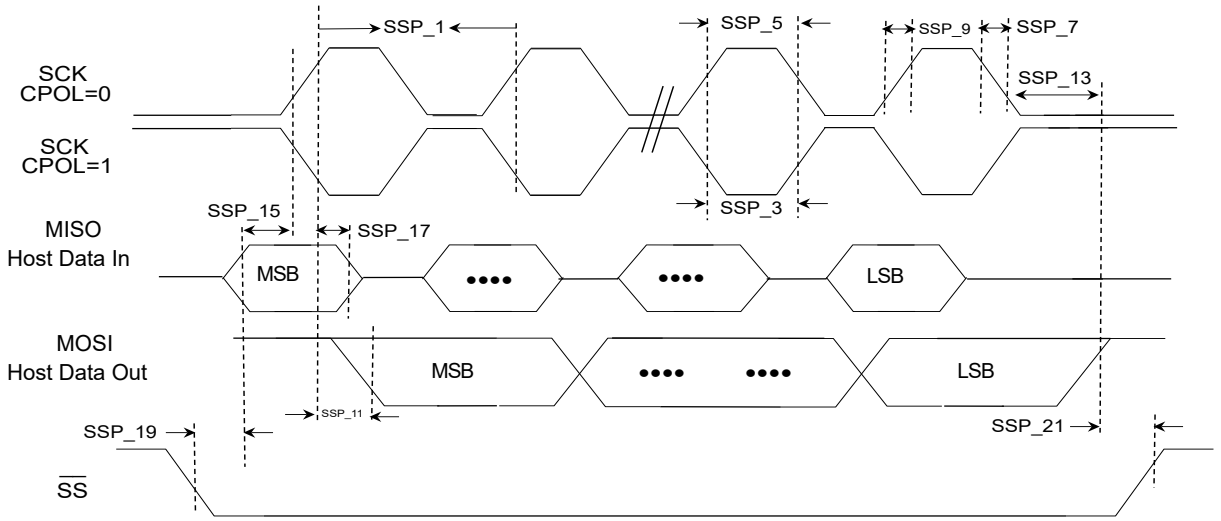
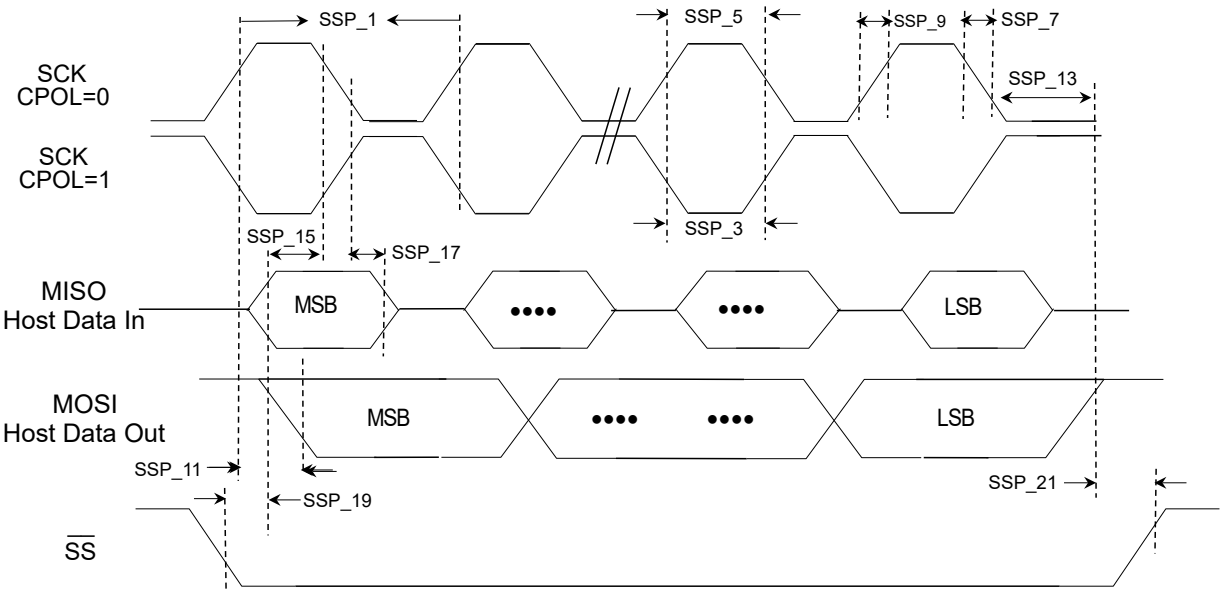


Figure 43-6. SPI Client Module CPHA = 1 Timing Diagrams



43.20 UART AC Electrical Specifications

Table 43-28. UART AC Electrical Specifications

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics		Min.	Typical	Max.	Units	Conditions
UT_1	F_{BRATE}	Baud rate	Asynchronous SAMPR = 16x mode	3.995	3.999	4.005	Mbps	C_{LOAD} = 30 $pF_{(MAX)}$
UT_3			Asynchronous SAMPR = 8x mode	7.981	7.993	7.981	Mbps	C_{LOAD} = 30 $pF_{(MAX)}$
UT_5			Asynchronous SAMPR = 3x mode	21.226	21.338	21.602	Mbps	C_{LOAD} = 30 $pF_{(MAX)}$
UT_6	T_{LW}	Baud clock period low time (Synchronous mode)		5	—	—	ns	—
UT_7	T_{HW}	Baud clock period high time (Synchronous mode)		5	—	—	ns	—
UT_8	T_S	Setup time before sampling edge		5	—	—	ns	—
	T_H	Hold time after sampling edge		5	—	—	ns	—
UT_26	F_{ICK}	Internal baud clock (Synchronous mode)		—	—	32	MHz	Maximum driven XCK of 32 MHz (DIV by 2 of internal clock (fixed pin)
				—	—	32	MHz	Maximum driven XCK of 32 MHz (DIV by 2 of internal clock (PPS)
UT_26	F_{XCK}	External baud clock (Synchronous mode)		—	—	32	MHz	Maximum external XCK (fixed pin)
				—	—	32	MHz	Maximum external XCK (PPS pin)

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
UT_22	F_{USART}	USART max GCLK_SERCOM (Synchronous mode)	—	—	32	MHz	Maximum driven XCK of 32 MHz (DIV by 2 of internal clock)
UT_24	F_{XCK}	USART external clock input (Synchronous mode)	—	—	32	MHz	Maximum external XCK

Note: These parameters are characterized but not tested in manufacturing.

43.21 I²C Module Electrical Specifications

Table 43-29. I²C Module Host Mode Electrical Specifications

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +85°C for Extended Temp			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CM_1	TL0:SCL	Host clock low time	100 kHz mode	4.7	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	1.12	—	µs	
			1 MHz mode	0.5	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_3	THI:SCL	Host clock high time	100 kHz mode	4.07	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	1.062	—	µs	
			1 MHz mode	0.3	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_5	TF:SCL	SDAx and SCLx fall time	100 kHz mode	—	250	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	20+(0.1* C_{Load})	250	ns	
			1 MHz mode	20+(0.1* C_{Load})	250	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_7	TR:SCL	SDAx and SCLx rise time	100 kHz mode	—	1000	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	20+(0.1* C_{Load})	300	ns	
			1 MHz mode	—	120	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_9	TSU:DAT	Data input setup time	100 kHz mode	104	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	104	—	ns	
			1 MHz mode	104	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_11	THD:DAT	Data input hold time	100 kHz mode	9	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	9	—	ns	
			1 MHz mode	9	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_13	TSU:STA	Start condition setup time	100 kHz mode	TL0:SCL +7	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode			ns	
			1 MHz mode			ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf

.....continued

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CM_15	THD:STA	Start condition hold time	100 kHz mode	4.4	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3mA, C_{LOAD} = 400 pf
			400 kHz mode	1.03	—	µs	
			1 MHz mode	0.369	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_17	TSU:ST0	Stop condition setup time	100 kHz mode	5.04	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3mA, C_{LOAD} = 400 pf
			400 kHz mode	0.77	—	µs	
			1 MHz mode	0.494	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_21	TAA:SCL	Output valid from clock	100 kHz mode	—	101	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	—	124	ns	
			1 MHz mode	—	130	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CM_23	TBF:SDA	Bus free time(1)	100 kHz mode	TL0:SCL	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode		—	ns	
			1 MHz mode		—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf

Note:

1. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

Figure 43-7. I^2C Start/Stop Bits Host Mode AC Timing Diagram

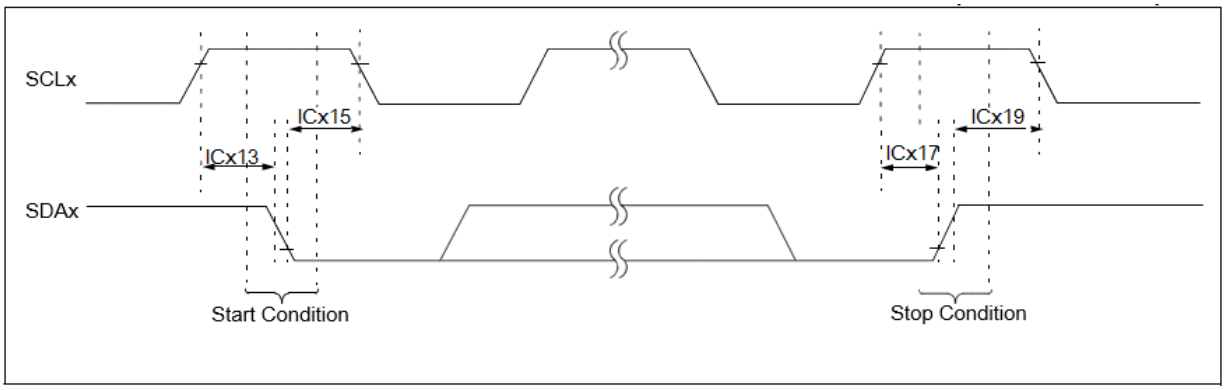


Figure 43-8. I²C Bus Data Host Mode AC Timing Diagram

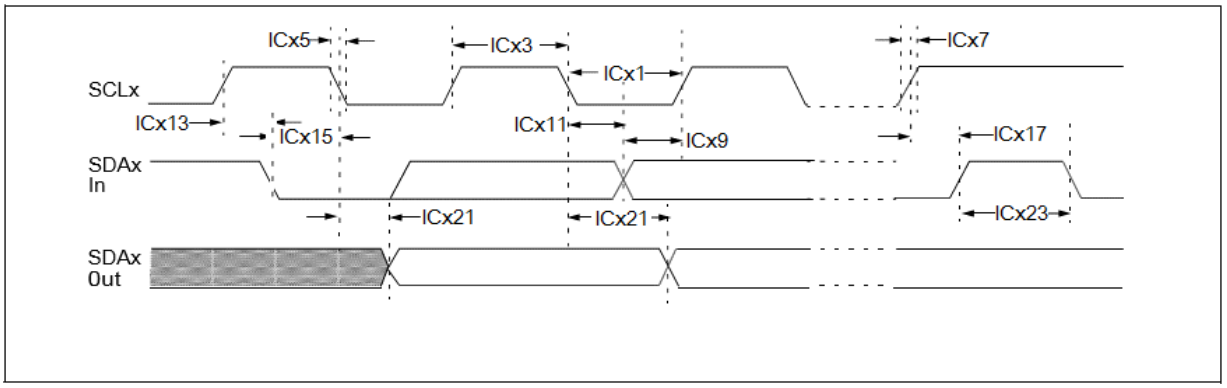


Table 43-30. I²C Module Client Mode Electrical Specifications

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CS_1	TL0:SCL	Client cock low time	100 kHz mode	4.7	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	1.12	—	µs	
			1 MHz mode	0.5	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CS_3	THI:SCL	Client clock high time	100 kHz mode	4.3	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	1.05	—	µs	
			1 MHz mode	0.3	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CS_5	TF:SCL	SDAx and SCLx fall time	100 kHz mode	—	250	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	20+(0.1*C_{Load})	250	ns	
			1 MHz mode	20+(0.1*C_{Load})	250	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CS_7	TR:SCL	SDAx and SCLx rise time	100 kHz mode	—	1000	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	20+(0.1*C_{Load})	300	ns	
			1 MHz mode	—	120	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
I2CS_9	TSU:DAT	Data input setup time	100 kHz mode	51	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	51	—	ns	
			1 MHz mode	51	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
I2CS_11	THD:DAT	Data input hold time	100 kHz mode	71	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	71	—	ns	
			1 MHz mode	71	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf

.....continued

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CS_13	TSU:STA	Start condition setup time	100 kHz mode	TL0:SCL +7	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	TL0:SCL +7	—	ns	
			1 MHz mode	TL0:SCL +7	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
I2CS_15	THD:STA	Start condition hold time	100 kHz mode	4.464	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	1.02	—	µs	
			1 MHz mode	0.358	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
I2CS_17	TSU:ST0	Stop condition setup time	100 kHz mode	4.9	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
			400 kHz mode	0.72	—	µs	
			1 MHz mode	0.49	—	µs	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
I2CS_21	TAA:SCL	Output valid from clock	100 kHz mode	—	128	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	—	134	ns	
			1 MHz mode	—	138	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf
I2CS_23	TBF:SDA	Bus free time (1)	100 kHz mode	TL0:SCL	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 3 mA, C_{LOAD} = 400 pf
			400 kHz mode	TL0:SCL	—	ns	
			1 MHz mode	TL0:SCL	—	ns	V_{DDIO} = 3.3V, $I_{PULL-UP}$ = 20 mA, C_{LOAD} = 550 pf

Note:

1. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

Figure 43-9. I²C Start/Stop Bits Client Mode AC Timing Diagram

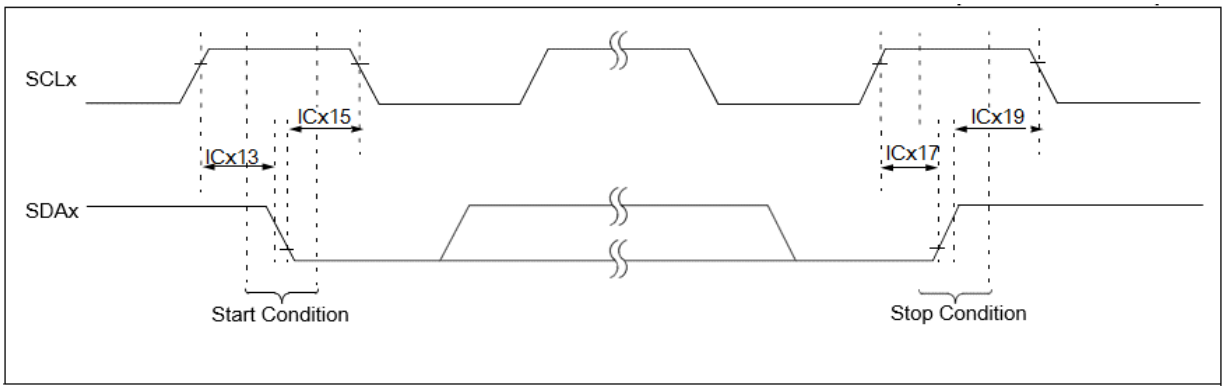
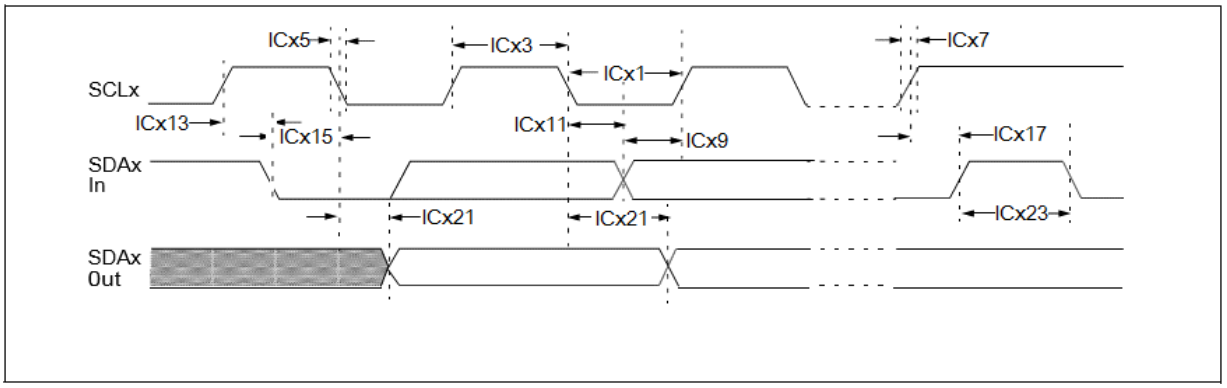


Figure 43-10. I²C Bus Data Client Mode AC Timing Diagram



43.22 QSPI Module Electrical Specifications

Table 43-31. QSPI Module Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions(2)
QSPI_1	FCLK	SQI serial clock frequency • SDR Host mode 0 and 2	—	—	32	MHz	V_{DDIO} = 3.3V, C_{LOAD} = 30 pf(MAX)
QSPI_2	FCLK	SQI serial clock frequency • SDR Host mode 1 and 3	—	—	32	MHz	V_{DDIO} = 3.3V, C_{LOAD} = 30 pf(MAX)
QSPI_3	TSCKH	Serial clock high time	1/(2*FCLK)	—	—	ns	—
QSPI_5	TSCKL	Serial clock low time	1/(2*FCLK)	—	—	ns	—
QSPI_7	TSCKR	Serial clock rise time	—	—	14.12	ns	See parameter DI27 I/O spec
QSPI_9	TSCKF	Serial clock fall time	—	—	14.6	ns	See parameter DI25 I/O spec
QSPI_11	TCSS	CS active setup time	5	—	—	ns	—
QSPI_13	TCSH	CS active hold time	5	—	—	ns	—
QSPI_19	TDIS	Data in setup time	6	—	—	ns	—
QSPI_21	TDIH	Data in hold time	3	—	—	ns	—
QSPI_23	TDOH	Data out hold	0	—	—	ns	—
QSPI_25	TDOV	Data out valid	—	—	6	ns	—

Notes:

1. Assumes $V_{DDIOx\ (min)}$ and 30 pF external load on all SQI pins unless otherwise noted.
2. These parameters are characterized, but not tested in manufacturing.

43.23 TCx Timer Capture Module AC Electrical Specifications

Table 43-32. TCx Timer Capture Module AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
TC_1	TCINLOW	Capture TCx input low time	2/fGLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TC_5 spec
TC_3	TCINHIGH	Capture TCx input high time	2/fGLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TC_5 spec
TC_5	TCINPERIOD	Capture input period	4/fGLK_TCx	—	—	ns	$V_{DDIO(min)}$
TC_7	TCOUTLOW	Compare TCx output low time	1/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TC_11 spec
TC_9	TCOUTHIGH	Compare TCx output high time	1/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TC_11 spec
TC_11	TCOUTPERIOD	Compare output period	2/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$
TC_13	fGCLK_TCx	TC peripheral module clock frequency	—	—	64	MHz	$V_{DDIO(min)}$
Note: These parameters are characterized but not tested in manufacturing.							

Figure 43-11. TCx Timer Capture Input Module AC Timing Diagram

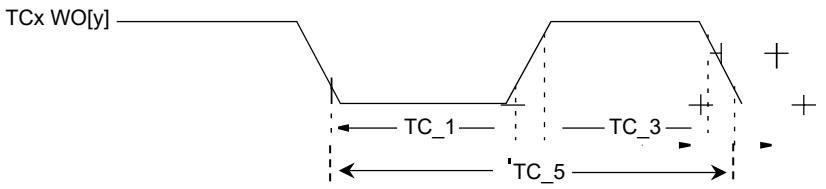
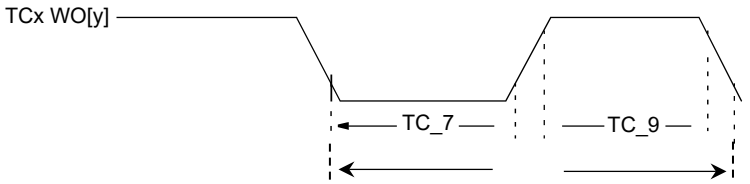


Figure 43-12. TCx Timer Capture Output Module AC Timing Diagram



43.24 TCCx Timer Capture Module AC Electrical Specifications

Table 43-33. TCCx Timer Capture Module AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
TCC_1	TCCINLOW	Capture TCCx input low time	2/fGLK_TCCx	—	—	ns	$V_{DDIO(min)}$ and meet TCC_5 spec

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
TCC_3	TCCINHIGH	Capture TCCx input high time	2/fGLK_TCCx	—	—	ns	$V_{DDIO(min)}$ and meet TCC_5 spec
TCC_5	TCCINPERIOD	Capture input period	4/fGLK_TCCx	—	—	ns	$V_{DDIO(min)}$
TCC_7	TCCOUTLOW	Compare TCCx output low time	1/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TCC_11 spec
TCC_9	TCCOUTHIGH	Compare TCCx output high time	1/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$ and meet TCC_11 spec
TCC_11	TCCOUTPERIOD	Compare output period	2/fGCLK_TCx	—	—	ns	$V_{DDIO(min)}$
TCC_13	fGCLK_TCCx	TCC peripheral module clock frequency	—	—	64	MHz	
TCC_15	TCCFD	Fault input to I/O pin change	—	—	63	ns	
TCC_17	TCCFLT	Fault input pulse width	12	—	—	ns	

Note: These parameters are characterized but not tested in manufacturing.

Figure 43-13. TCCx Timer Capture Input Module AC Timing Diagram

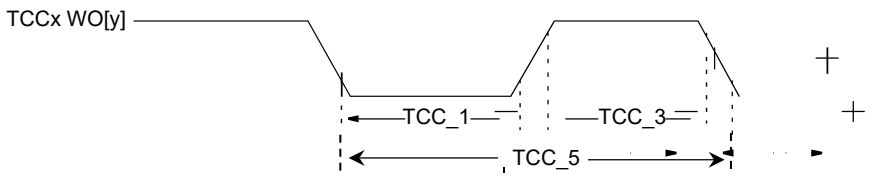


Figure 43-14. TCCx Timer Capture Output Module AC Timing Diagram

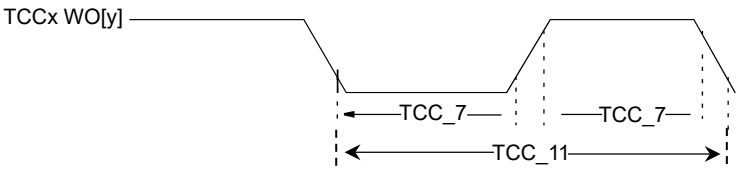
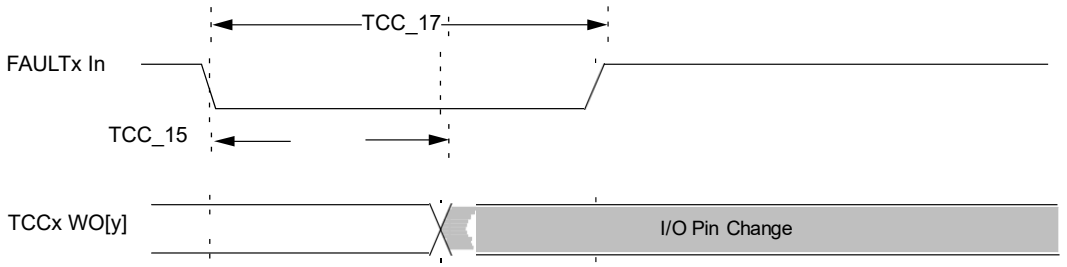


Figure 43-15. TCC_x Timer Compare Fault Output Module AC Timing Diagram



43.25 FLASH NVM AC Electrical Specifications

Table 43-34. FLASH NVM AC Electrical Specifications

AC Characteristics				Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics		Min.	Typical	Max.	Units	Conditions
NVM_1	F_{RETEN}	Flash data retention		20	—	—	Yrs	Under all conditions less than absolute maximum ratings specifications
NVM_3	EP	Cell endurance (Flash erase and Write operation)		20k	—	—	Cycles	
NVM_5	FREAD	Flash read	2 wait states	—	—	—	MHz	V_{DDIO} = 3.3V
NVM_7	TFPW	Program cycle time	Write page	4	—	6	ms	V_{DDIO} = 1.9V
NVM_9	TCE	Chip erase cycle time	Erase chip	37	—	42	ms	
NVM_15	TFEP	Page erase time	Erase page	6	—	8	ms	—

Note:

1. The maximum FLASH operating frequencies are given in the table above but are limited by the Embedded Flash access time when the processor is fetching code out of it. This field defines the number of Wait states required to access the Embedded Flash Memory.

43.26 Frequency Meter AC Electrical Specifications

Table 43-35. Frequency Meter AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
FM_1	FM_{LOW}	REFI input low time	6.98	—	—	ns	$V_{DDIO\ (min)}$ and meet FM5 spec
FM_3	FM_{HIGH}	REFI input high time	5.1	—	—	ns	
FM_5	FM_{PERIOD}	REFI input period	15	—	—	ns	$V_{DDIO\ (min)}$
FM_7	fGCLK_FREQM_REF	FREQM reference	—	—	65.43	MHz	
FM_9	fGCLK_FREQM_MSR	FREQM measure	—	—	66.66	MHz	

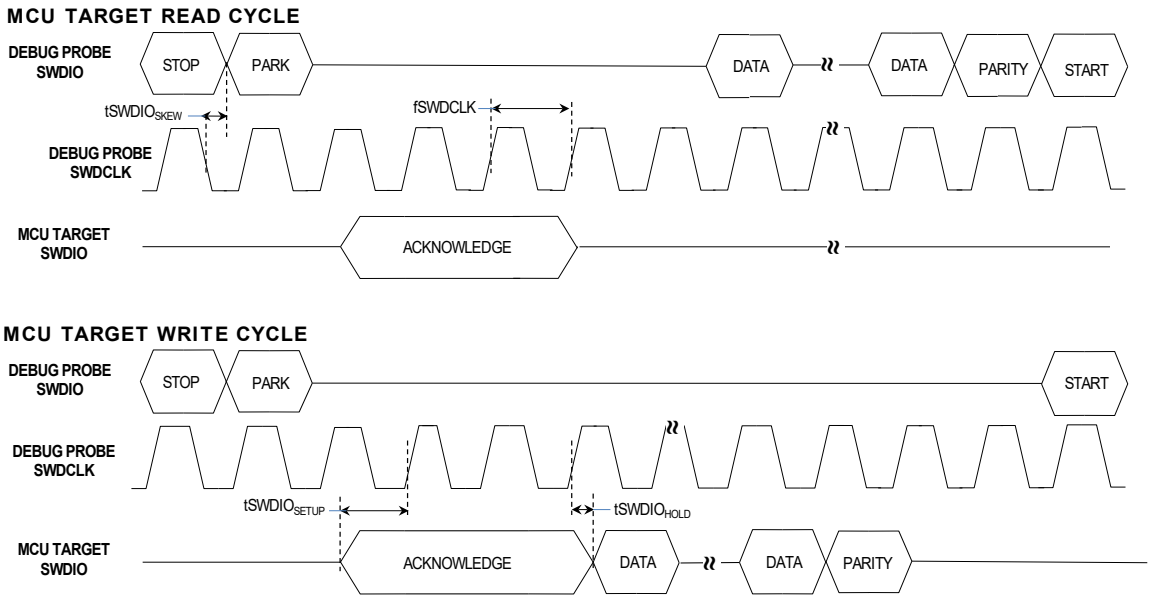
43.27 SWD 2-Wire AC Electrical Specifications

Table 43-36. SWD 2-Wire AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
SWD_1	fSWDCLK	SWDCLK clock frequency	—	—	25	MHz	V_{DD} at 1.9V – 3.6V
SWD_3	$tSWDCLK_{HIGH}$	SWDCLK clock high time	20	—	—	ns	—
SWD_5	$tSWDCLK_{LOW}$	SWDCLK clock low time	20	—	—	ns	—
SWD_4	$tSWDCLK_{RISE}$	SWDCLK clock rise time	—	—	9.5	ns	—

.....continued							
AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
SWD_6	$tSWDCLK_{FALL}$	SWDCLK clock fall time	—	—	9.5	ns	—
SWD_7	$tSWDIO_{SKEW}$	SWDIO skew	-5	—	+5	ns	—
SWD_9	$tSWDIO_{SETUP}$	SWDIO setup time	4	—	—	ns	—
SWD_11	$tSWDIO_{HOLD}$	SWDIO hold time	1	—	—	ns	—
Note: These parameters are characterized but not tested in manufacturing.							

Figure 43-16. SWD 2-Wire Read/Write AC Timing Diagram



43.28 FRC Clock AC Electrical Specifications

Table 43-37. FRC Clock AC Electrical Specifications

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
FRC_1	F20	FRC accuracy	—	±0.9	—	%	-40°C ≤ T_A ≤ +85°C
			—	±5	—	%	-40°C ≤ T_A ≤ +125°C
FRC_2	F_{INT}	FRC frequency	—	8	—	MHz	—
FRC_3	T_{INT}	FRC period	—	1/FINT	—	ns	—
FRC_4	T_{SUFRC}	FRC start-up time	—	9.82	13.46	µs	—
FRC_5	T_{STABLE}	FRC stabilization period	—	8 x TINT	—	ns	8 clocks
FRC_9	D_{CYCLE}	Duty cycle of square wave output	—	50	—	%	—
FRC_14	C_{USE}	User tune	-1.488	—	1.646	%	Rounded to three decimals

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
FRC_15	S_{USE}	User tune step size	0.0418	0.047	0.098	%	Rounded to three decimals
FRC_16	IFRC_ON	Current consumed when oscillator is running	—	262	375	µA	FOSC = 8.00 MHz; V_{DDCORE} = 1.2 V; T_J = 105°C

Notes:

1. Data in the "Typ" column is at 3.3V, T_A = +25°C unless otherwise stated.
2. Data for this parameter is Preliminary. This parameter is characterized but not tested in manufacturing.
3. This assumes osctune[5:0] = 000000 and test calibration is complete.
4. This parameter is characterized but not tested in manufacturing.

43.29 Bluetooth Low Energy RF Characteristics

Table 43-38. Bluetooth Low Energy RF Characteristics

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions(2)
BTG1	FREQ	Frequency range of operation	2402	—	2480	MHz	—
BTTX1	TXPWR:PA	Bluetooth® transmit power PA	—	11	—	dBm	—
BTX2	TXIB:1MBPS	In-band emission for FTX ± -2 MHz	—	-40	—	dBm	—
		In-band emission for FTX ± -(3+N) MHz	—	-46	—	dBm	—
BTX3	TXIB:2MBPS	In-band emission for FTX ± -4 MHz	—	-41	—	dBm	—
		In-band emission for FTX ± -5 MHz	—	-52	—	dBm	—
		In-band emission for FTX ± -(6+N) MHz	—	-53	—	dBm	—
BTRX1	RXSENSE	Receiver sensitivity at 1 Mbps	—	-98	—	dBm	—
		Receiver sensitivity at 2 Mbps	—	-94.5	—	dBm	—
		Receiver sensitivity at S=8	—	-108	—	dBm	—
		Receiver sensitivity at S=2	—	-102.5	—	dBm	—
BTRX2	MAXINSIG	Maximum input signal level at 1 Mbps	—	-2	—	dBm	—
		Maximum input signal level at 2 Mbps	—	-2	—	dBm	—
		Maximum input signal level at S=2	—	-2	—	dBm	—
		Maximum input signal level at S=8	—	-2	—	dBm	—

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions(2)
BTRX3	CI1M:COCH	C/I Co channel rejection	—	14	—	dB	—
	CI1M: ± -1 MHz	C/I adjacent channel rejection	—	16	—	dB	—
	CI1M: ± -2 MHz	C/I adjacent channel rejection	—	7	—	dB	—
	CI1M:ADJ(3+n)	C/I alternate channel rejection	—	8	—	dB	—
	CI1M:IMG	C/I image frequency rejection	—	12	—	dB	—
	CI1M:IMG ± -1 MHz	C/I adjacent channel to image freq rejection	—	11	—	dB	—
BTRX4	CIS2:COCH	C/I Co channel rejection	—	9	—	dB	—
	CIS2: ± -1 MHz	C/I adjacent channel rejection	—	20	—	dB	—
	CIS2: ± -2 MHz	C/I adjacent channel rejection	—	15	—	dB	—
	CIS2:ADJ(3+n)	C/I alternate channel rejection	—	7	—	dB	—
	CIS2:IMG	C/I image frequency rejection	—	15	—	dB	—
	CIS2:IMG ± -1 MHz	C/I adjacent channel to image freq rejection	—	13	—	dB	—
BTRX5	CIS8:COCH	C/I Co channel rejection	—	5	—	dB	—
	CIS8: ± -1 MHz	C/I adjacent channel rejection	—	20	—	dB	—
	CIS8: ± -2 MHz	C/I adjacent channel rejection	—	11	—	dB	—
	CIS8:ADJ(3+n)	C/I alternate channel rejection	—	3	—	dB	—
	CIS8:IMG	C/I image frequency rejection	—	11	—	dB	—
	CIS2:IMG ± -1 MHz	C/I adjacent channel to image freq rejection	—	15	—	dB	—
BTRX6	CI2M:COCH	C/I Co channel rejection	—	13	—	dB	—
	CI2M: ± -2 MHz	C/I adjacent channel rejection	—	14	—	dB	—
	CI2M: ± -4 MHz	C/I adjacent channel rejection	—	12	—	dB	—
	CI2M:ADJ(6+2n)	C/I alternate channel rejection	—	15	—	dB	—
	CI2M:IMG	C/I image frequency rejection	—	14	—	dB	—
	CI2M:IMG ± -2 MHz	C/I adjacent channel to image freq rejection	—	13	—	dB	—

.....continued

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions(2)
BTRX7	BLOCK1M:<2 GHZ	Blocking performance from 30-2 GHz	—	20	—	dB	—
	BLOCK1M:2 GHZ<SIG<2399 MHz	Blocking performance from 2003-2399 MHz	—	19	—	dB	—
	BLOCK1M:2484 MHZ<SIG<2977 MHz	Blocking performance between 2484-2997 MHz	—	20	—	dB	—
	BLOCK1M:3 GHZ<SIG<12.75 GHz	Blocking performance between 3-12.5 GHz	—	20	—	dB	—
BTRX8	BLE1M:INTERMOD	Inter modulation performance for BLEM	—	10.5	—	dB	—
	BLE2M:INTERMOD	Inter modulation performance for BLEM	—	11.5	—	dB	—

Notes:

1. Data in the "Typ" column is at T_A = 25°C.
2. This parameter is characterized but not tested in manufacturing.

Table 43-39. Bluetooth Low Energy RF Current Characteristics

AC Characteristics(3)					Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	RF Power	CPU Frequency	Min.	Typ.(1)	Max.	Units	Conditions(2)
IBLETX1	IDDTXPA1M	Current Consumption with TX in MLDO mode 1 Mbps	+11 dBm	64 MHz	—	49	—	mA	—
IBLETX2			+11 dBm	32 MHz	—	45.5	—	mA	—
IBLETX3			10 dBm	64 MHz	—	44	—	mA	—
IBLETX4			10 dBm	32 MHz	—	40	—	mA	—
IBLETX5			0 dBm	64 MHz	—	26	—	mA	—
IBLETX6			0 dBm	32 MHz	—	22	—	mA	—
IBLERX1	IDDRXBLE1M	Current consumption at RX signal MLDO mode	-90 dBm	64 MHz	—	20	—	mA	—
IBLERX2		Current consumption at RX signal MLDO mode	-90 dBm	32 MHz	—	16	—	mA	—

Notes:

1. Data in the "Typ" column is at V_{DD} = 3.3V, T_A = 25°C.
2. This parameter is characterized but not tested in manufacturing.
3. The current number includes the FW default operation current consumption.

Figure 43-17. Bluetooth Low Energy Transmit Power vs. Frequency

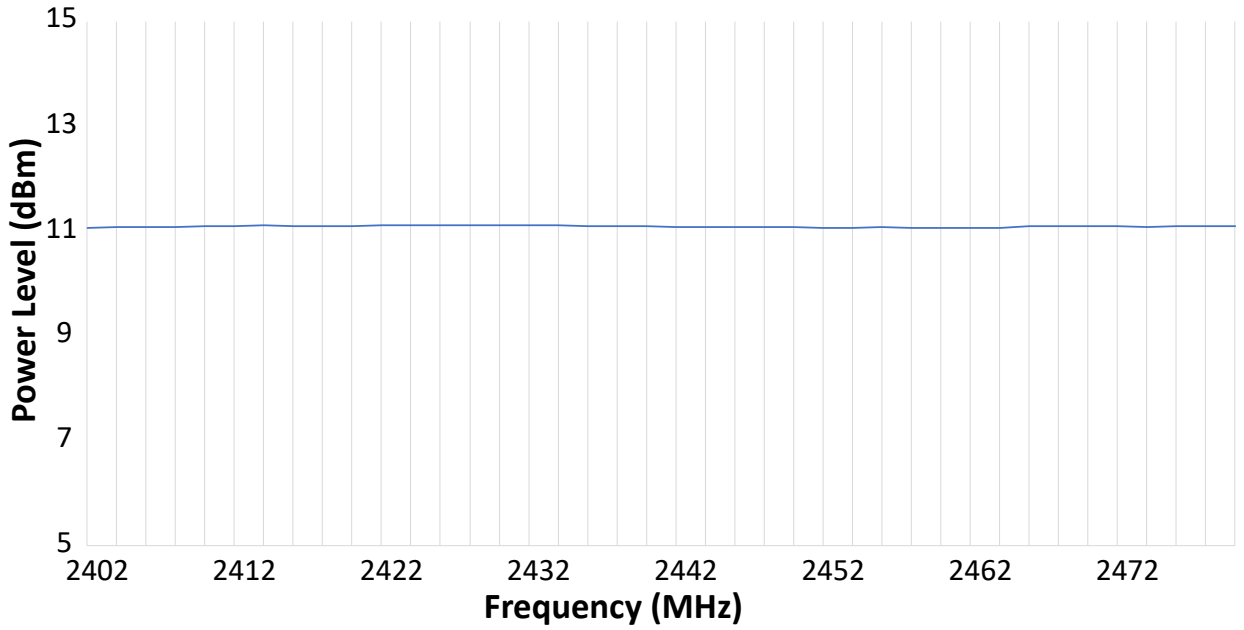


Figure 43-18. Bluetooth Low Energy Transmit Power vs. Temperature

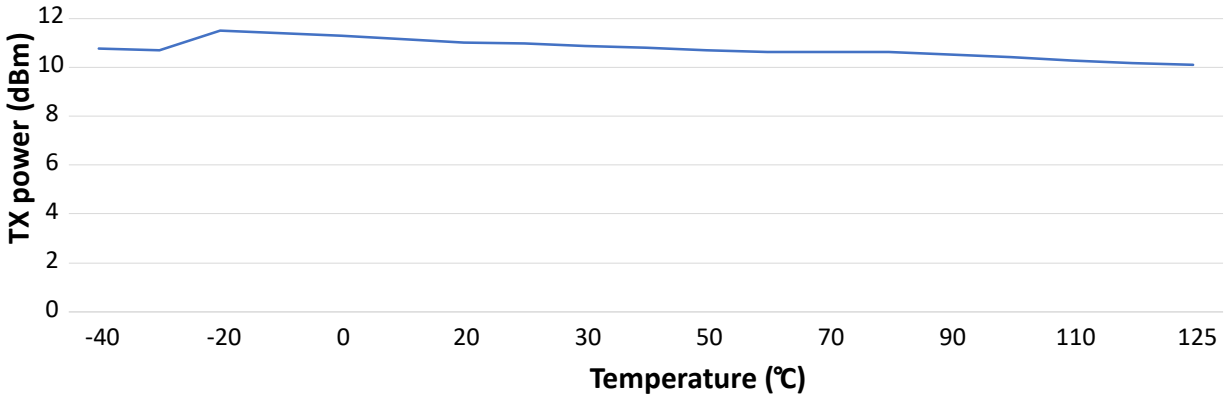


Figure 43-19. Bluetooth Low Energy Transmit Power vs. Transmit Power Level

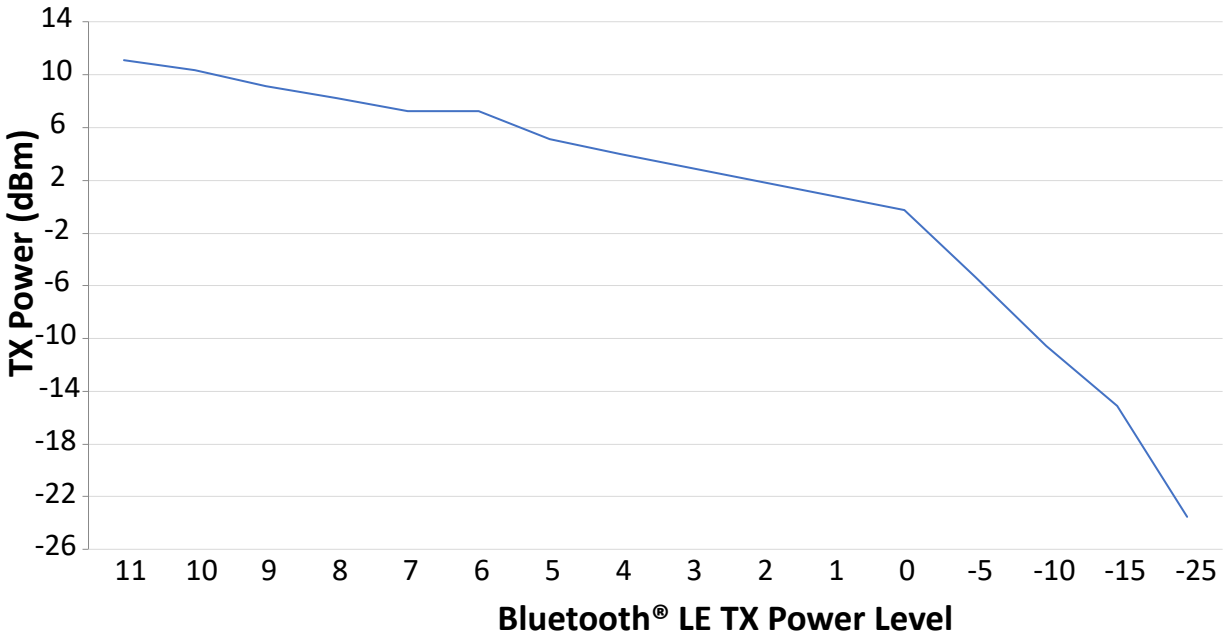


Figure 43-20. Bluetooth Low Energy Receive Sensitivity vs. Temperature

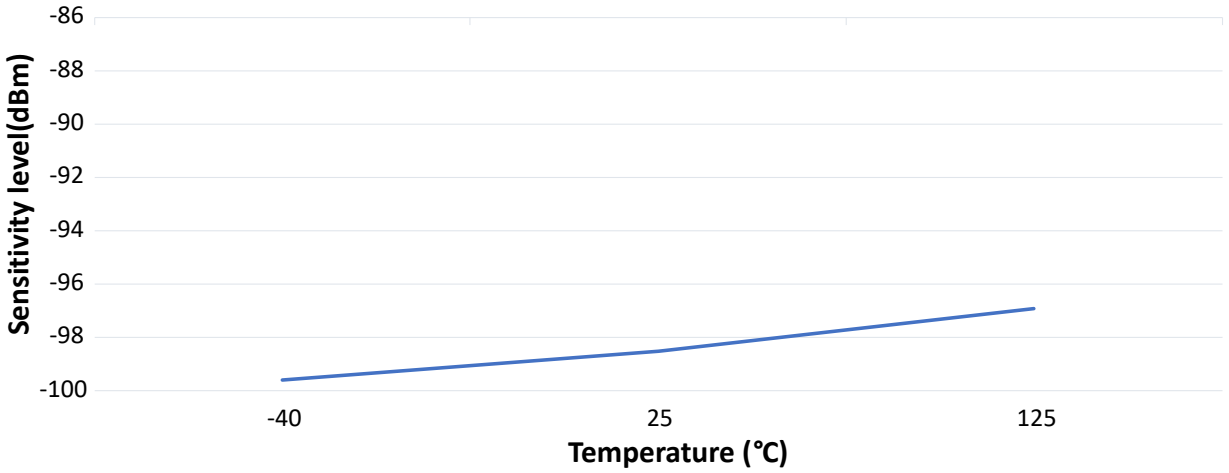


Figure 43-21. WBZ35x Module Bluetooth Low Energy Receive Sensitivity vs. Temperature

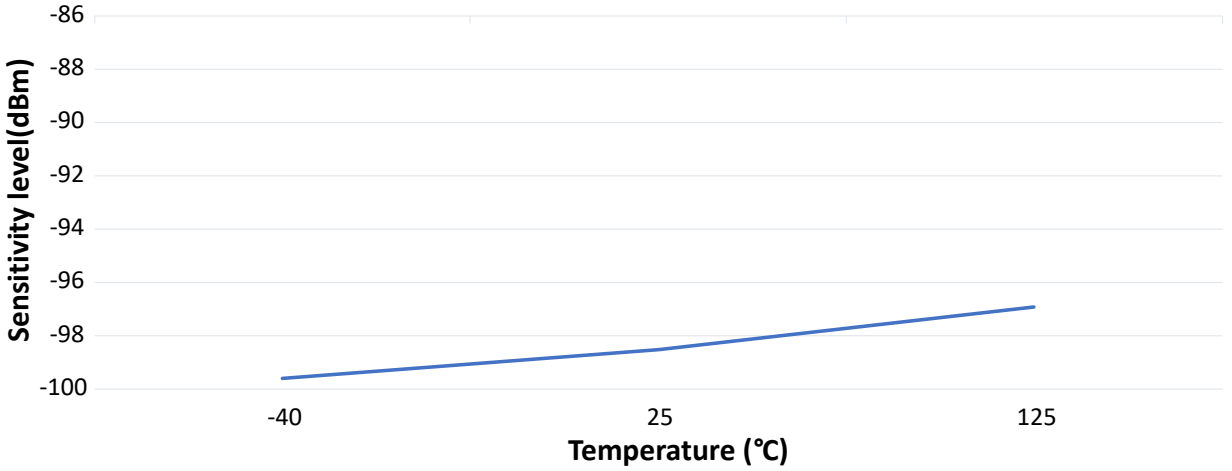


Figure 43-22. Bluetooth Low Energy Receive Sensitivity vs. Frequency

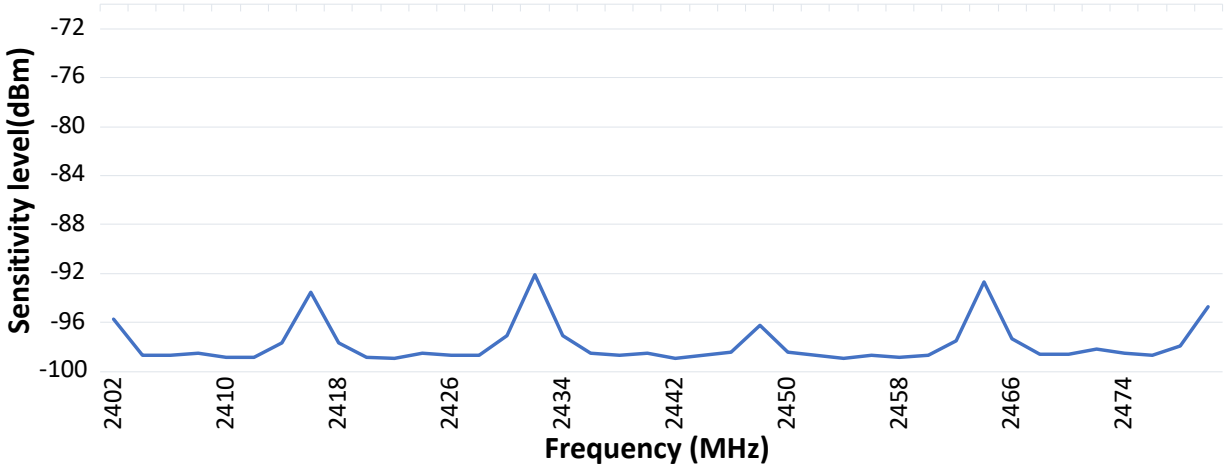


Figure 43-23. WBZ35x Module Bluetooth Low Energy Receive Sensitivity vs. Frequency

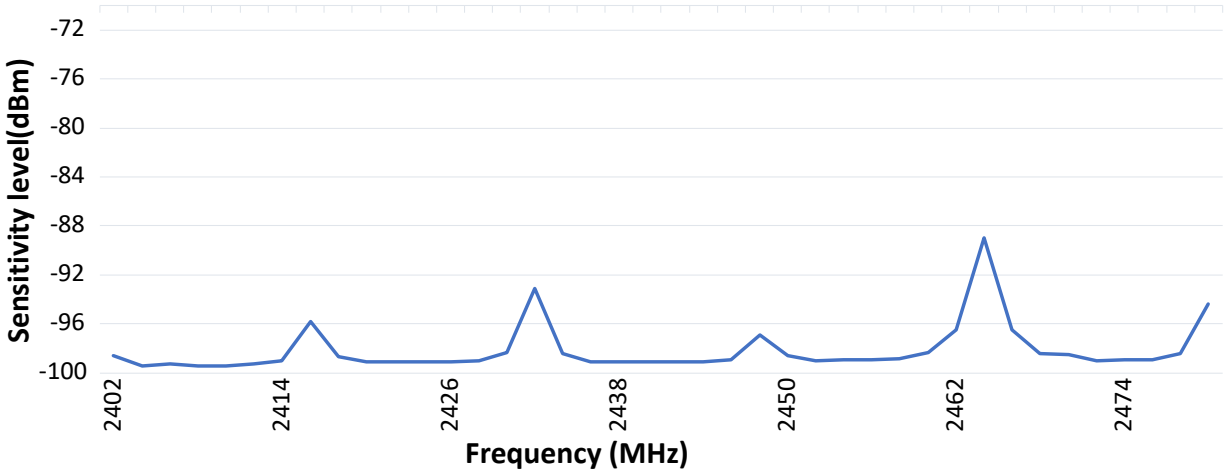


Figure 43-24. Bluetooth Low Energy Transmit Power vs. VDD Supply Voltage

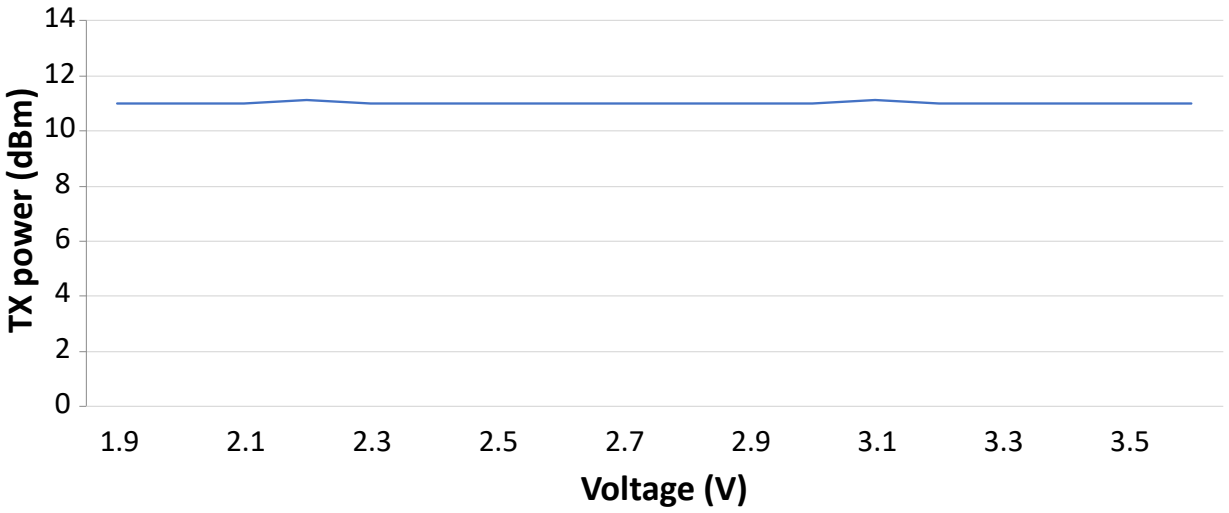


Figure 43-25. Bluetooth Low Energy Receive Sensitivity vs. VDD Supply Voltage

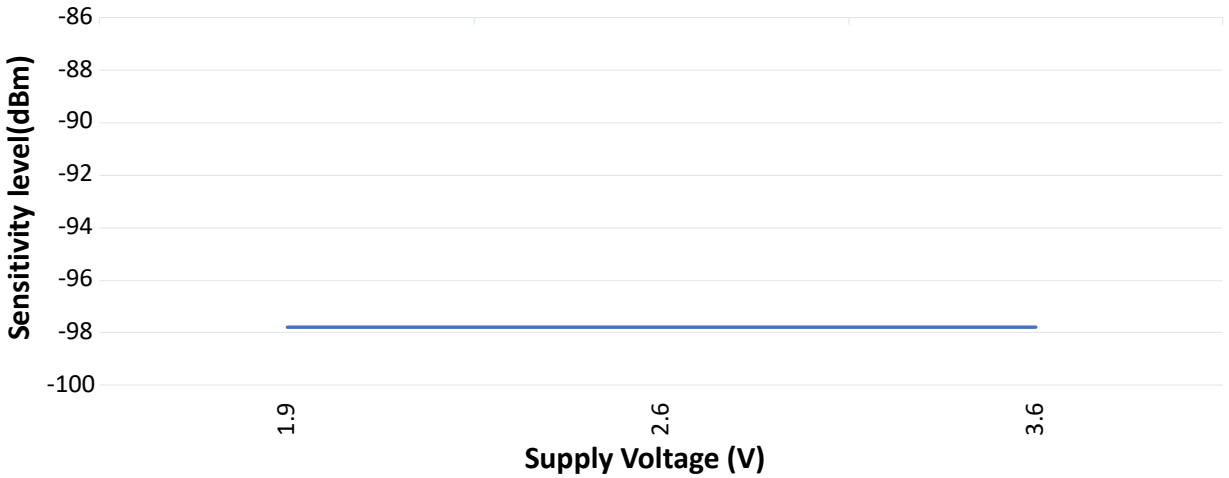


Figure 43-26. Bluetooth Low Energy RX CI Margin

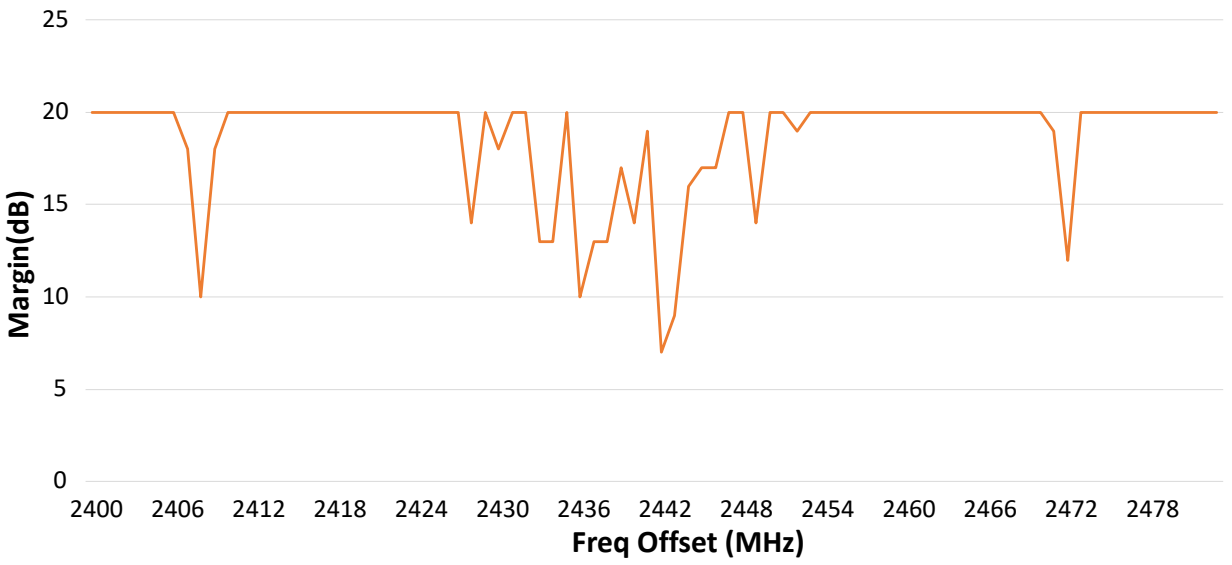


Figure 43-27. Bluetooth Low Energy Transmit Current vs. Temperature

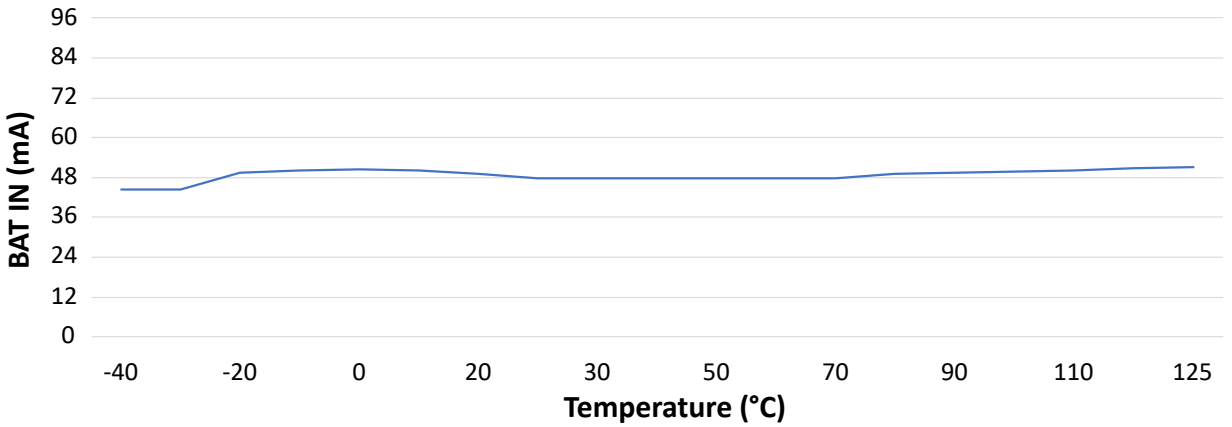
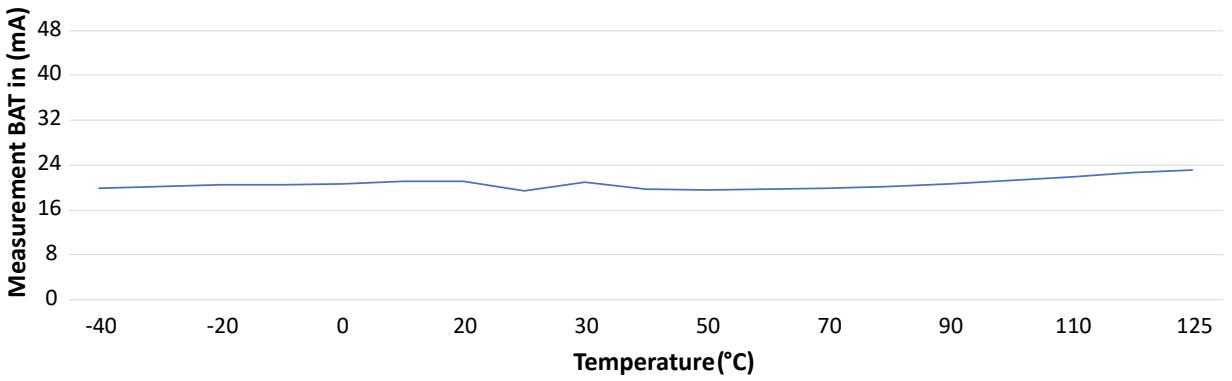


Figure 43-28. Bluetooth Low Energy Receive Current vs. Temperature



43.30 Zigbee RF Characteristics

Table 43-40. Zigbee RF Characteristics

AC Characteristics			Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions(2)
ZBG1	FREQ	Frequency range	2405	—	2480	MHz	—
ZBG2	FCH	Channel spacing	—	5	—	MHz	—
ZBG3	PSDU	Bit rate	—	250	—	kbps	—
ZBT1	TXOPPA	Transmit output power PA	—	11	—	dBm	—
ZBT2	POWERRANGE	Output power range	-12	—	11	dBm	—
ZBT3	EVM	Error vector magnitude	—	10	—	%RMS	—
ZBT4	PA2HAR	Second harmonic from PA	—	-45	—	dBm	—
ZBT5	PA3HAR	Third harmonic from PA	—	-55	—	dBm	—
ZBRX1	SENS	Receiver sensitivity in 250 kbps	—	-103	—	dBm	—
		Receiver sensitivity in 500 kbps	—	-99	—	dBm	—
		Receiver sensitivity in 1 Mbps	—	-96	—	dBm	—
		Receiver sensitivity in 2 Mbps	—	-90	—	dBm	—
ZBRX2	PMAX	Maximum input level	—	-2	—	dBm	—
ZBRX3	PACRP	Adjacent channel rejection +5 MHz	—	19	—	dB	—
ZBRX4	PACRN	Adjacent channel rejection -5 MHz	—	18	—	dB	—
ZBRX5	PALRP	Alternate channel rejection +10 MHz	—	16	—	dB	—
ZBRX6	PALRN	Alternate channel rejection -10 MHz	—	12	—	dB	—
ZBRX7	RSSIRANGE	Dynamic range of RSSI	—	80	—	dB	—
ZBRX8	RSSIRES	Resolution of RSSI	—	1	—	dB	—
ZBRX9	RSSIBASEVAL	Minimum value of RSSI	—	-104	—	dBm	—

Notes:

1. Data in “Typ” column is at T_A = 25°C.
2. This parameter is characterized but not tested in manufacturing.

Table 43-41. Zigbee RF Current Characteristics

AC Characteristics[3]					Standard Operating Conditions: V_{DD} = 1.9V to 3.6V (unless otherwise stated) Operating Temperature: -40°C ≤ T_A ≤ +85°C for Industrial Temp -40°C ≤ T_A ≤ +125°C for Extended Temp				
Param. No.	Symbol	Characteristics	RF Power	CPU Frequency	Min.	Typ.[1]	Max.	Units	Conditions[2]
IZBTX1	IDDTX	TX current consumption in MLDO mode 250 kbps	+11 dBm	64 MHz	—	50	—	mA	—
IZBTX2			+11 dBm	32 MHz	—	46.5	—	mA	—
IZBTX3			10 dBm	64 MHz	—	45	—	mA	—
IZBTX4			10 dBm	32 MHz	—	41	—	mA	—
IZBTX5			0 dBm	64 MHz	—	26	—	mA	—
IZBTX6			0 dBm	32 MHz	—	21.5	—	mA	—
IZBRX1	IDDRXZB	Current consumption at Rx signal level -90 dBm in MLDO mode 250 kbps	-90 dBm	64 MHz	—	20	—	mA	—
IZBRX2			-90 dBm	32 MHz	—	16	—	mA	—
IZBRX3	IDDRXZBRPC	Current consumption at Rx signal level -90 dBm in MLDO mode 250 kbps RPC	-90 dBm	64 MHz	—	18.5	—	mA	—
IZBRX4			-90 dBm	32 MHz	—	14.5	—	mA	—

Notes:

1. Data in the “Typ” column is at V_{DD} = 3.3V, T_A = 25°C.
2. This parameter is characterized but not tested in manufacturing.
3. The current number includes the FW default operation current consumption.

Figure 43-29. Zigbee Transmit Power vs. Temperature

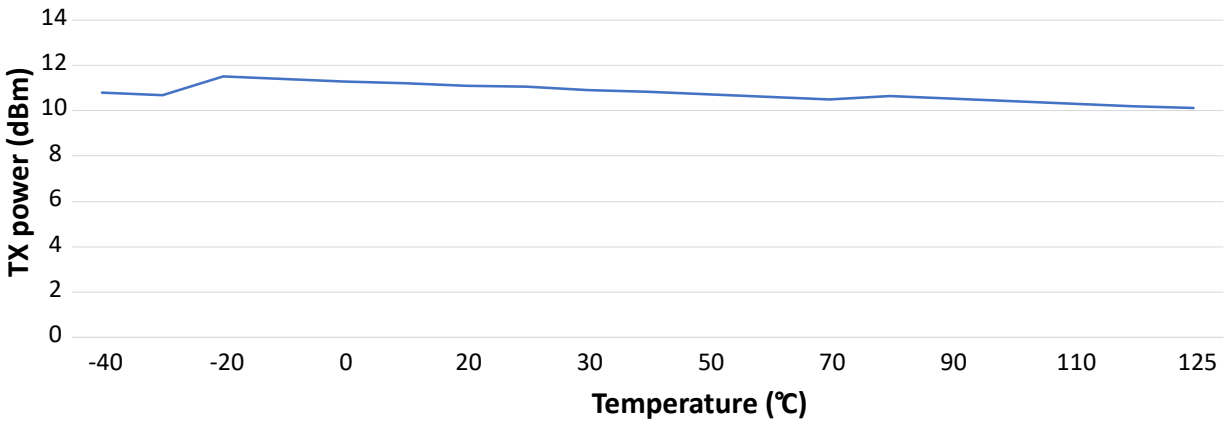


Figure 43-30. Zigbee TX Setting vs. Measurement Power

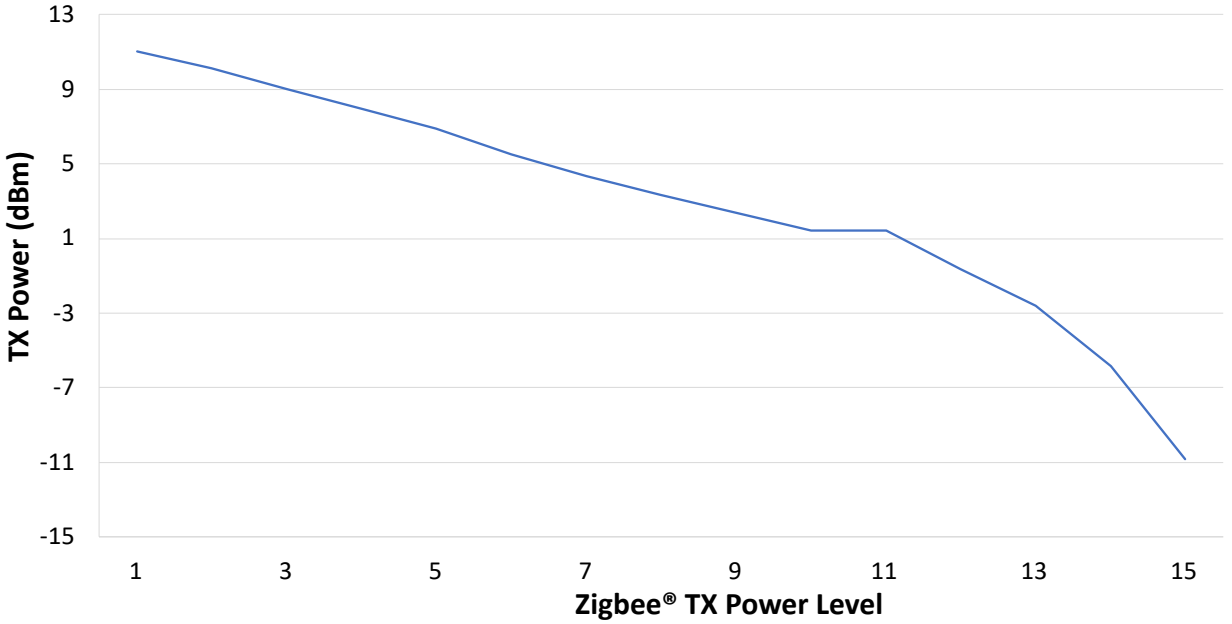


Figure 43-31. Zigbee Transmit Power vs. Frequency

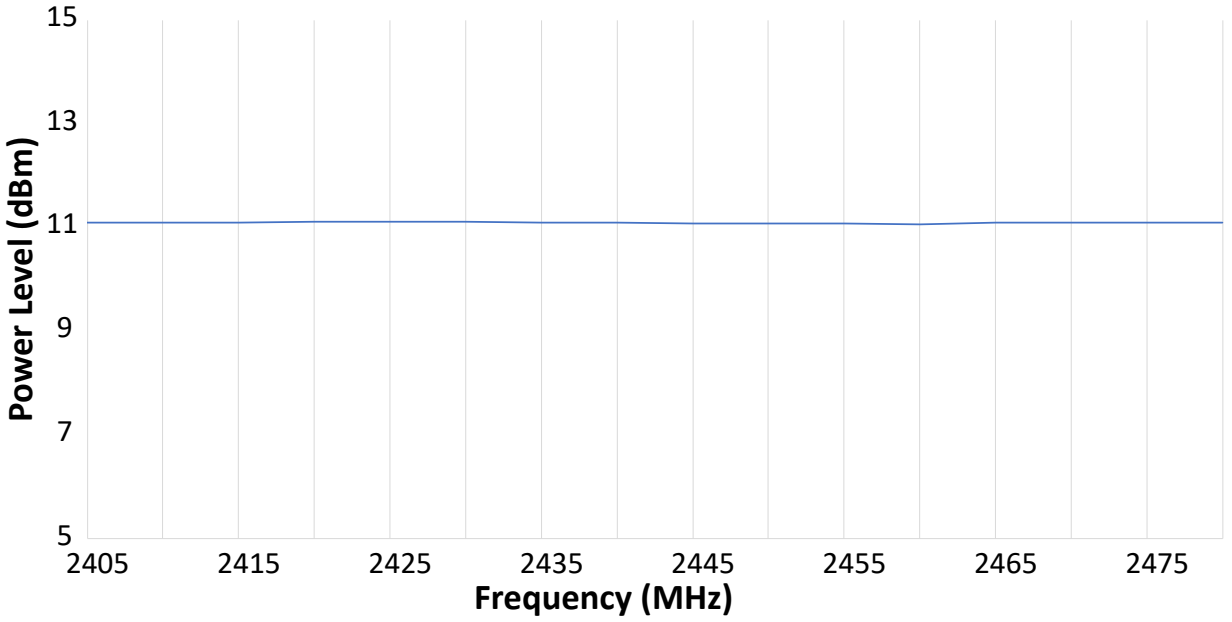


Figure 43-32. Zigbee Receive Sensitivity vs. Temperature

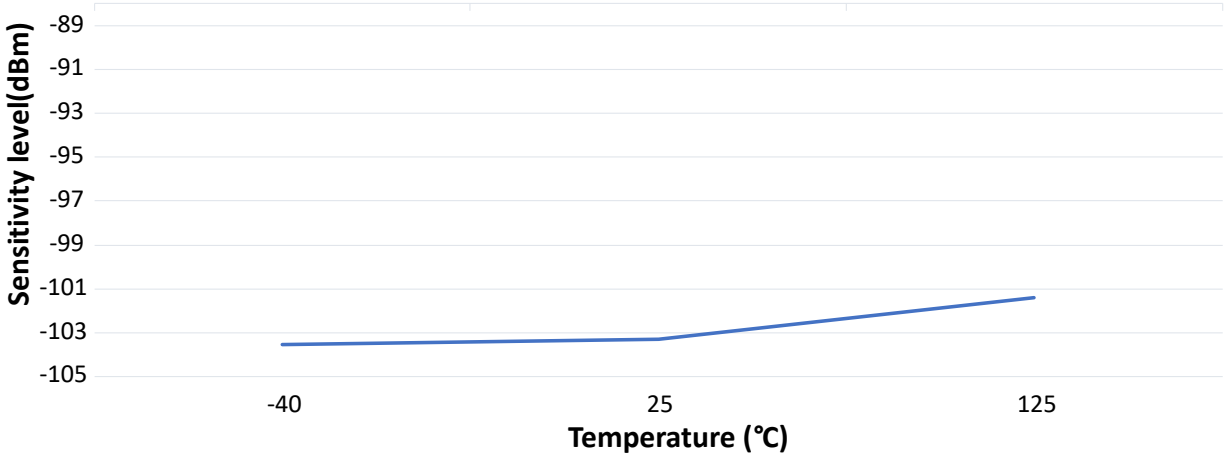


Figure 43-33. WBZ35x Module Zigbee Receive Sensitivity vs. Temperature

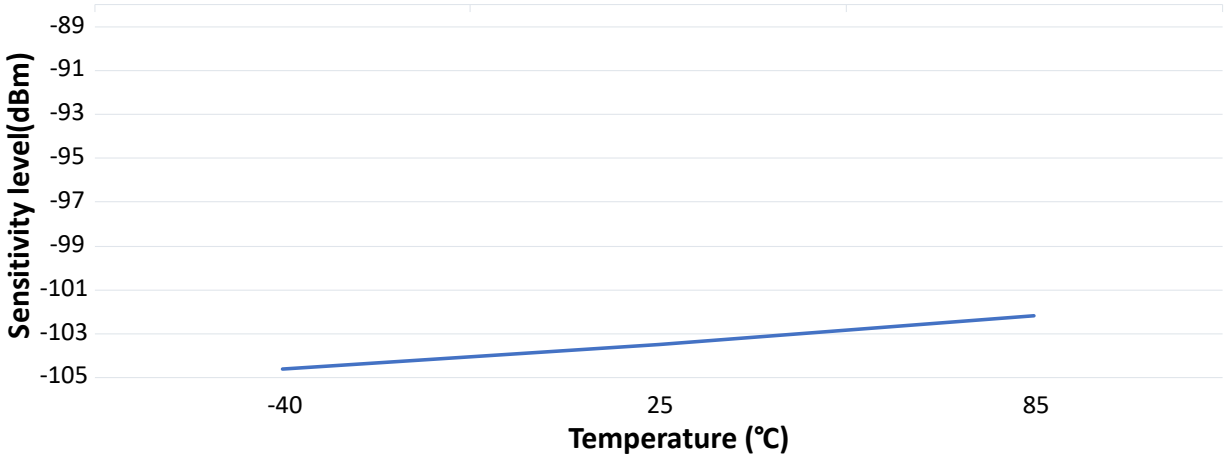


Figure 43-34. Zigbee Receive Sensitivity vs. Frequency

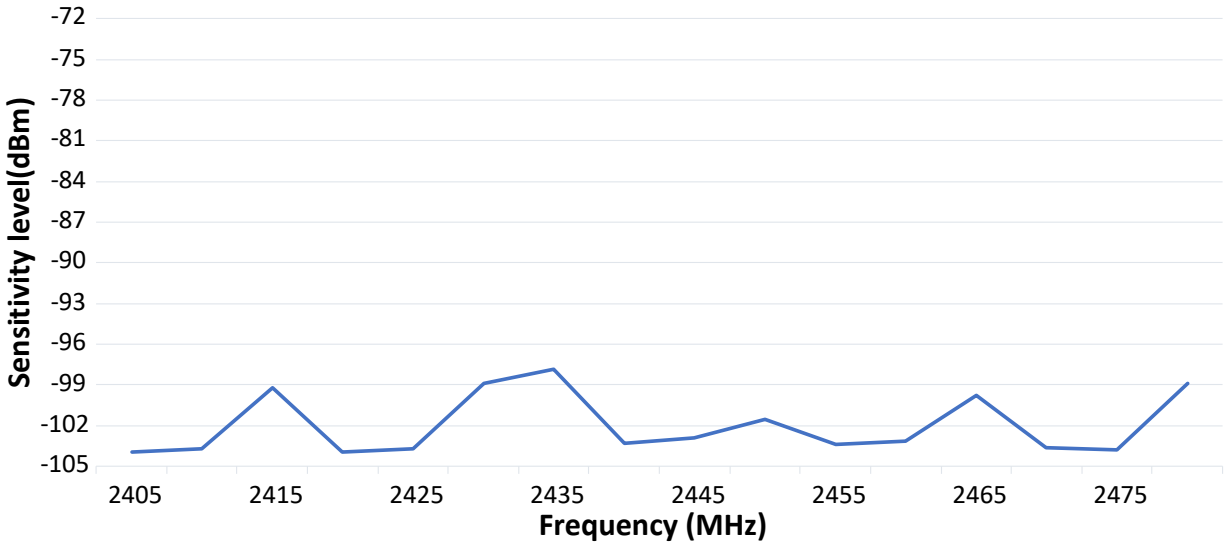


Figure 43-35. WBZ35x Module Zigbee Receive Sensitivity vs. Frequency

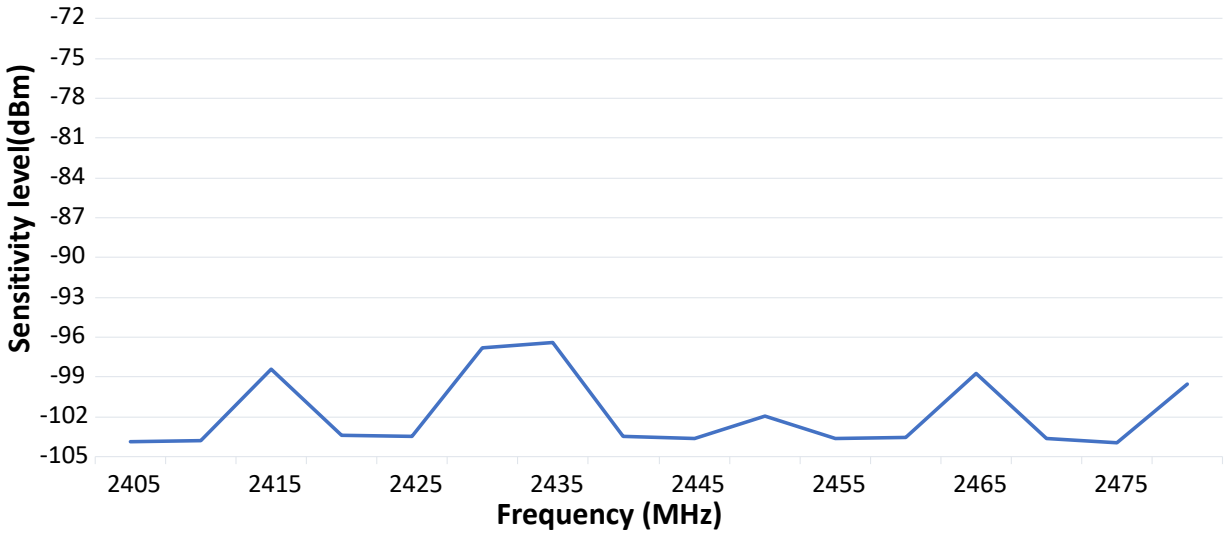


Figure 43-36. Zigbee Transmit Power vs. VDD Supply Voltage

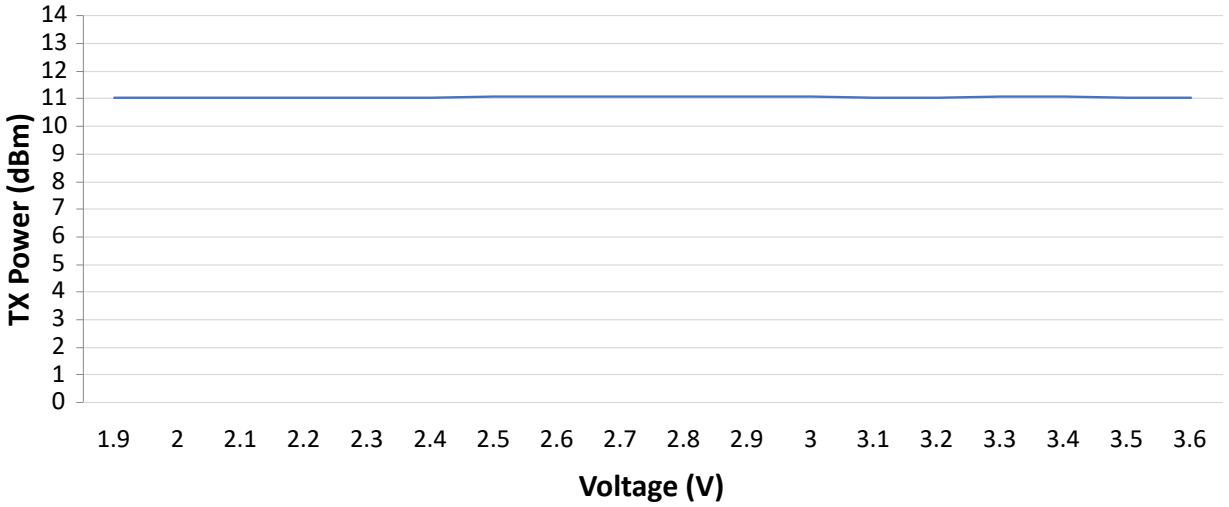


Figure 43-37. Zigbee Receive Sensitivity vs. VDD Supply Voltage

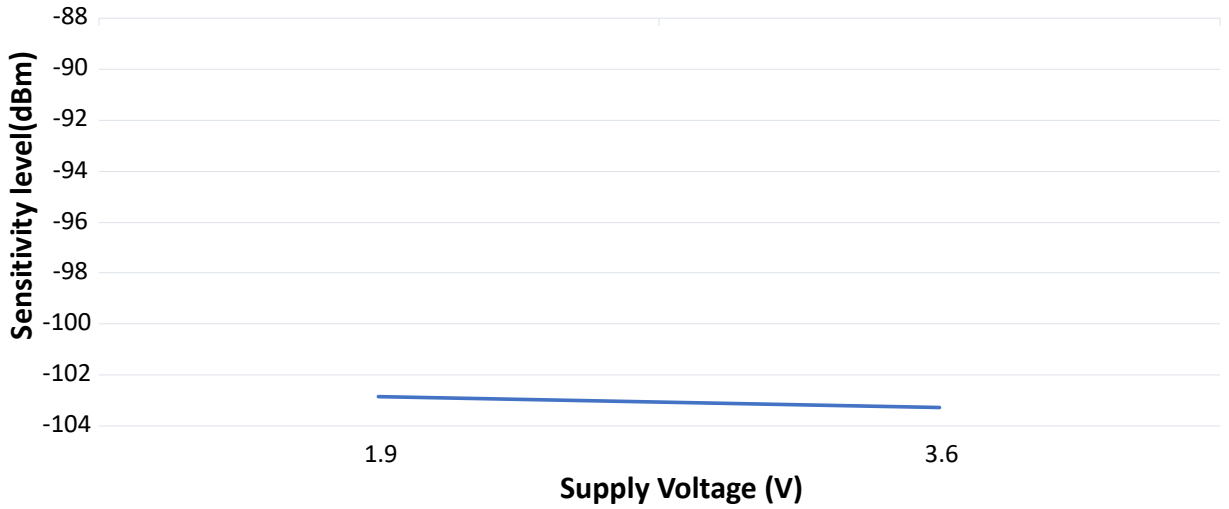


Figure 43-38. Zigbee ACR +-5M Margin

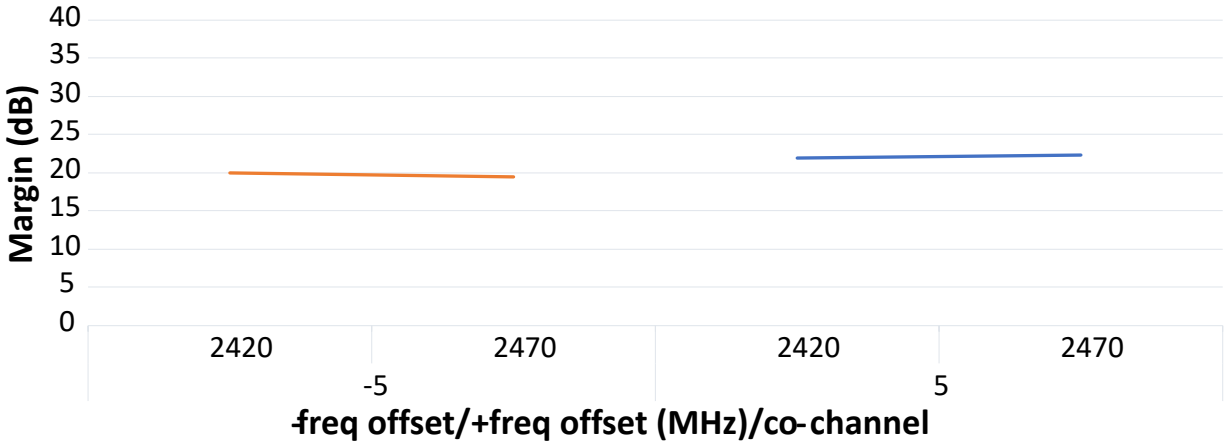


Figure 43-39. Zigbee ACR +-10M Margin

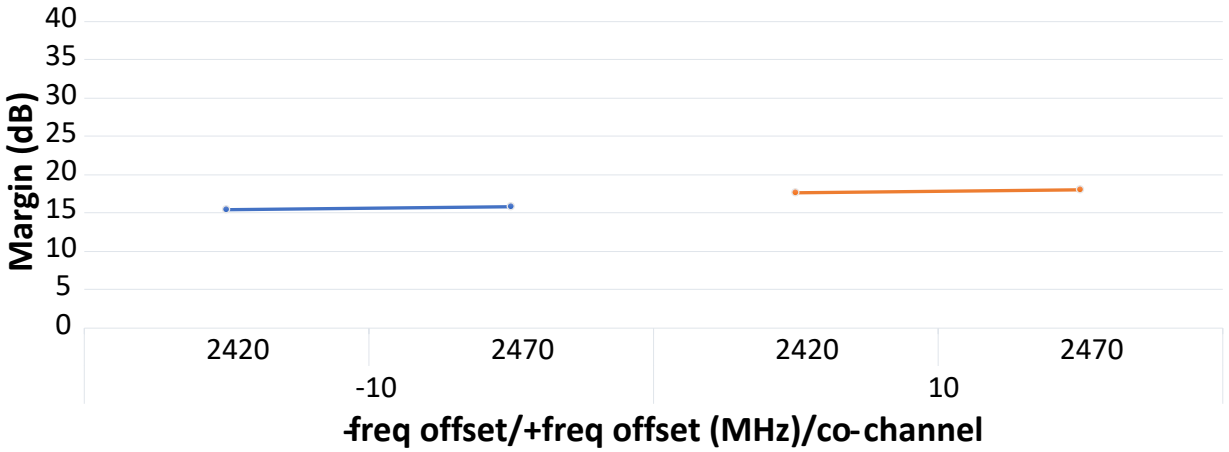


Figure 43-40. Zigbee Transmit Current vs. Temperature

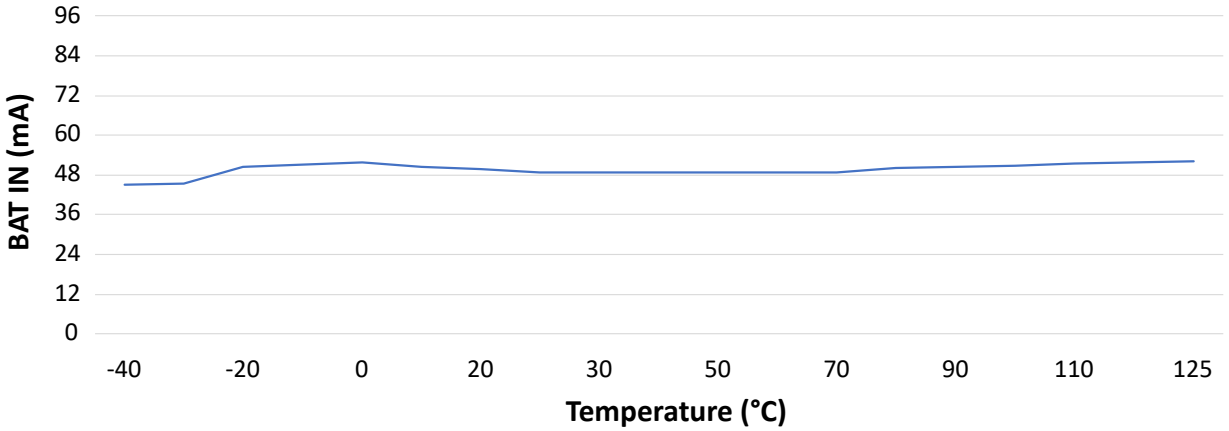
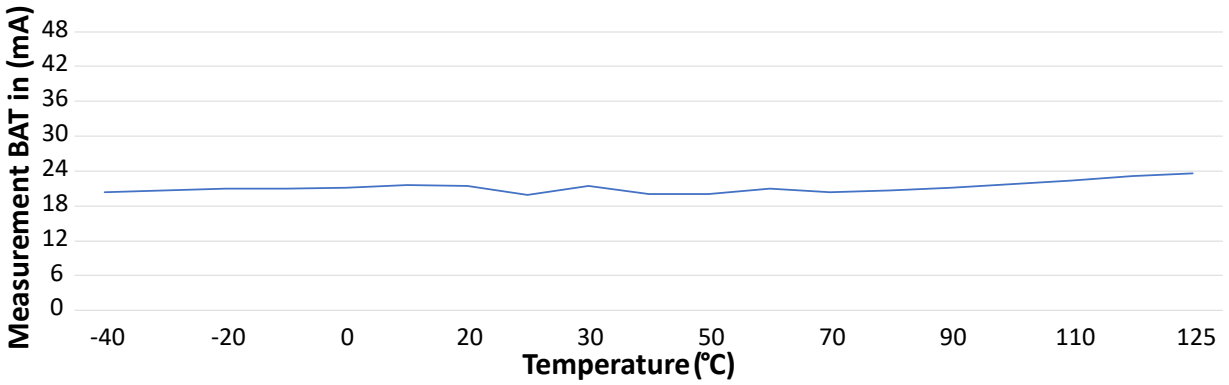


Figure 43-41. Zigbee Receive Current vs. Temperature



44. Packaging Information

This chapter provides information on package markings, dimension and footprint of the PIC32CX-BZ3 and the WBZ35x family.

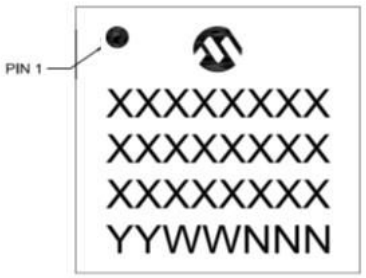
44.1 PIC32CX-BZ3 SoC Packaging Information

For the most current package drawings, see the Microchip Packaging Specification located at www.microchip.com/en-us/support/package-drawings.

44.1.1 PIC32CX-BZ3 SoC Package Marking

Figure 44-1. PIC32CX5109BZ31048 Package Marking

48L VQFN-WFS 6x6

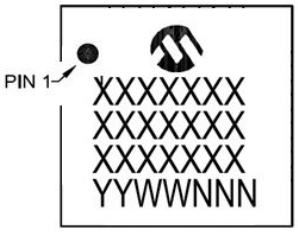


Example

Legend: XX...X Customer-specific information **Year code (last digit of calendar year)** **YY Year code (last 2 digits of calendar year)** **WW Week code (week of January 1 is week '01')** **NNN Alphanumeric traceability code** **Pb-free JEDEC designator for Matte Tin (Sn)** **This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.**
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Figure 44-2. PIC32CX5109BZ31032 Package Marking

32L VQFN-WFS 5x5



Example

Legend: XX...X Customer-specific information **Year code (last digit of calendar year)** **YY Year code (last 2 digits of calendar year)** **WW Week code (week of January 1 is week '01')** **NNN Alphanumeric traceability code** **Pb-free JEDEC designator for Matte Tin (Sn)** **This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.**
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

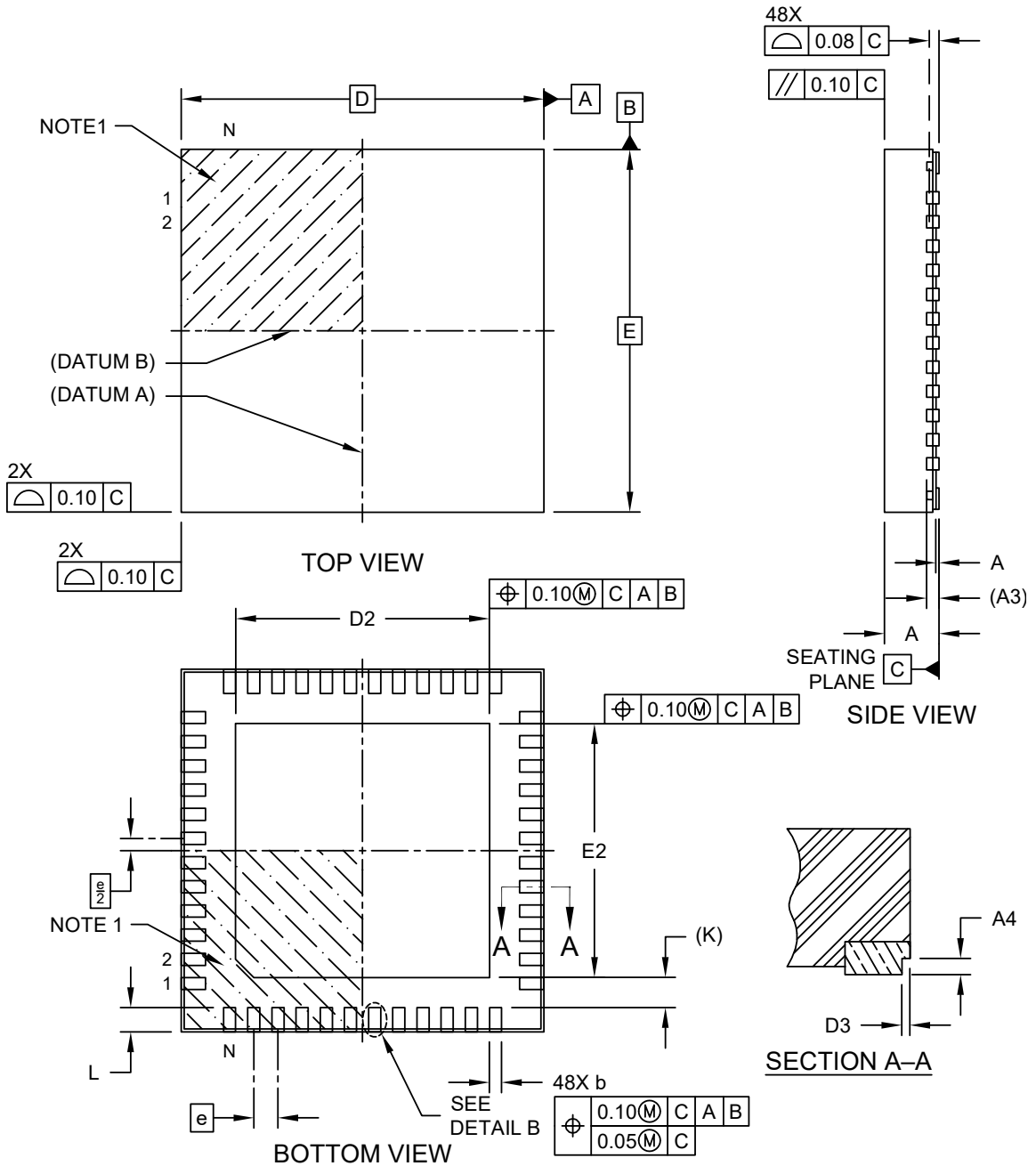
44.1.2 PIC32CX-BZ3 SoC Packaging Dimension

This section provides the package dimension details of the PIC32CX-BZ3 SoC.

44.1.2.1 PIC32CX5109BZ31048 SoC Packaging Dimension

48-Lead Very Thin Plastic Quad Flat, No Lead Package (ZWX) - 6x6x0.9 mm Body [VQFN] With 4.20 mm Exposed Pad and Stepped Wettable Flanks

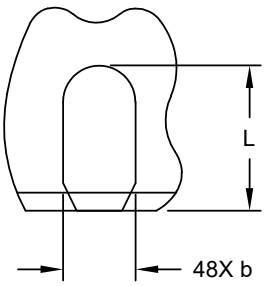
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



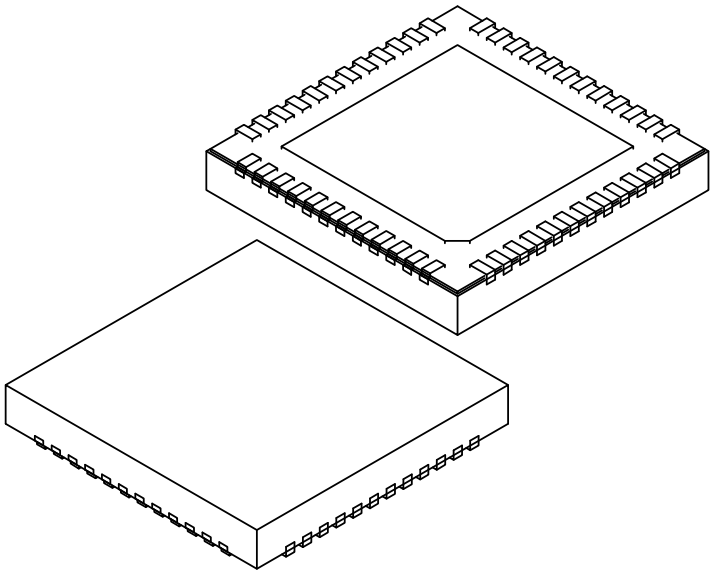
Microchip Technology Drawing C04-531 Rev B Sheet 1 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (ZWX) - 6x6x0.9 mm Body [VQFN] With 4.20 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



DETAIL B
ALTERNATE TERMINAL CONFIGURATION



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		48	
Pitch	e		0.40 BSC	
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.203 REF	
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	4.10	4.20	4.30
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	4.10	4.20	4.30
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K		0.50 REF	
Wettable Flank Step Cut Length	D3	-	-	0.085
Wettable Flank Step Cut Height	A4	0.10	-	0.19

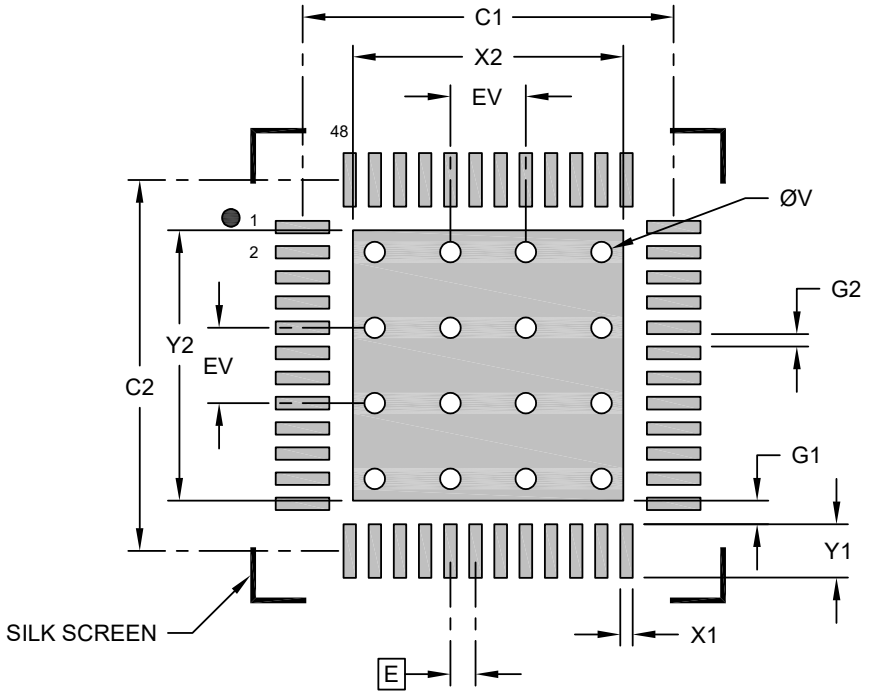
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-531 Rev B Sheet 2 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (ZWX) - 6x6x0.9 mm Body [VQFN] With 4.20 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Center Pad Width	X2			4.30
Center Pad Length	Y2			4.30
Contact Pad Spacing	C1		5.90	
Contact Pad Spacing	C2		5.90	
Contact Pad Width (Xnn)	X1			0.20
Contact Pad Length (Xnn)	Y1			0.85
Contact Pad to Center Pad (Xnn)	G1	0.38		
Contact Pad to Contact Pad (Xnn)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

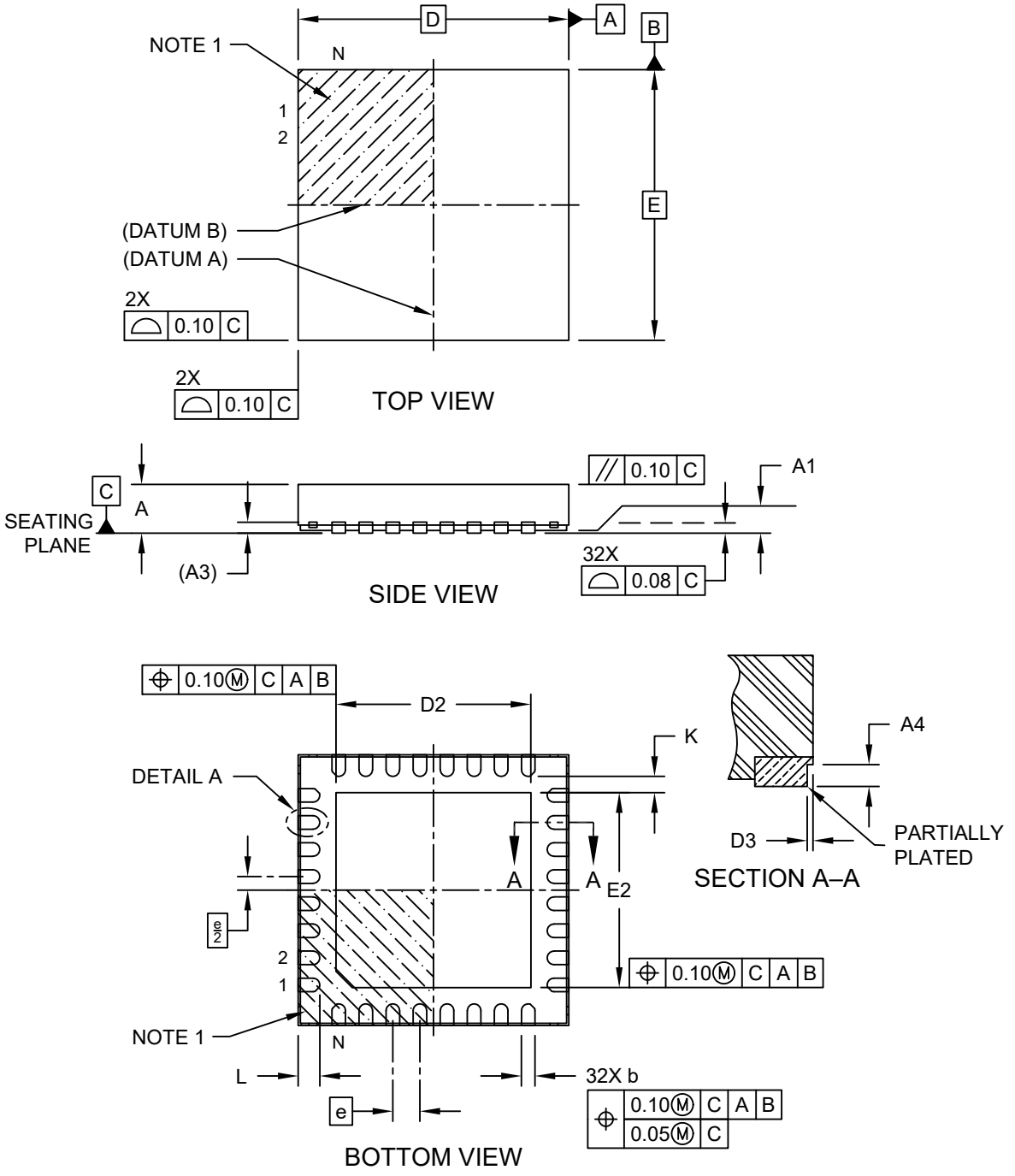
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2531 Rev B

44.1.2.2 PIC32CX5109BZ31032 SoC Packaging Dimension

32-Lead Very Thin Plastic Quad Flat, No Lead Package (RTB) - 5x5 mm Body [VQFN] With 3.6x3.6 mm Exposed Pad and Stepped Wettable Flanks

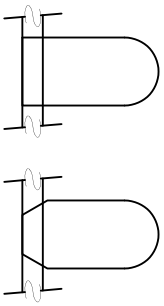
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



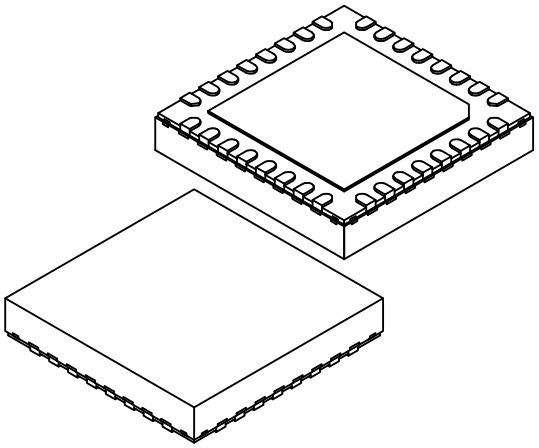
Microchip Technology Drawing C04-21391 Rev H Sheet 1 of 2

32-Lead Very Thin Plastic Quad Flat, No Lead Package (RTB) - 5x5 mm Body [VQFN] With 3.6x3.6 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



DETAIL 1
ALTERNATE TERMINAL CONFIGURATIONS



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		32	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.035	0.05
Terminal Thickness	A3		0.203 REF	
Overall Length	D		5.00 BSC	
Exposed Pad Length	D2	3.50	3.60	3.70
Overall Width	E		5.00 BSC	
Exposed Pad Width	E2	3.50	3.60	3.70
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-
Wettable Flank Step Cut Width	D3	-	-	0.085
Wettable Flank Step Cut Depth	A4	0.10	-	0.19

Dimensions D3 and A4 above apply to all new products released after November 1, and all products shipped after January 1, 2019, and supersede dimensions D3 and A4 below.
No physical changes are being made to any package; this update is to align cosmetic and tolerance variations from existing suppliers.

Wettable Flank Step Length	D3	0.035	0.06	0.085
Wettable Flank Step Height	A4	0.10	-	0.19

Notes:

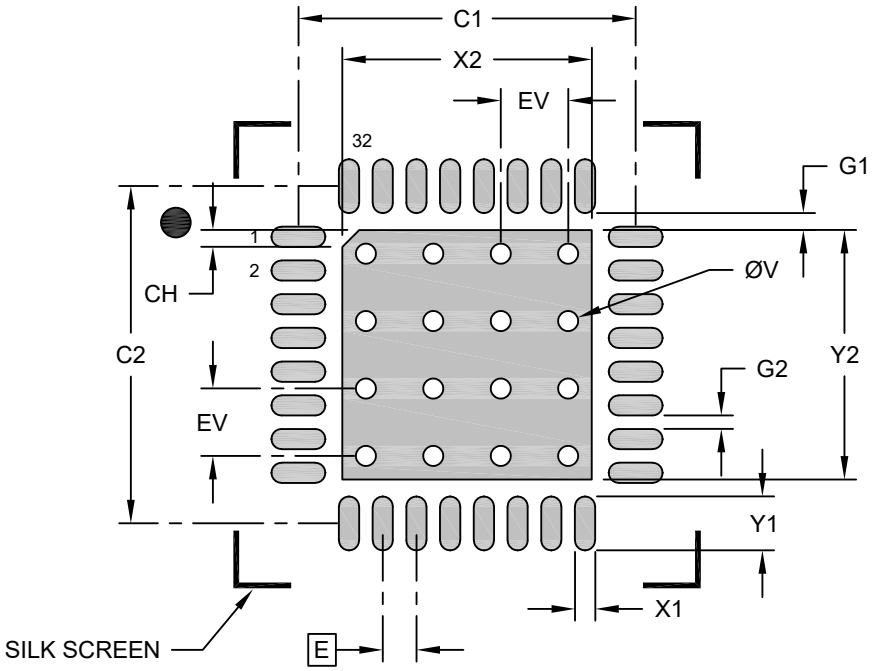
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21391 Rev H Sheet 2 of 2

32-Lead Very Thin Plastic Quad Flat, No Lead Package (RTB) - 5x5 mm Body [VQFN] With 3.6x3.6 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	X2			3.70
Optional Center Pad Length	Y2			3.70
Exposed Pad 45° Corner Chamfer	CH		0.25	
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X32)	X1			0.30
Contact Pad Length (X32)	Y1			0.80
Contact Pad to Center Pad (X32)	G1	0.25		
Contact Pad to Contact Pad (X28)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23391 Rev H

44.2 WBZ35x Module Packaging Information

44.2.1 WBZ35x Module Packaging Marking

Figure 44-3. WBZ35x Module Packaging Marking

Legend:

XX....X	Module part number and version and regulatory designator
YY:	Year code (last 2 digits of calendar year)
WW	Week code (week of January 1 is week "01")
NNN	Alphanumeric traceability code

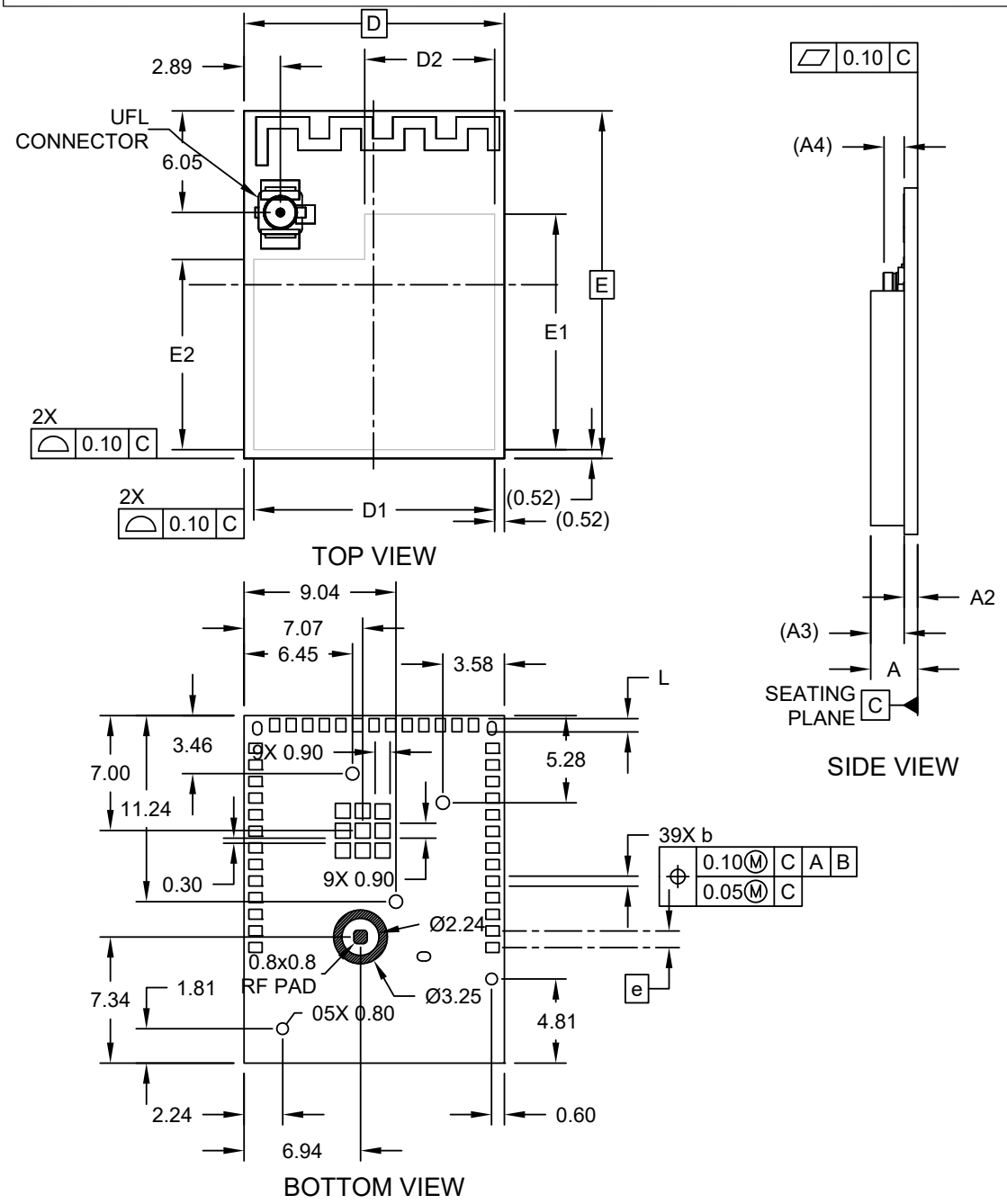
44.2.2 WBZ35x Module Packaging Dimension

This section provides the package dimension details of the WBZ35x Module.

44.2.2.1 WBZ351 Module Packaging Dimension

39-Lead PCB Module (2XW) - 15.5x20.7x2.8mm [MODULE] for WBZ351 With Metal Shield and Coaxial Connector

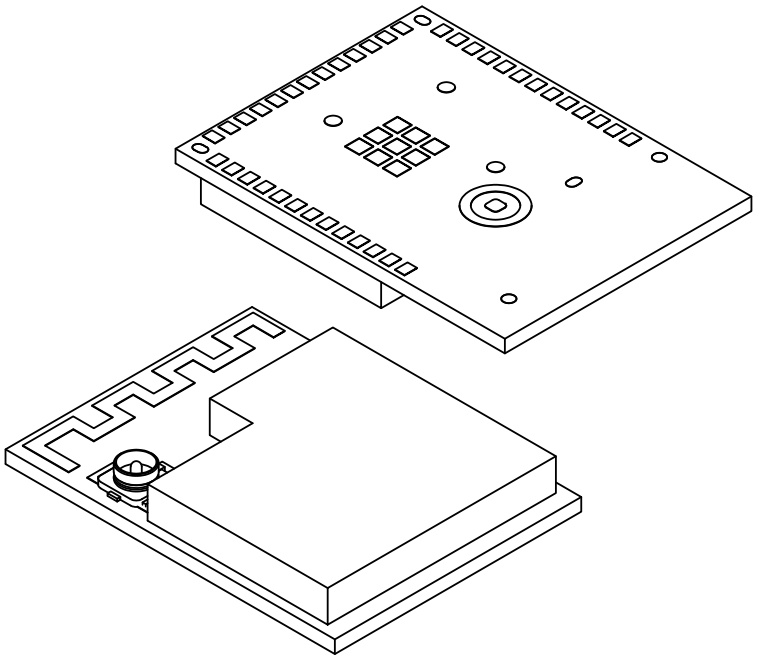
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-10054 Rev A Sheet 1 of 2

39-Lead PCB Module (2XW) - 15.5x20.7x2.8mm [MODULE] for WBZ351 With Metal Shield and Coaxial Connector

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		39	
Terminal Pitch	e		1.00 BSC	
Overall Height	A	2.70	2.80	2.90
PCB Thickness	A2	0.70	0.80	0.90
Shield Height	A3		2.00 REF	
UFL Connector Height	A4		1.25 REF	
Overall Length	D		15.50 BSC	
Overall Width	E		20.70 BSC	
Shield Length	D1	14.25	14.35	14.45
Shield Length	D2	7.64	7.74	7.84
Shield Width	E1	13.93	14.03	14.13
Shield Width	E2	11.23	11.33	11.43
Terminal Width	b	0.50	0.60	0.70
Terminal Length	L	0.70	0.80	0.90

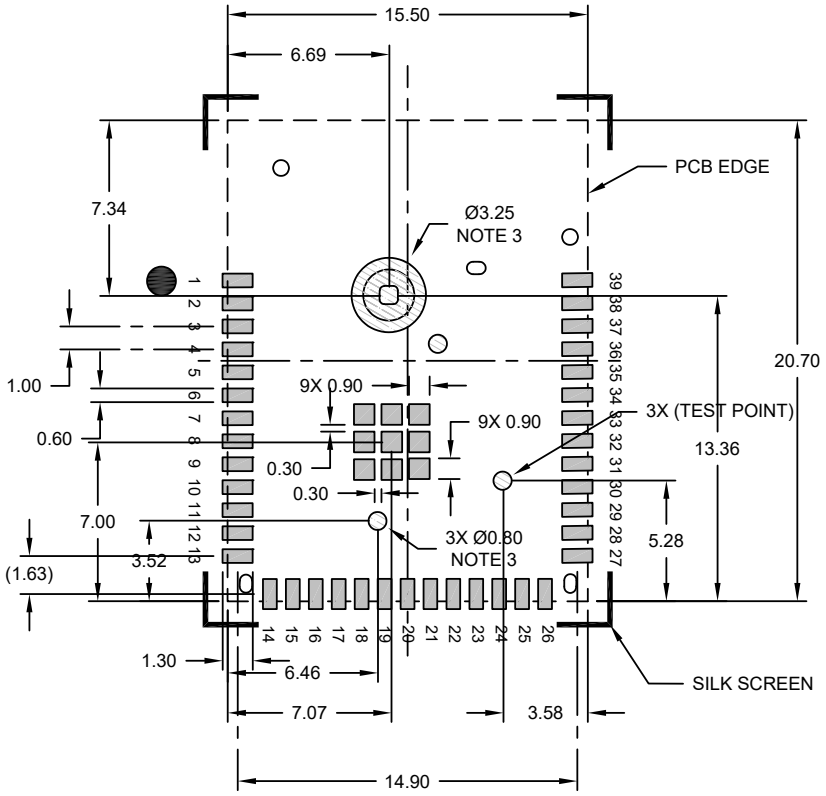
Notes:

1. All Dimensions are in Millimeters
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-10054 Rev A Sheet 2 of 2

39-Lead PCB Module (2XW) - 15.5x20.7x2.8mm [MODULE] for WBZ351 With Metal Shield and Coaxial Connector

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



COPPER KEEPOUT ZONE

RECOMMENDED LAND PATTERN

Notes:

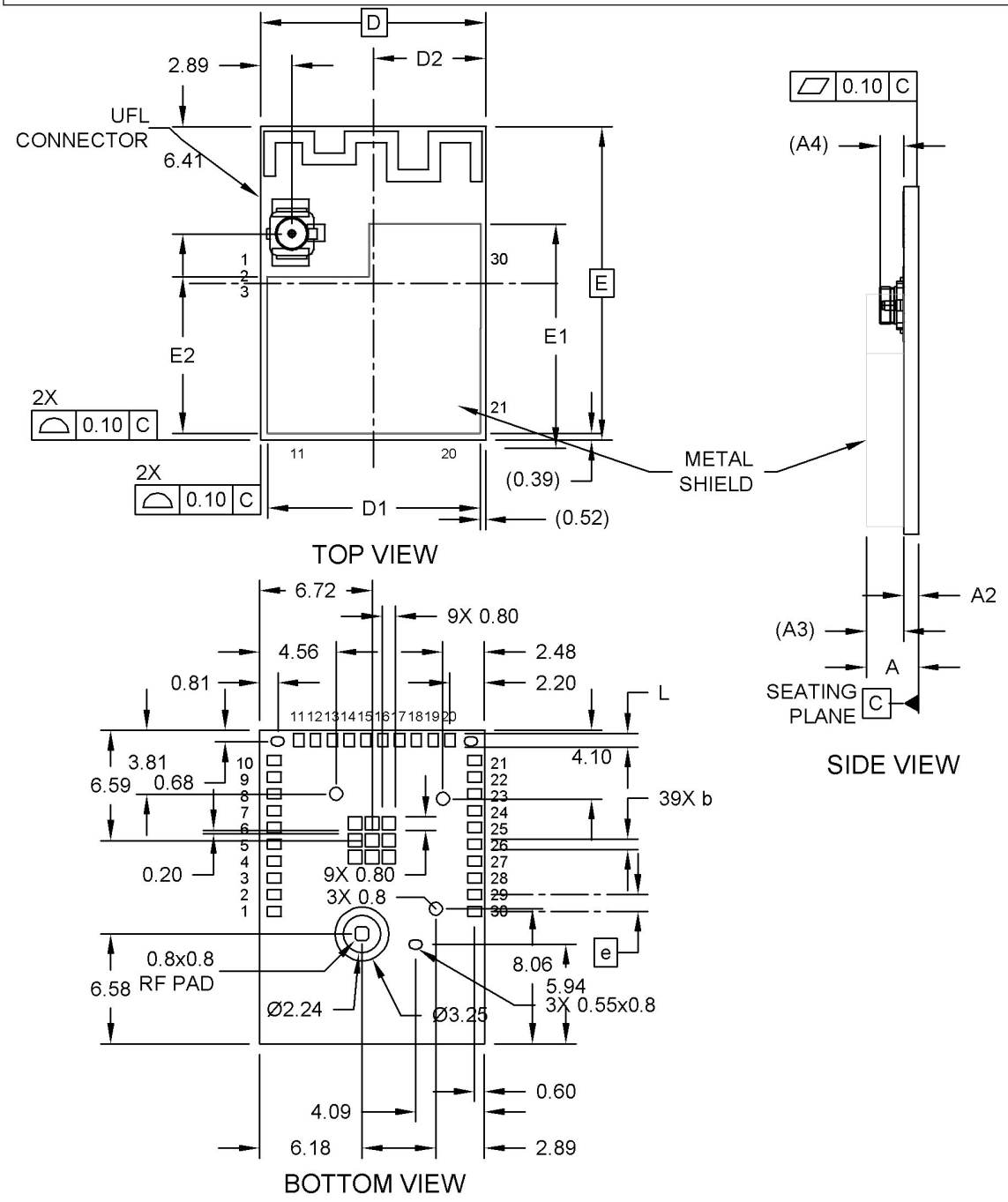
1. All dimensions are in millimeters.
2. Keep this area free from all metal, including ground flll.
3. Keep these areas free from routes and exposed copper. Ground fill with solder mask may be placed here.

Microchip Technology Drawing C04-12054 Rev A

44.2.2.2 WBZ350 Module Packaging Dimension

30-Lead PCB Module (3BW) - 13.4x18.7x2.8mm [MODULE]
With Metal Shield and Coaxial Connector

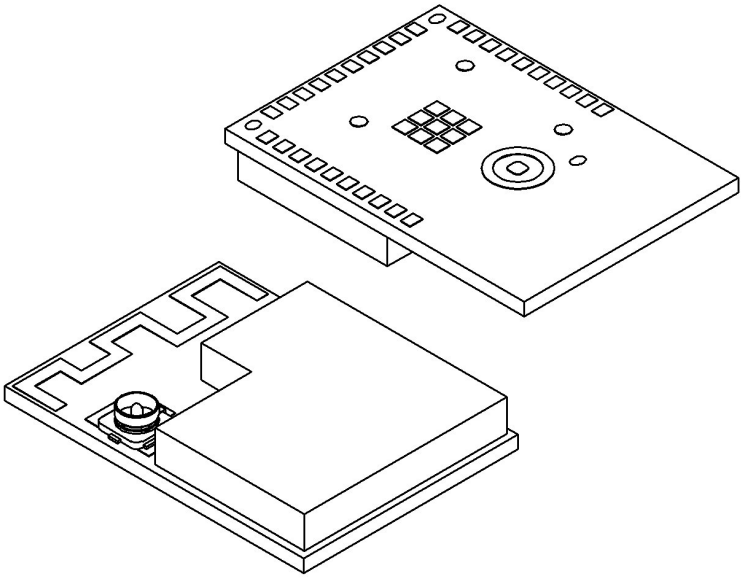
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-10055 Rev B Sheet 1 of 2

30-Lead PCB Module (3BW) - 13.4x18.7x2.8mm [MODULE]
With Metal Shield and Coaxial Connector

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		39	
Terminal Pitch	e		1.00 BSC	
Overall Height	A	2.70	2.80	2.90
PCB Thickness	A2	0.70	0.80	0.90
Shield Height	A3		2.00 REF	
UFL Connector Height	A4		1.25 REF	
Overall Length	D		13.40 BSC	
Overall Width	E		18.70 BSC	
Shield Length	D1	12.14	12.24	12.34
Shield Length	D2	6.60	6.70	6.80
Shield Width	E1	12.14	12.24	12.34
Shield Width	E2	8.92	9.02	9.12
Terminal Width	b	0.50	0.60	0.70
Terminal Length	L	0.70	0.80	0.90

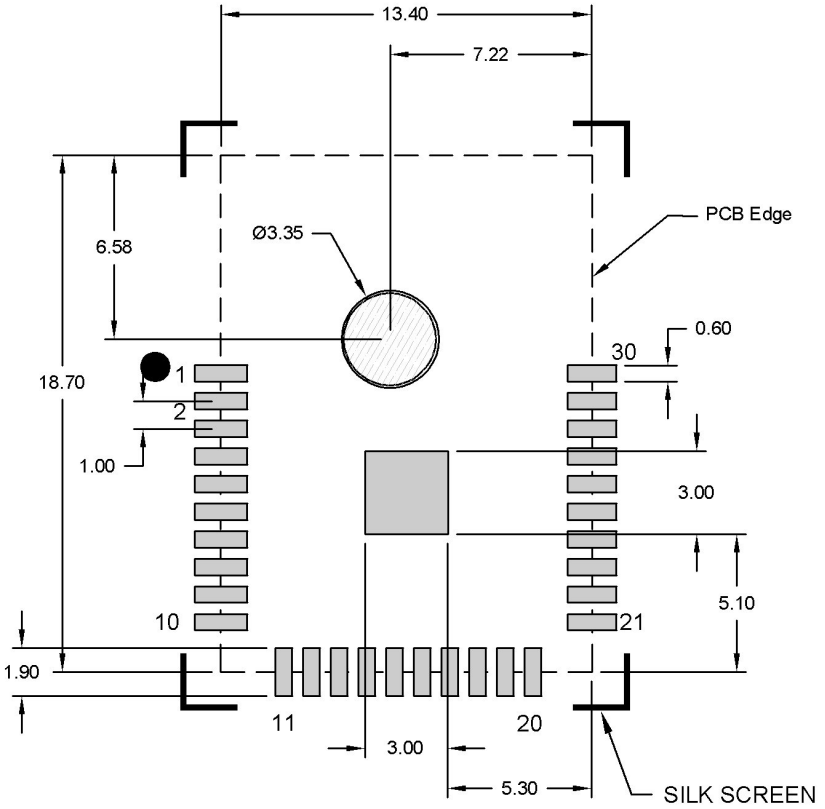
Notes:

1. All Dimensions are in Millimeters
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-10055 Rev B Sheet 2 of 2

30-Lead PCB Module (3BW) - 13.4x18.7x2.8mm [MODULE] With Metal Shield and Coaxial Connector

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

COPPER KEEPOUT ZONE

Notes:

1. All dimensions are in millimeters.
2. Keep this area free from all metal, including ground flll.
3. Keep these areas free from routes and exposed copper. Ground fill with solder mask may be placed here.

Microchip Technology Drawing C04-12055 Rev B

45. Appendix A: Regulatory Approval

The WBZ350PC module has received regulatory approval for the following countries:

- Bluetooth® Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ350
- Canada/ISED:
  - IC: 20266-WBZ350
  - HVIN: WBZ350PC
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC

The WBZ350PE module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ350
- Canada/ISED:
  - IC: 20266-WBZ350
  - HVIN: WBZ350PE
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC
- China: 24J999P60003

The WBZ350UC module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ350
- Canada/ISED:
  - IC: 20266-WBZ350
  - HVIN: WBZ350UC
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC

The WBZ350UE module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ350
- Canada/ISED:
  - IC: 20266-WBZ350
  - HVIN: WBZ350PE
- Europe/CE
- Japan/MIC

- Korea/KCC
- Taiwan/NCC
- China: 24J999P6000324J999P60004(M)

The WBZ351PC module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ351
- Canada/ISED:
  - IC: 20266-WBZ351
  - HVIN: WBZ351PC
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC

The WBZ351PE module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ351
- Canada/ISED:
  - IC: 20266-WBZ351
  - HVIN: WBZ351PE
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC
- China: 2023DJ14927

The WBZ351UC module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ351
- Canada/ISED:
  - IC: 20266-WBZ351
  - HVIN: WBZ351UC
- Europe/CE
- Japan/MIC
- Korea/KCC
- Taiwan/NCC

The WBZ351UE module has received regulatory approval for the following countries:

- Bluetooth Special Interest Group (SIG) QDID: 176346
- United States/FCC ID: 2ADHKWBZ351
- Canada/ISED:
  - IC: 20266-WBZ351
  - HVIN: WBZ351UE
- Europe/CE

- Japan/MIC
- Korea/KCC
- Taiwan/NCC
- China: 2023DJ2471

45.1 United States

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have received Federal Communications Commission (FCC) CFR47 Telecommunications, Part 15 Subpart C “Intentional Radiators” single-modular approval in accordance with Part 15.212 Modular Transmitter approval. Single-modular transmitter approval is defined as a complete RF transmission sub-assembly, designed to be incorporated into another device, that must demonstrate compliance with FCC rules and policies independent of any host. A transmitter with a modular grant can be installed in different end-use products (referred to as a host, host product or host device) by the grantee or other equipment manufacturer, then the host product may not require additional testing or equipment authorization for the transmitter function provided by that specific module or limited module device.

The user must comply with all of the instructions provided by the Grantee, which indicate installation and/or operating conditions necessary for compliance.

A host product itself is required to comply with all other applicable FCC equipment authorization regulations, requirements, and equipment functions that are not associated with the transmitter module portion. For example, compliance must be demonstrated: to regulations for other transmitter components within a host product; to requirements for unintentional radiators (Part 15 Subpart B), such as digital devices, computer peripherals, radio receivers, etc.; and to additional authorization requirements for the non-transmitter functions on the transmitter module (i.e., Suppliers Declaration of Conformity (SDoC) or certification) as appropriate (e.g., Bluetooth and Wi-Fi transmitter modules may also contain digital logic functions).

45.1.1 Labeling and User Information Requirements

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have been labeled with its own FCC ID number, and if the FCC ID is not visible when the module is installed inside another device, then the outside of the finished product into which the module is installed must display a label referring to the enclosed module. This exterior label must use the following wording:

For the WBZ350PC/WBZ350PE/ WBZ350UC/WBZ350UE module	Contains Transmitter Module FCC ID: 2ADHKWBZ350 or Contains FCC ID: 2ADHKWBZ350 **This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation**.
For the WBZ351PC/WBZ351PE/ WBZ351UC/WBZ351UE module	Contains Transmitter Module FCC ID: 2ADHKWBZ351 or Contains FCC ID: 2ADHKWBZ351 **This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation**.

The user's manual for the finished product must include the following statement:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

Additional information on labeling and user information requirements for Part 15 devices can be found in KDB Publication 784748, which is available at the FCC Office of Engineering and Technology (OET) Laboratory Division Knowledge Database (KDB) apps.fcc.gov/oetcf/kdb/index.cfm.

45.1.2 RF Exposure

All transmitters regulated by FCC must comply with RF exposure requirements. KDB 447498 General RF Exposure Guidance provides guidance in determining whether proposed or existing transmitting facilities, operations or devices comply with limits for human exposure to Radio Frequency (RF) fields adopted by the Federal Communications Commission (FCC).

From the FCC Grant: Output power listed is conducted. This grant is valid only when the module is sold to OEM integrators and must be installed by the OEM or OEM integrators. This transmitter is restricted for use with the specific antenna(s) tested in this application for Certification and must not be co-located or operating in conjunction with any other antenna or transmitters within a host device, except in accordance with FCC multi-transmitter product procedures.

WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE: These modules are approved for installation into mobile or/and portable host platforms.

45.1.3 Helpful Web Sites

- Federal Communications Commission (FCC): www.fcc.gov.
- FCC Office of Engineering and Technology (OET) Laboratory Division Knowledge Database (KDB) apps.fcc.gov/oetcf/kdb/index.cfm.

45.2 Canada

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have been certified for use in Canada under Innovation, Science and Economic Development Canada (ISED, formerly Industry Canada) Radio Standards Procedure (RSP) RSP-100, Radio Standards Specification (RSS) RSS-Gen and RSS-247. Modular approval permits the installation of a module in a host device without the need to recertify the device.

45.2.1 Labeling and User Information Requirements

Labeling Requirements (from RSP-100 - Issue 12, Section 5): The host product shall be properly labeled to identify the module within the host device.

The Innovation, Science and Economic Development Canada certification label of a module shall be clearly visible at all times when installed in the host device; otherwise, the host product must be labeled to display the Innovation, Science and Economic Development Canada certification number of the module, preceded by the word “Contains” or similar wording expressing the same meaning, as follows:

For the WBZ350PC/WBZ350PE/ WBZ350UC/WBZ350UE module	**Contains IC: 20266-WBZ350**
For the WBZ351PC/WBZ351PE/ WBZ351UC/WBZ351UE module	**Contains IC: 20266-WBZ351**

User Manual Notice for License-Exempt Radio Apparatus (from Section 8.4 RSS-Gen, Issue 5, February 2021): User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both:

This device contains license-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's license-exempt RSS(s). Operation is subject to the following two conditions:

(1) This device may not cause interference;

(2) This device must accept any interference, including interference that may cause undesired operation of the device.

L'émetteur/récepteur exempt de licence contenu dans le présent appareil est conforme aux CNR d'Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:

1. L'appareil ne doit pas produire de brouillage;

2. L'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Transmitter Antenna (From Section 6.8 RSS-GEN, Issue 5, February 2021): User manuals, for transmitters shall display the following notice in a conspicuous location:

This radio transmitter IC: 20266-WBZ350 and IC: 20266-WBZ351 have been approved by Innovation, Science and Economic Development Canada to operate with the antenna types listed below, with the maximum permissible gain indicated. Antenna types not included in this list that have a gain greater than the maximum gain indicated for any type listed are strictly prohibited for use with this device.

Le présent émetteur radio IC: 20266-WBZ350 and IC: 20266-WBZ351 a été approuvé par Innovation, Sciences et Développement économique Canada pour fonctionner avec les types d'antenne énumérés cidessous et ayant un gain admissible maximal. Les types d'antenne non inclus dans cette liste, et dont le gain est supérieur au gain maximal indiqué pour tout type figurant sur la liste, sont strictement interdits pour l'exploitation de l'émetteur.

Immediately following the above notice, the manufacturer shall provide a list of all antenna types approved for use with the transmitter, indicating the maximum permissible antenna gain (in dBi) and required impedance for each.

45.2.2 RF Exposure

All transmitters regulated by Innovation, Science and Economic Development Canada (ISED) must comply with RF exposure requirements listed in RSS-102 - Radio Frequency (RF) Exposure Compliance of Radio communication Apparatus (All Frequency Bands).

This transmitter is restricted for use with a specific antenna tested in this application for certification, and must not be co-located or operating in conjunction with any other antenna or transmitters within a host device, except in accordance with Canada multi-transmitter product procedures.

WBZ350 and WBZ351: The devices operate at an output power level which is within the ISED SAR test exemption limits at any user distance.

45.2.3 Helpful Web Sites

Innovation, Science and Economic Development Canada (ISED): www.ic.gc.ca/.

45.3 Europe

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules is/are a Radio Equipment Directive (RED) assessed radio module that is CE marked and has been manufactured and tested with the intention of being integrated into a final product.

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules has/have been tested to RED 2014/53/EU Essential Requirements mentioned in the following European Compliance table.

Table 45-1. European Compliance Information

Certification	Standard	Article
Safety	EN 62368	3.1a
Health	EN 62311	
EMC	EN 301 489-1	3.1b
	EN 301 489-17	
Radio	EN 300 328	3.2

The ETSI provides guidance on modular devices in the "*Guide to the application of harmonised standards covering articles 3.1b and 3.2 of the RED 2014/53/EU (RED) to multi-radio and combined radio and non-radio equipment*" document available at http://www.etsi.org/deliver/etsi_eg/203300_203399/20 3367/01.01.01_60/eg_203367v010101p.pdf.

Note: To maintain conformance to the standards listed in the preceding European Compliance table, the module shall be installed in accordance with the installation instructions in this data sheet and shall not be modified. When integrating a radio module into a completed product, the integrator becomes the manufacturer of the final product and is therefore responsible for demonstrating compliance of the final product with the essential requirements against the RED.

45.3.1 Labeling and User Information Requirements

The label on the final product that contains the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules must follow CE marking requirements.

45.3.2 Conformity Assessment

From ETSI Guidance Note EG 203367, section 6.1, when non-radio products are combined with a radio product:

If the manufacturer of the combined equipment installs the radio product in a host non-radio product in equivalent assessment conditions (i.e. host equivalent to the one used for the assessment of the radio product) and according to the installation instructions for the radio product, then no additional assessment of the combined equipment against article 3.2 of the RED is required.

45.3.2.1 Simplified EU Declaration of Conformity

Hereby, Microchip Technology Inc. declares that the radio equipment type WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE is in compliance with Directive 2014/53/EU.

The full text of the EU declaration of conformity, for this product, is available at www.microchip.com/design-centers/wireless-connectivity/.

45.3.3 Helpful Websites

A document that can be used as a starting point in understanding the use of Short Range Devices (SRD) in Europe is the European Radio Communications Committee (ERC) Recommendation 70-03 E, which can be downloaded from the European Communications Committee (ECC) at: http://www.ecodocdb.dk/.

Additional helpful web sites are:

- Radio Equipment Directive (2014/53/EU): https://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/red_en
- European Conference of Postal and Telecommunications Administrations (CEPT): http://www.cept.org
- European Telecommunications Standards Institute (ETSI): http://www.etsi.org
- The Radio Equipment Directive Compliance Association (REDCA):

http://www.redca.eu/

45.4 Japan

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have received type certification and is required to be labeled with its own technical conformity mark and certification number as required to conform to the technical standards regulated by the Ministry of Internal Affairs and Communications (MIC) of Japan pursuant to the Radio Act of Japan.

Integration of this module into a final product does not require additional radio certification provided installation instructions are followed and no modifications of the module are allowed. Additional testing may be required:

- If the host product is subject to electrical appliance safety (for example, powered from an AC mains), the host product may require Product Safety Electrical Appliance and Material (PSE) testing. The integrator should contact their conformance laboratory to determine if this testing is required
- There is an voluntary Electromagnetic Compatibility (EMC) test for the host product administered by VCCI: www.vcci.jp/vcci_e/index.html

45.4.1 Labeling and User Information Requirements

The label on the final product which contains the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module(s) must follow Japan marking requirements. The integrator of the module should refer to the labeling requirements for Japan available at the Ministry of Internal Affairs and Communications (MIC) website.

For the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules, due to a limited module size, the technical conformity logo and ID is displayed in the data sheet and/or packaging and cannot be displayed on the module label. The final product in which this module is being used must have a label referring to the type certified module inside:

WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE module

WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module

45.4.2 Helpful Web Sites

- Ministry of Internal Affairs and Communications (MIC): www.tele.soumu.go.jp/e/index.htm.
- Association of Radio Industries and Businesses (ARIB): www.arib.or.jp/english/.

45.5 Korea

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have received certification of conformity in accordance with the Radio Waves Act. Integration of this module into a final product does not require additional radio certification provided installation instructions are followed and no modifications of the module are allowed.

45.5.1 Labeling and User Information Requirements

The label on the final product which contains the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module(s) must follow KC marking requirements. The integrator of the module should refer to the labeling requirements for Korea available on the Korea Communications Commission (KCC) website.

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules are labeled with its own KC mark. The final product requires the KC mark and certificate number of the module:

WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE module

R-R-mcp-WBZ350

WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module

R-R-mcp-WBZ351

45.5.2 Helpful Websites

- Korea Communications Commission (KCC): www.kcc.go.kr.
- National Radio Research Agency (RRA): rra.go.kr.

45.6 Taiwan

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules have received compliance approval in accordance with the Telecommunications Act. Customers seeking to use the compliance approval in their product should contact Microchip Technology sales or distribution partners to obtain a Letter of Authority.

Integration of this module into a final product does not require additional radio certification provided installation instructions are followed and no modifications of the module are allowed.

45.6.1 Labeling and User Information Requirements

For the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules, due to the limited module size, the NCC mark and ID are displayed in the data sheet only and cannot be displayed on the module label:

CCAN24Y10262T0

WBZ350PC module

CCAN24Y10260T9

WBZ350PE module

WBZ350UC module

WBZ350UE module

WBZ351PC module

WBZ351PE module

WBZ351UC module

WBZ351UE module

The user's manual should contain following warning (for RF device) in traditional Chinese:

根據 NCC LP0002 低功率射頻器材技術規範_章節 3.8.2：

取得審驗證明之低功率射頻器材，非經核准，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。

低功率射頻器材之使用不得影響飛航安全及干擾合法通信；經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。

前述合法通信，指依電信管理法規定作業之無線電通信。

低功率射頻器材須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

此模組於取得認證後將依規定於模組本體標示審驗合格標籤，並要求平台廠商於平台上標示本產品內含發射器模組

45.6.2 Helpful Web Sites

National Communications Commission (NCC): www.ncc.gov.tw

45.7 China

The WBZ350PE/WBZ351PE modules has/have received certification of conformity in accordance with the China MIIT Notice 2014-01 of State Radio Regulation Committee (SRRC) certification scheme. Integration of this module into a final product does not require additional radio certification, provided installation instructions are followed and no modifications of the module are allowed. Refer to SRRC certificate available in WBZ350PE/WBZ351PE product page for expiry date.

The WBZ350UE/WBZ351UE modules have received certification of conformity in accordance with the China MIIT Notice 2014-01 of State Radio Regulation Committee (SRRC) certification scheme under Limited Modular Approval (LMA). Integration of this module into a final product does require additional radio certification (radiated spurious emission test and EMC test), provided installation instructions are followed and no modifications of the module are allowed. Refer to SRRC certificate available in WBZ350UE/WBZ351UE product page for expiry date.

45.7.1 Labeling and User Information Requirements

The WBZ350PE/WBZ351PE and WBZ350UE/WBZ351UE modules are labeled with its own CMIIT ID as follows:

WBZ350PE module

CMIIT ID: 24J999P60003

WBZ351PE module

CMIIT ID: 2023DJ14927

WBZ350UE module

CMIIT ID: 24J999P60004(M)

WBZ351UE module

CMIIT ID: 2023DJ14948(M)

When Host system is using an approved Full Modular Approval (FMA) radio: The host must bear a label containing the statement “This device contains SRRC approved Radio module CMIIT ID: 24J999P60003,CMIIT ID: 2023DJ14927, CMIIT ID: 24J999P60004(M),CMIIT ID: 2023DJ14948(M).

45.8 UKCA (UK Conformity Assessed)

The WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module is a UK conformity assessed radio module that meets all the essential requirements according to CE RED requirements.

45.8.1 Labeling Requirements for Module and User’s Requirements

The label on the final product that contains the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE module must follow UKCA marking requirements.

The UKCA mark above is printed on the module itself or on the packing label.

Additional details for the label requirement are available at:

https://www.gov.uk/guidance/using-the-ukca-marking#check-whether-you-need-to-use-the-new-ukca-marking.

45.8.2 UKCA Declaration of Conformity

Hereby, Microchip Technology Inc. declares that the radio equipment type the WBZ350PC/WBZ350PE/WBZ350UC/WBZ350UE and WBZ351PC/WBZ351PE/WBZ351UC/WBZ351UE modules are in compliance with the Radio Equipment Regulations 2017. The full text of the UKCA declaration of conformity for this product is available (under *Documents* > *Certifications*) at: www.microchip.com/en-us/product/WBZ351PE.

45.8.3 Helpful Websites

For more information on the UKCA regulatory approvals, refer to the www.gov.uk/guidance/placing-manufactured-goods-on-the-market-in-great-britain.

45.9 Other Regulatory Information

- For information about other countries' jurisdictions not covered here, refer to the www.microchip.com/design-centers/wireless-connectivity/certifications.
- Should other regulatory jurisdiction certification be required by the customer, or the customer needs to recertify the module for other reasons, contact Microchip for the required utilities and documentation.

46. Appendix B: Acronyms and Abbreviations

Table 46-1. Acronyms and Abbreviations

Acronyms	Abbreviations
AC	Analog Comparator
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AHB	Advanced High-Speed Bus
Apple® NCS	Apple Notification Center Service
AON	Always ON
APB	Advanced Peripheral Bus
API	Application Programming Interface
Arb	Arbiter
BFM	Boot Flash Memory
BG	Bandgap
Bluetooth® LE	Bluetooth Low Energy
BOD	Brown-out Detect
BOM	Bill of Material
BOR	Brown-out Reset
CCL	Configurable Custom Logic
CDM	Charged Device Model
CFG	System Configuration and Register Locking
CFGCON	Configuration Control
CLDO	Core Low Dropout Regulator
CLK	Clock
CM	Configuration Mismatch Reset
CM4CPU	Cortex M4F Processor
CMCC	Cortex M Cache Controller
CMR	Configuration Mismatch Reset
CNCON	Change Notice Control
CNEN	Change Notice Enable
CNF	Change Notice Flag
CNPDE	Change Notice Pull-down Enable
CNPUE	Change Notice Pull-up Enable
CNSTAT	Change Notice Status
CP	Charge Pump
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRM	Continuous Read Mode
CoreSight™ ROM	CoreSight ROM
CRU	Clock and Reset Unit
CS	Chip Select
CVD	Capacitive Voltage Divider
DAC	Digital-to-Analog Converter
DAP	Debug Access Port

.....continued

Acronyms	Abbreviations
DCC	Debug Communication Channel
DDR	Double Data Rate
DED	Double-bit Error Detected
DFE	Digital Front-End
DMAC	Direct Memory Access Controller
DMA RD	DMA-Read
DMA WR	DMA-Read
DMT	Deadman Timer
DMTCON	Deadman Timer Control
DMTEN	Deadman Timer Enable
DMTR	Deadman Timer Reset
DSCTRL	Deep Sleep System Controller
DSP	Digital Signal Processor
DSU	Device Service Unit
DSWDT	Deep Sleep WDT
DTI	Dead-Time Insertion
ECC	Error Correction Code
ECDH	Elliptic-curve Diffie-Hellman encryption
EIC	External Interrupt Controller
EOC	End-Of-Convert
EOS	Electrical Overstress
ERR	Error
ESD	Electrostatic Discharge
ETB	Embedded Trace Buffer
ETM	Embedded Trace Module
EVSYS	Event System Interface
FC	Flash Controller
FCS	Frame Check Sequence
FEM	Front-End Module
FPB	Flash Patch and Breakpoint Unit
FPU	Floating Point Unit
FRC	Fast RC Oscillator
FRCDIV	FRC Divider
FREQM	Frequency Meter
FSCM	Fail-safe Clock Monitor
FSM	Finite State Machine
GATT	Generic Attribute Profile (Bluetooth)
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose I/O
HBM	Human Body Model
HCI	Host Controller Interface
I2C	Inter-Integrated Circuit
IoT	Internet of Things
IRQ	Interrupt Request

.....continued

Acronyms	Abbreviations
ITM	Instrumentation Trace Macrocell
LDO	Low Dropout Regulator
LE	Low-Energy
LIN	Local Interconnect Network
LNA	Low-Noise Amplifier
LO	Local Oscillator
LPCLK	Low Power Clock Generation
LPRC	Low Power RC Oscillator
LUT	Each Look-up Table
LVD	Low-Voltage Detect
MCLRF	Master Clear Filter
MCS	Master Clock Switch
MCU	Microcontroller Unit
MFRQ	Match Frequency
MLDO	Main LDO
MPU	Memory Protection Unit
NDA	Non-Disclosure Agreement
NFRQ	Normal Frequency
NMCLR	External Reset
NMI	Non-maskable Interrupt
NMITR	NMI Time-out Reset
NPWM	Normal Pulse-Width Modulation
NVIC	Nested Vector Interrupt Controller
NVM	Non-Volatile Memory
NVMOP	Non-Volatile Memory Operation
OCD	On-Chip Debug
OTA	Over-the-Air
OTF	On-the-Fly
OTMX	Output Matrix
OTP	One Time Programmable
PA	Power Amplifier
PAC	Peripheral Access Controller
PCHE	Prefetch Cache
PFM	Program Flash Memory
PHY	Physical Layer
PLL	Phase-Locked Loop
PMD	Peripheral Module Disable
PMU	Power Management Unit
POR	Power-on Reset
POSC	Primary Crystal Oscillator
PPS	Peripheral Pin Select
PPW	Period and Pulse-Width
PSDU	PLCP Service Data Unit
PSM	Pulse Skipping Mode
PWM	Pulse Width Modulation

.....continued

Acronyms	Abbreviations
QoS	Quality of Service
QSPI	Quad I/O Serial Peripheral Interface
QSPI	Quad SPI interface
RAMECC	RAM Error Correction Code
RCON	Reset Control
REG	Register
RF	Radio Frequency
RFFE	RF Front-End
RoT	Root of Trust
RPC	Remote Procedure Call
RSSI	Receive Signal Strength Indication
RTC	Real-Time Counter
RTCC	Real-Time Clock Calender
RTOS	Real-Time Operating System
RTSP	Run-Time Self Programming
RX	Receive
RXC	Receive Complete
SEC	Single-bit Error Corrected
SERCOM	Serial Communication Interface
SERCOM I^2C	SERCOM Inter-Integrated Circuit
SFD	Start Frame Delimiter (Zigbee®)
SFR	Special Function Register
SHA	Secure Hash Algorithm
SMBus™	System Management Bus
SoC	System-on-Chip
SOSC	Secondary Crystal Oscillator
SPI	Serial Peripheral Interface
SRCON	Slew Rate Control
SSL	SPI Select Low
SWD	Serial Wire Debug
SWR	Software Reset
SWV	Serial Wire Viewer
Synth	Synthesizer
SYSRST	System Reset
TC	Timer Counter
TCC	Timer Counter Control
TCM	Tightly Coupled Memory
TPIU	Trace Port Interface Unit
TRX	Transmit/Receive
TX	Transmit
TXC	Transmit Complete
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VGA	Variable Gain Amplifier
WDT	Watchdog Timer

.....continued

Acronyms	Abbreviations
WDTO	Watchdog Time-out Reset
WDTR	Watchdog Timer Reset
XDS	Extreme Deep Sleep
XTAL	Crystal Oscillator
ZPBOR	Zero-power BOR

47. Document Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Section	Description
C	05/2024	1.1. PIC32CX-BZ3 SoC Ordering Information	Added ordering information of PIC32CX5109BZ31032
		1.2. WBZ35x Module Ordering Information	Added ordering information of WBZ350 Module
		4.6.1. PCB Antenna	Added antenna specification and radiation patterns for WBZ350 and WBZ351 modules
		45. Appendix A: Regulatory Approval	Added regulatory information for WBZ350 Module
B	02/2024	PIC32CX-BZ3 SoC Family Features	Changed Coremark® value from 2.36 to 2.42
		1.1. PIC32CX-BZ3 SoC Ordering Information	Ordering info for 32QFN SoC and Module is deleted
		1.2. WBZ35x Module Ordering Information	• Added UKCA, MIC, KCC, NCC, SRRC in Regulatory Certification for WBZ351 • Ordering info for 32QFN SoC and Module is deleted
		5. Pinout and Signal Descriptions List	• Updated the pin name description • Added note related to power supply pins and SWCLK pins
		9.1. Cortex M4F Processor	Removed “ARM® TrustZone® security” information in the Introduction section
		17.3.7. Peripheral Clock Generation (GCLK)	Added note “Only REFO1 - REFO4 can get routed to chip IOs”
		18. Power Management Unit (PMU)	Removed “Software Restore” topic
		32.8.3. CTRLC	Minor update
		36.13.1. ADCCON1	Added note for “Fast Synchronous UPB Clock bit”
		41.8.2. CTRLBCLR	Minor update
		41.8.6. WEXCTRL	Removed “DTIEN3” bit
		41.8.16. WAVE	Removed “SWAP3” bit
		43.5. Power Supply DC Module Electrical Specifications	Minor update
		43.11. Wake-Up Timing from Low Power Modes AC Electrical Specifications	Added new topic
		43.12. I/O PIN AC/DC Electrical Specifications	Updated with new values
		43.16. DAC Module Electrical Specifications	Updated with new values
		43.19. SPI Module Electrical Specifications	Updated with new values
		43.20. UART AC Electrical Specifications	Updated with new values
		43.21. I2C Module Electrical Specifications	Updated with new values
		43.23. TCx Timer Capture Module AC Electrical Specifications	Added Timing Diagram
		43.24. TCCx Timer Capture Module AC Electrical Specifications	Added Timing Diagram
		43.26. Frequency Meter AC Electrical Specifications	Minor update
		43.27. SWD 2-Wire AC Electrical Specifications	Added Timing Diagram
A	11/2023	Document	Initial revision

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure

that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

ISBN: 978-1-6683-4387-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Hod Hasharon** Tel: 972-9-775-5100 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820